

**CICS/VSE to CICS OS/2 and CICS/6000
A Guide to Client/Server Solutions**

Document Number GG24-4262-00

June 1994

International Technical Support Organization
Boeblingen Center

Take Note!

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

First Edition (June 1994)

This edition applies to:

- CICS/VSE* Version 2, Release 2, Program Number 5686-026, a standard product included in the VSE/ESA Operating System
- CICS OS/2* Version 2, Program Number 5688-101, for use with the OS/2 Operating System
- IBM AIX* CICS/6000 Version 1 Release 1, Program Number 5765-148, and IBM AIX Client for CICS/6000, Program Number 5765-152, running on the AIX Operating System

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

An ITSO Technical Bulletin Evaluation Form for reader's feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. 3222 71032-02
Postfach 1380
71032 Boeblingen, Germany

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

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

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

Abstract

This document describes how to implement applications that use distributed processing techniques provided by the CICS family of products. Although CICS is available on many operating system platforms the scope of this document is limited to conversations between VSE/ESA, AIX and OS/2 operating systems.

This publication is intended for system engineers or programmers responsible for implementing distributed applications using CICS/VSE as host and CICS OS/2 or CICS/6000 as workstation platforms. The reader is assumed to have a good knowledge of CICS transaction processing concepts and a working knowledge of the subject operating system platforms. In addition, he should be familiar with the corresponding communication protocols, that is ACF/VTAM*, TCP/IP and NetBIOS.

(167 pages)

Contents

Abstract	iii
Special Notices	xiii
Preface	xv
How This Document is Organized	xv
Related Publications	xvi
International Technical Support Organization Publications	xvii
Acknowledgments	xvii

Part 1. Introduction

1

Chapter 1. Client/Server Computing with CICS/VSE	3
---	---

Part 2. Implementing CICS/VSE to CICS OS/2 Client/Server Solutions

5

Chapter 2. CICS OS/2 to CICS/VSE Overview	7
2.1 CICS OS/2	7
2.2 The CICS OS/2 Server	7
2.3 The CICS OS/2 Client	7
2.4 Our Distributed Application (BMS based)	8
2.5 Our Distributed Application (PM based)	8
Chapter 3. CICS OS/2 to CICS/VSE Implementation Environment	9
3.1 Hardware	9
3.2 Software	9
3.3 Network Configuration	9
Chapter 4. Define and Customize the VSE/ESA Host	11
4.1 Host Definition Overview	11
4.1.1 IOCDS and VM Definitions for the 3172	12
4.1.2 Attach 3172 to VSE Machine	12
4.1.3 3172 Definition in VSE/ESA	13
4.1.4 3172 Customization	13
4.1.5 Customization in VSE/VTAM	15
4.2 CICS/VSE Customization	19
4.2.1 CICS/VSE Resource Definition Using Macro	19
4.2.2 CEDA Transaction/RDO	22
Chapter 5. Define and Customize the CICS OS/2 Systems	27
5.1 LAPS Installation and Customization	27
5.2 Communication Manager Customization	29
5.2.1 Configure DLC Token-Ring	30
5.2.2 Defining Local Node Characteristics	31
5.2.3 Define SNA Host Connections	32
5.2.4 Defining Optional SNA Features	32
5.2.5 Defining the Partner LU	35
5.2.6 Defining the Logon Mode	36
5.2.7 Defining Transaction Programs	36
5.3 CICS OS/2 Server Customization	40

5.3.1	System Initialization Table (SIT)	41
5.3.2	TCS Terminal Control Table (System Entry)	42
5.3.3	Program Control Table (PCT)	44
5.3.4	Processing Program Table (PPT)	45
5.4	CICS OS/2 Client Customization	46
5.4.1	Enabling NETBIOS	46
5.4.2	Install Client Software	46
5.4.3	Tailoring the Client Configuration	46
5.4.4	Folders for Client Control and TR Emulator	46
Chapter 6. Overview of our Sample Application		49
6.1	Front-end Application Implementation on CICS OS/2 (BMS based)	49
6.1.1	Install MicroFocus COBOL	49
6.1.2	Translate BMS Maps	49
6.1.3	Translate, Compile and Link Front-end	50
6.2	Application Implementation on CICS/VSE	50
6.2.1	Translate, Compile and Link Backend	50
6.2.2	Create VSAM Base File	51
6.2.3	Create VSAM Alternate Index	51
6.2.4	Sample of BMS Application Screen	51
6.3	Front-end Application Implementation on OS/2 (PM based)	52

Part 3. Implementing CICS/VSE to CICS/6000 Client/Server Solutions 53

Chapter 7. CICS/6000 to CICS/VSE Overview		55
7.1	CICS/6000	55
7.2	The CICS/6000 Server	55
7.3	The CICS/6000 Client	55
7.4	Our Distributed Application	56
Chapter 8. CICS/6000 to CICS/VSE Implementation Environment		57
8.1	Hardware	57
8.2	Software	57
8.3	Network Configuration	58
Chapter 9. Installing and Configuring the IBM AIX CICS/6000 Server		61
9.1	Planning a CICS/6000 Installation	61
9.2	Preliminary AIX Configuration	62
9.3	CICS/6000 Installation	62
9.4	CICS/6000 Server Configuration	63
9.4.1	Configuring DCE for CICS/6000	63
9.4.2	Configuring Encina for CICS/6000	63
9.4.3	Creating and Starting a CICS/6000 Region	64
9.5	CICS/6000 Server Installation Summary	64
Chapter 10. Installing and Configuring the IBM AIX CICS/6000 Client		65
10.1	Planning an AIX CICS/6000 Client Installation	65
10.2	Preliminary AIX Configuration	65
10.3	Configuring DCE for CICS/6000 Client Support	66
10.4	Accessing the CICS/6000 Region	66
Chapter 11. Connecting the CICS/6000 Server to the Host		67
11.1	Introduction to CICS/6000 Intercommunication	67
11.1.1	Network Protocol Considerations	68

11.1.2 Using SNA Communication	68
11.1.3 Using TCP/IP Communication	69
11.2 VSE/ESA Host Customization	69
11.2.1 ACF/VTAM Customization	69
11.2.2 CICS/VSE Customization	70
11.3 CICS/6000 Connection Customization	73
11.3.1 AIX Environment Customization	73
11.3.2 Token-Ring Adapter Customization	74
11.3.3 DCE Customization	75
11.3.4 AIX SNA Services/6000 Customization	75
11.3.5 CICS/6000 Customization for Inter System Communications	89
11.4 Interconnection Operations	95
11.5 Monitoring the Encina PPC Gateway	97
11.6 Operational Hints	97
11.6.1 VSE Host Operational Hints	97
11.6.2 CICS/6000 Operational Hints	98
11.6.3 AIX SNA Services/6000 Operational Hints	98
11.7 CICS/6000 to CICS/VSE Customization Summary	100
11.7.1 Reference of Implementation Tasks	100
11.7.2 CICS/6000 to CICS/VSE Name Cross Reference	102
Chapter 12. Implementing an Application in CICS/6000	103
12.1 Application Description	103
12.2 Application Development*	104
12.3 CICS Table Entries	105
12.3.1 Transaction Definition for Sample Transaction 'TECH'	106
12.3.2 BMS Mapset Definition for Sample Application	107
12.3.3 Defining the Sample Front-end Program	108
12.3.4 Defining the Sample Backend Program	109

Part 4. Operation and Definition Examples 111

Appendix A. VSE Host Operation and Debugging	113
A.1 VSE Host Operation Hints	113
A.1.1 CEMT	113
A.1.2 VTAM Display Sessions	113
A.1.3 How to Trace with VTAM	114
A.2 VTAM Start List	116
A.3 VTAM Trace Example	117
A.3.1 VTAM Trace to OS/2	117
A.4 VSE Virtual Machine Directory	121
Appendix B. Definition Examples for CICS/6000 and AIX SNA Services/6000	123
B.1 AIX SNA Services/6000 Profiles	123
B.2 Definition Examples for CICS/6000 Region	128
B.3 Definition Examples for CICS/6000 Communication	130
B.4 Application Definition Examples	131
B.4.1 Transaction Definition	131
B.4.2 Program Definitions	132
Appendix C. Sample Application Job Streams and Definitions	133
C.1 Front-end Program Source (File Name TECCPROG.CCP)	133
C.2 Backend Program Source (Program CICSVSE)	145
C.3 Compile Backend Program Job	151

C.4 BMS Maps (File Name NEW1MS.MAP)	152
C.5 COBOL Directives (File COBOL.DIR)	157
C.6 CICS OS/2Tailor Environment Variables (File CICSENV.COMD)	158
C.7 Create VSAM Base File	158
C.8 Load VSAM Base File	159
C.9 Sample VSAM Load Program	159
C.10 Define VSAM Alternate Index	160
C.11 Build VSAM Alternate Index	161
List of Abbreviations	163
Index	165

Figures

1.	CICS OS/2 to CICS/VSE Network Diagram	10
2.	3172 IOCDS Entries	12
3.	Device Address Definitions in SYSTEM CONFIG File	12
4.	VM Directory Entry for VSE Machine	13
5.	VSE IPL Procedure	13
6.	3172 ICP Configuration	14
7.	VSE/VTAM XCA Major Node for 3172	15
8.	VSE/VTAM SWNET Major Node for 3172	17
9.	LOGMODE Entries	18
10.	Application Major Node	19
11.	DFHSIT Parameters	20
12.	DFHCNV Coding for Sample COMMAREA	21
13.	DFHFCT Entries for Application File	22
14.	Group Definition of CICSOS2	23
15.	Connection for CICS OS/2	23
16.	Session Definition for CICS OS/2 Connection	24
17.	Application Program Definition	26
18.	LAPS Configuration Menu for the CICS OS/2 Server	28
19.	LAPS Token-Ring Adapter Customization	29
20.	Token-Ring DLC Adapter Parameters	30
21.	Local Node Characteristics	31
22.	Local Node Options Alias Name	31
23.	Connection Definition Panel	32
24.	List of Optional SNA Features	33
25.	Local LU Definition Panel	34
26.	Partner LU Definition	35
27.	Mode Definition Panel	36
28.	Transaction Program Definition Main Menu	37
29.	CPMI Transaction Program Definition	37
30.	CPMI Transaction Program Definition (Additional TP Parameters)	38
31.	CVMI Transaction Program Definition	38
32.	CRSR Transaction Program Definition	39
33.	CRTE Transaction Program Definition	40
34.	System Initialization Table-1	41
35.	System Initialization Table-2	42
36.	Terminal Control Table (System Entry)	43
37.	PCT Program Control Table for TECH	44
38.	Remote Program Entry for Backend Program	45
39.	Application BMS Screen	51
40.	Application PM Window	52
41.	CICS/6000 to CICS/VSE Network Diagram	59
42.	CICS/6000 VSE/VTAM SWNET Major Node for 3172	70
43.	Connection for CICS/6000	71
44.	CICS/6000 Session Definition	72
45.	AIX Environment Customization	73
46.	Token-Ring Adapter Customization	74
47.	DCE Customization for Encina PPC Gateway	75
48.	VSE62 Attachment Profile	77
49.	VSE62 Token-Ring Logical DLC Profile	78
50.	VSE62 Token-Ring Physical DLC Profile	79
51.	LU62VSE Connection Profile	80

52.	CICS6K Local Logical Unit	81
53.	LU62VSE Mode List Profile	82
54.	LU62PS Mode Profile	83
55.	LU62VSEt TPN List Profile	84
56.	TDEFAULT TPN Profile	85
57.	RDEFAULT Remote TPN List Profile	86
58.	RDEFAULT Remote TPN Profile	87
59.	SNA Profile	88
60.	VSE62 SNA Control Point Profile	89
61.	CD for SNA Connected System	91
62.	CD for the Encina PPC Gateway	93
63.	PPC's configFile	96
64.	Manage CICS/6000 Resource(s)	105
65.	Defining the Sample Transaction	106
66.	MEW1MS Mapset Definition	107
67.	Defining the Sample Program TECCPROG	108
68.	Defining the Backend Program CICSVSE	109
69.	CEMT Inquiry Connection	113
70.	The Status of the Sessions	113
71.	Console Log for VTAM Trace	115
72.	VTAM Trace Data for CICS OS/2	117

Tables

1.	LPPs Required for an IBM AIX CICS/6000 Server	61
2.	LPPs for IBM AIX CICS/6000 ISC Support	61
3.	Summary of CICS/6000 Installation Variables	64
4.	LPPs Required for IBM AIX Client for CICS/6000	65
5.	Encina PPC Gateway Administration Tool Services	97
6.	VSE/ESA Connection Customization Summary	100
7.	AIX Connection Customization Summary	101
8.	CICS/6000 Cross Reference Summary	102

Special Notices

This publication is intended to help Systems Engineers, Marketing Representatives and Customers to implement Client/Server solutions using the CICS/VSE, CICS OS/2 and CICS/6000 transaction processing environment.

The document describes how to install and customize the CICS environment on each of the three platforms involved, that is VSE/ESA for the host system, OS/2 for PS/2* and AIX* for RISC/6000* workstations. In addition, the implementation of a sample application is explained.

The information in this publication is not intended as the specification of any programming interfaces that are provided by components of various solutions discussed here. See the PUBLICATIONS section of the IBM Programming Announcement for respective products described in this document 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, 208 Harbor Drive, Stamford, CT USA 06904.

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

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

ACF/VTAM
AIX/6000*
CICS
CICS/ESA

AIX
Application Development
CICS OS/2
CICS/MVS

CICS/VSE
CICS/6000
IBMLink
MVS/XA
OS/400
PROFS*
RISC System/6000
VSE/ESA

CICS/400
IBM
MVS/ESA
OS/2
Presentation Manager
PS/2
VM/ESA
VTAM

The following terms, which are denoted by a double asterisk (**) in this publication, are trademarks of other companies:

DOS
HP** and HP/UX
IEEE

Macintosh
Micro Focus
Microsoft, Windows
Motif
UNIX
Windows
Windows NT

Microsoft Corporation
Hewlett Packard Company
Institute of Electrical and Electronics
Engineers
Apple Computer Inc.
Micro Focus Ltd.
Microsoft Corporation
Open Software Foundation, Inc.
UNIX System Laboratories, Inc.
Microsoft Corporation
Microsoft Corporation

Preface

The purpose of this document is to help IBM personnel and/or customers to implement *Client/Server Computing* solutions that are based on the distributed transaction capabilities of the CICS family of products. It contains a description of the installation, customization and implementation steps required for a sample application running on VSE/ESA, OS/2 and AIX. The job streams and source code of the sample application are provided in the appendix of this book and are available as package VSECSDEM on disk IBMVSE managed by VMTOOLS at BOEVM3.

This document is intended for persons who want to implement distributed transaction processing applications running on VSE/ESA* hosts and PS/2 or RISC/6000 workstations.

How This Document is Organized

The document is organized as follows:

- Chapter 1, "Client/Server Computing with CICS/VSE"
This chapter contains a very brief introduction into the functions and facilities available for CICS to CICS communication between CICS/VSE and other CICS platforms.
- Chapter 2, "CICS OS/2 to CICS/VSE Overview"
This chapter provides a brief overview of CICS OS/2 to CICS/VSE functions.
- Chapter 3, "CICS OS/2 to CICS/VSE Implementation Environment"
This chapter describes our hardware and software environment and shows the network configuration we used.
- Chapter 4, "Define and Customize the VSE/ESA Host"
This chapter provides the definitions required in VSE/ESA to implement our distributed application.
- Chapter 5, "Define and Customize the CICS OS/2 Systems"
This chapter describes how to implement the CICS OS/2 server and client for our distributed application.
- Chapter 6, "Overview of our Sample Application"
This chapter gives an overview of the distributed application we used in the CICS OS/2 to CICS/VSE environment.
- Chapter 7, "CICS/6000 to CICS/VSE Overview"
This chapter provides a brief overview of CICS/6000 to CICS/VSE functions.
- Chapter 8, "CICS/6000 to CICS/VSE Implementation Environment"
This chapter describes our hardware and software environment and shows the network configuration we used.
- Chapter 9, "Installing and Configuring the IBM AIX CICS/6000 Server"
This chapter describes how to install and customize the CICS/6000 server.
- Chapter 10, "Installing and Configuring the IBM AIX CICS/6000 Client"

This chapter describes how to install and customize the CICS/6000 client.

- Chapter 11, “Connecting the CICS/6000 Server to the Host”

This chapter describes how the CICS/6000 server and the VSE/ESA host are connected to each other.

- Chapter 12, “Implementing an Application in CICS/6000”

This chapter describes how our distributed sample application is implemented in the CICS/6000 environment.

- Appendix A, “VSE Host Operation and Debugging”
- Appendix B, “Definition Examples for CICS/6000 and AIX SNA Services/6000”
- Appendix C, “Sample Application Job Streams and Definitions”

Related Publications

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

CICS OS/2 Publications

- *CICS OS/2 Application Programming, SC33-0883*
- *CICS OS/2 Customization, SC33-0880*
- *CICS OS/2 Intercommunication, SC33-0825*
- *CICS OS/2 Operation, SC33-0881*
- *CICS OS/2 Problem Determination, SC33-1005*

CICS/6000 Publications

- *AIX CICS/6000 Application Programming Guide, SC33-0814*
- *AIX CICS/6000 Application Programming Reference, SC33-0886*
- *AIX CICS/6000 CICS-Supplied Transactions, SC33-0813*
- *AIX CICS/6000 Customization and Operation, GC33-0816*
- *AIX CICS/6000 Intercommunication Release 1, SC33-0815*
- *AIX CICS/6000 Messages and Codes, SC33-0817*
- *AIX CICS/6000 Problem Determination, SC33-0818*

CICS/VSE Publications

- *CICS/VSE System Programming Reference, SC33-0711*
- *CICS Family: Interproduct Communication, SC33-0824*
- *CICS Communicating from CICS/ESA and CICS/VSE, SC33-0825*
- *CICS/VSE Intercommunication Guide Version 2 Release 2, SC33-0701*

Other IBM Publications

- *AIX SNA Server/6000: User Guide, SC31-7002*
- *AIX SNA Server/6000: Configuration Reference, SC31-7014*
- *VTAM Messages and Codes, SC31-6433*
- *VTAM Operation, SC31-6435*
- *VTAM Resource Definition Reference, SC31-6438*

- *VTAM Network Implementation Guide, SC31-6434*
- *SNA Network Product Formats, LY43-0081*

Micro Focus COBOL Publications**

- *Micro Focus COBOL for DOS**, Windows** and OS/2, COBOL System Reference*
- *Micro Focus COBOL for DOS, Windows and OS/2, Language Reference, April 1993*
- *Micro Focus COBOL for UNIX**, Getting Started, April 1993*
- *Micro Focus COBOL for UNIX, COBOL User Guide, April 1993*

International Technical Support Organization Publications

- *AIX CICS/6000 Installation and Configuration: A Guide to Implementation, GG24-4091*

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

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

To get listings of redbooks online, VNET users may type:

TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG

How to Order ITSO Technical Bulletins (Redbooks)

IBM employees in the USA may order ITSO books and CD-ROMs using PUBORDER. Customers in the USA may order by calling 1-800-879-2755 or by faxing 1-800-284-4721. Visa and Master Cards are accepted. Outside the USA, customers should contact their IBM branch office.

Customers may order hardcopy redbooks individually or in customized sets, called GBOFs, which relate to specific functions of interest. You may also order redbooks in online format on CDROM collections, which contain the redbooks for multiple products.

Acknowledgments

The advisor for this project was:

Werner Stieber
International Technical Support Organization, Center Boeblingen

The authors of this document are:

Ken Mitchell
IBM Canada

Mark Toplak
IBM Slovenia

This publication is the result of a residency conducted at the International Technical Support Organization, Center Boeblingen.

Thanks to the following people for their advice and guidance provided in the production of this document:

John Bennett
ITSO Center Boeblingen

Patricia Savage
IBM United Kingdom

Arnd Schneider
IBM Germany

Yukihiko Yoshida
ITSO Center Boeblingen

Part 1. Introduction

This document describes the installation and customization aspects of CICS/VSE to CICS OS/2 and CICS/VSE to CICS/6000, two client/server solutions available within the CICS family of products.

This part provides a brief introduction into client/server computing with CICS/VSE.

Chapter 1. Client/Server Computing with CICS/VSE

As a member of the CICS family of transaction processing monitors, CICS/VSE provides all functions which allow for cooperation with other CICS systems. These facilities include:

- **EXEC CICS API** a common programming interface
- **APPC/LU 6.2 protocols** to establish conversations with other CICS systems
- **Inter System Communication (ISC)** providing several techniques for CICS to CICS communication, that is **Transaction Routing (TR)**, **Distributed Program Link (DPL)**, and **Distributed Transaction Processing (DTP)**.

CICS/VSE is, therefore, well suited for creating applications which can be distributed across all CICS platforms, that is, implementing a client/server environment which is based on transaction processing.

Currently the CICS family comprises:

- **CICS/ESA*** for MVS/ESA* environments
- **CICS/MVS*** for MVS/XA* environments
- **CICS/VSE** for VSE/ESA environments
- **CICS/400*** for OS/400* environments
- **CICS/6000*** for AIX environments
- **CICS OS/2** for OS/2 environments

Statements of direction have been made to provide CICS on the following non-IBM platforms:

- **HP/UX****
- **Apple Macintosh****
- **Windows NT****

From the ISC facilities mentioned above we will use **DPL** in the application example described in this book. This is because programmers are very familiar with this technique of using a "called" subroutine to perform various tasks without disturbing the main flow of the application. In addition it is, from a conceptual point of view, very similar to the **Remote Procedure Call (RPC)** within TCP/IP.

Distributed Program Link is accomplished within CICS via the 'EXEC CICS LINK PROGRAM()' command which passes control to a second program, named in the LINK command. The second program executes and, when it completes, control returns to the first program at the instruction following the LINK command. The linked-to program can return data to the linking program if the LINK command has used the COMMAREA option to pass the address of a communication area. For more details on CICS Distributed Program Link refer to *CICS/VSE Intercommunication Guide Version 2 Release 2, SC33-0701*.

CICS/VSE 2.2 supports inbound and outbound Distributed Program Link from CICS/6000 and CICS/6000.

Part 2. Implementing CICS/VSE to CICS OS/2 Client/Server Solutions

Chapter 2. CICS OS/2 to CICS/VSE Overview

The next four chapters document the parameters used to customize the environment described in Figure 1 on page 10 to make our sample application run in a distributed fashion. That is, with our front-end program running on CICS OS/2 using Distributed Program Link to a backend program on CICS/VSE.

2.1 CICS OS/2

CICS OS/2 is a member of the CICS family which enables CICS applications to run on an OS/2* platform. In addition to the standard Inter System Communication (**ISC**) facilities, CICS OS/2 also provides an **External Call Interface (ECI)** and an **External Presentation Interface (EPI)** by which non-CICS applications can call CICS programs or transactions directly. These non-CICS programs may use a GUI for presentation services.

- By using the CICS **EPI**, the non-CICS program appears to CICS OS/2 as one or more 3270 terminals. It may start CICS* transactions and send and receive 3270 data streams.
- By using the CICS **ECI**, the non-CICS program may call a CICS program and pass it data via a COMMAREA. The called CICS program appears as if it had been called using an EXEC CICS LINK command with the COMMAREA option.
- The CICS OS/2 application may then invoke **ISC** facilities to communicate with CICS/VSE functions or applications. It can then return to the non-CICS program where the data may be presented to the user with tools such as Presentation Manager*.
- By using a facility called External Transaction Initiation, CICS **ETI**, non-CICS programs on a CICS OS/2 server may invoke CICS transactions in a dynamically created 3270 terminal environment.

2.2 The CICS OS/2 Server

The multi-user version of CICS OS/2 creates a CICS environment under OS/2 which can support communication with other CICS platforms and other client workstations. The multi-user version communicates with other CICS platforms via APPC and with other workstations via NetBIOS. OS/2 programs on the CICS OS/2 server may use ECI, EPI or ETI to invoke CICS programs and transactions. A 3270 BMS style interface is also available on the CICS OS/2 server to invoke CICS transactions. CICS command level programs written in COBOL or C using minimum BMS will run on CICS OS/2. An OS/2 PS/2 can function as a LAN server as well as a CICS OS/2 server and as a gateway, via CICS **ISC**, to a CICS/VSE host.

2.3 The CICS OS/2 Client

This scenario can be further extended to OS/2, DOS or Windows client machines which are LAN attached. The CICS OS/2 Client Support software for DOS, Windows, or OS/2 can be installed directly on these workstations or invoked from a LAN server disk. This code is often referred to as the '**CICS skinny client**' code.

Non-CICS application programs running on these client workstations can communicate with this 'skinny client' code using the **ECI** or **EPI** interfaces. The clients communicate via NetBIOS to the CICS OS/2 server machine across a Token-Ring. The CICS OS/2 server may in turn use ISC to communicate with a CICS/VSE host.

Alternatively, the client machines can invoke CICS **3270** style applications directly. The 'skinny client' software in this case simply creates a 3270 environment on the client machine then performs a Transaction Route to the CICS OS/2 server. The server may, in turn, invoke other ISC facilities to communicate with a CICS/VSE host.

2.4 Our Distributed Application (BMS based)

The application used to demonstrate the implementation of CICS OS/2 to CICS/VSE in this document is composed of two programs. These two programs can be on the same or different CICS platforms.

- The front-end program is a CICS program which handles the interface to the user via Basic Mapping Support (BMS) maps.
- The backend program handles VSAM file operations.

The front-end program LINKs to the backend passing a COMMAREA. The backend updates this COMMAREA with data from the VSAM file and RETURNS control to the front-end.

2.5 Our Distributed Application (PM based)

We also tested a Presentation Manager based front-end program. This non-CICS program was written in the C language for the OS/2 environment and CALLED the same CICS/VSE backend as our BMS based front-end program. It used **ECI** to pass a COMMAREA to the linked CICS backend.

After installing and running the BMS based application, we could 'snap on' the PM based front-end to an OS/2 skinny client **with no changes** to the CICS OS/2 server or CICS/VSE host environments.

In our scenario there are two major benefits to using the PM front-end:

- A much friendlier user interface.
- Reduced resource requirements on the CICS OS/2 server. Our BMS based transaction was written as conversational. By moving the front-end off the CICS OS/2 server and on to the individual client machines, significant resources were freed on the server.

Chapter 3. CICS OS/2 to CICS/VSE Implementation Environment

This chapter describes the hardware and software used in our project.

Our environment consists of:

- a VSE/ESA host running CICS/VSE
- a CICS OS/2 server workstation
- a CICS OS/2 client workstation

CICS/VSE, CICS OS/2 server and CICS OS/2 client are connected to each other via a Token-Ring LAN running SNA protocols. The network is illustrated in Figure 1 on page 10.

3.1 Hardware

- An IBM 9221 Model 150 as the host with:
 - 64MB main memory
 - Token-Ring connection via channel attached IBM 3172
- An IBM PS/2 Model 90 attached to the IBM Token-Ring LAN as CICS OS/2 server
- An IBM PS/2 Model 70 attached to the IBM Token-Ring LAN as CICS OS/2 client

3.2 Software

- VM/ESA R2.1 with GCS and ACF/VTAM V3.4.1 in the IBM 9221
- VSE/ESA Version 1.3.3 guest under VM in the IBM 9221 with
 - ACF/VTAM V3.4.0
 - CICS/VSE V2.2
- Interconnect Controller Program (ICP) Version 3.2 in the IBM 3172 Model 3
- OS/2 Version 2.1 in the server workstation running
 - CM/2 V1.0
 - CICS OS/2 V2.0 with Corrective Service Diskette (CSD) 1
 - Micro Focus COBOL for OS/2 V3.1.31
- OS/2 Version 2.1 and downloaded CICS OS/2 V2.0 in the client workstation

3.3 Network Configuration

Figure 1 on page 10 illustrates the network described above including the MAC-addresses used in the appropriate VTAM, CM/2 and 3172 ICP definitions.

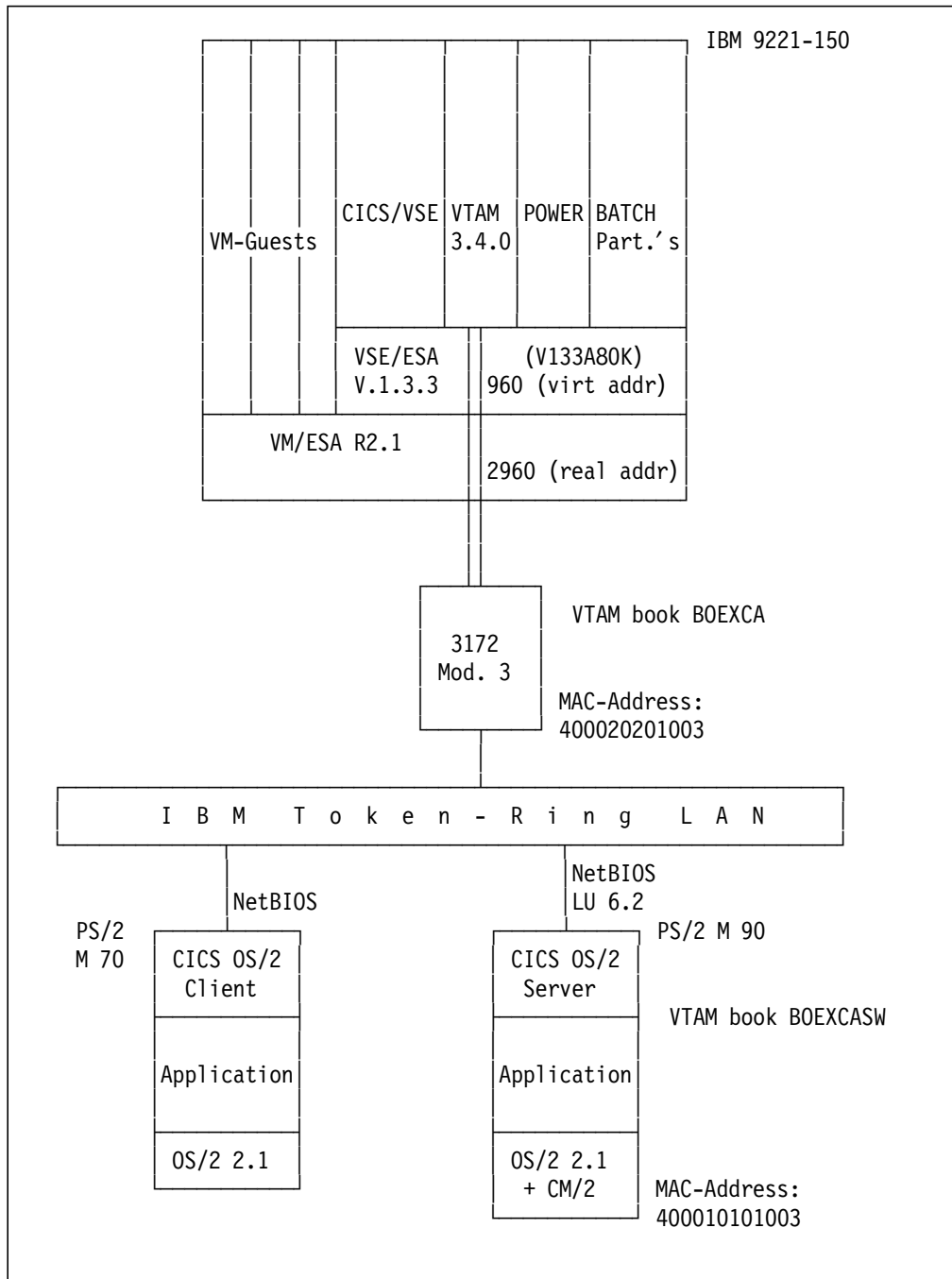


Figure 1. CICS OS/2 to CICS/VSE Network Diagram

Chapter 4. Define and Customize the VSE/ESA Host

This chapter describes the definitions in **VM**, **VSE**, **CICS/VSE** and **VTAM** which were used to implement our distributed CICS application.

Our CICS OS/2 is defined to the owning VTAM as an **independent LU**, or Physical Unit (PU) Type 2.1 in order to support APPC parallel sessions. In a typical CICS OS/2 Version 2 environment, parallel sessions would be required to allow CICS OS/2 to perform ISC functions to a host for multiple clients simultaneously.

The independent LU is defined in VSE VTAM, CICS/VSE and the OS2 Communications Manager.

To automatically start up the 3172 and other major nodes during VTAM* initialization, their names must be included in the VTAM configuration list.

Definition for the skinny clients

No SNA definitions are required for the OS/2, DOS or Windows client workstations to communicate to CICS/VSE via the CICS OS/2 server. These workstations communicate via NetBIOS to the CICS OS/2 server. ISC sessions are established automatically between CICS/VSE and the CICS OS/2 independent LU on behalf of the clients whenever necessary. SNA definition is, therefore, **not** required.

Figure 1 on page 10 provides a summary of the attachment configuration and VTAM book names. This diagram should be referenced while studying the sample definitions provided in this document.

4.1 Host Definition Overview

In our configuration the VSE virtual machine named V133A80K owns the 3172.

The following must be prepared to permit VSE to control the 3172:

- The 3172 must be defined to VM/ESA.
 - In our case, this is done via the 9221 machine's **IOCDS** and VM/ESA R2.1's automatic device sensing and dynamic configuration capability.
- The address of the 3172 must be **attached** to the VSE machine.
 - In our case this is done via a **DEDICATE** statement in the VM directory for V133A80K.
- The address of the 3172 must be defined in the **VSE IPL** procedure.
 - This is done via an **ADD** statement in our **\$IPLBOE.PROC**.
- The 3172 **Interconnect Control Program (ICP)** must be configured to reflect its operating environment.
 - In our case this is done with the customization parameters found in Figure 6 on page 14.
- **VSE/VTAM** books and tables must be created to describe the 3172 and Token-Ring.

- In our case by creating the VTAM books BOEXCA, BOEXCASW, and the LOGMODE entry LU62PS.

To permit CICS/VSE to establish communication with CICS OS/2 the following must be prepared:

- The VTAM **application major node** entry for CICS/VSE should reflect some special parameters to allow sessions with Independent LUs.
- CICS **connections** and **sessions** must be defined.

4.1.1 IOCDs and VM Definitions for the 3172

4.1.1.1 IBM 3172 Definition in IBM 9221 IOCDs

The 3172 is a channel attached device and is defined in the 9221 IOCDs (I/O Configuration Data Set).

The following entries in the IOCDs describe the 3172:

```

CHPID PATH=((29)),TYPE=BL
CNTLUNIT CUNUMBR=3172,PATH=(29),UNIT=3172,UNITADD=((60,32)),          X
          SHARED=N,PROTOCOL=S4
IODEVICE ADDRESS=(2960,32),CUNUMBR=(3172),UNIT=3172

```

Figure 2. 3172 IOCDs Entries

4.1.1.2 IBM 3172 Definition to VM/ESA

VM/ESA R2.1 uses dynamic configuration and automatic device sensing. Static device definition is not required. The statements shown in Figure 3 are included in the VM SYSTEM CONFIG file, to activate and auto-sense device addresses 0000-FFFF during system initialization.

```

Devices,
  Online_at_IPL 0000-FFFF,
  Sensed       0000-FFFF

```

Figure 3. Device Address Definitions in SYSTEM CONFIG File

4.1.2 Attach 3172 to VSE Machine

For the 3172 to be owned by VSE/ESA, it must be attached or dedicated to the VSE/ESA virtual machine. In our environment this was accomplished with the **DEDICATE** statement as in Figure 4 on page 13 where **2960** is the real address and **960** the virtual address.

```

USER V133A80K PASSWORD 0032M 64M G
ACCOUNT V133A80K V133A80K
OPTION MAXCON 150
MACHINE ESA
CONSOLE 0009 3215 T
.
.
DEDICATE 960 2960
.
.
MDISK 191 3380 001 049 DISK01 MW ALL
LINK MAINT 190 0190 RR
MDISK 991 3380 1770 885 DISK01 MW RPASS WPASS MPASS

```

Figure 4. VM Directory Entry for VSE Machine

4.1.3 3172 Definition in VSE/ESA

The 3172 is **ADDED** via the IPL procedure as in Figure 5. VSE/ESA is to treat it as a channel to channel adapter (CTCA). The device is not to be sensed at IPL time (EML). This guarantees that the device characteristics implied by CTCA are not changed dynamically at IPL time by VSE/ESA.

```

CATALOG $IPLBOE.PROC REPLACE=YES
009,$$$SUPX,VSIZE=120M,VIO=512K,VPOOL=128K,LOG
ADD 009,3277
ADD 00C,2540R
.
.
.
ADD 960,CTCA,EML           3172 GATEWAY
.
.
ADD FFF,CONS                DUMMY CONSOLE, DO NOT DELETE
.
.
SVA SDL=300,GETVIS=768K,PSIZE=640K
/+

```

Figure 5. VSE IPL Procedure

4.1.4 3172 Customization

Parameters used in the customization of the Interconnect Control Program (ICP) for the 3172 are described in Figure 6 on page 14. There are two parameters here which must match definitions elsewhere in our example:

- **Subchannel.** This value must match the control unit and unit portion of the real address defined to the host for the 3172. In our case the real address **2960** is defined to VM/ESA
- **Node address.** This value is the **MAC Address** used by OS/2's Communication Manager to allow our CICS OS/2 server to find the 3172 gateway to our CICS/VSE host.

3172-3 Configuration Summary

3172 Name : IS23172
 3172 Type : 3172-3 LAN Gateway
 Int Enhancement Feature (IEF) : Yes
 User Data : LAN Gateway for 9221-150(IS2)
 Location : Building 02 Room 018

ICP Base Code Version..... : 3.02.00
 ICP IEF Code Version..... : 3.02.00
 APARs/Patches applied..... : None

Profile Name : TRL3172

Slot	Name	Adapter Type
1	Unassigned	
2	Unassigned	
3	Unassigned	
4	CHAN29	Parallel Channel
5	Reserved	
6	Unassigned	
7	TOK1	Token-Ring 16/4
8	Fixed Disk	

LAN Function Name : SNAGATE
 Channel Adapter Name : CHAN29

Subchannels	To Channel	To LAN	LAN Adapter	Block Delay	Maximum Response
60	TOCHN060	TOLAN060	TOK1	10	100

Slot : 4
 Adapter Name : CHAN29
 Adapter Type : Parallel Channel
 Transfer Mode and Speed : 4.5 MB Data Streaming
 SNA Management Services : No

Slot : 7
 Adapter Name : TOK1
 Adapter Type : Token-Ring 16/4
 Relative Adapter Number : 0
Node Address : **400020201003**
 Data Rate (Mbps) : 4
 To Operator Facility : No
 Combined Functional Addresses : 000000000000

IEEE** 802.2 (LLC)
 Response Timer (T1) : 10 = 2000 ms
 Acknowledgment Timer (T2) . : 1 = 80 ms
 Inactivity Timer(Ti) . : 250 = 30000 ms

Figure 6. 3172 ICP Configuration

4.1.5 Customization in VSE/VTAM

The following VTAM definitions in VSE VTAM are required to implement our network configuration:

- **XCA Major Node.** To define the 3172.
- **SWNET Major Node.** To define the LAN resources that VTAM can address through the 3172.
- **Mode Table entry.** To define the rules and protocols by which sessions between CICS/VSE and CICS OS/2 will be governed.
- **VTAM APPL Major Node.** To define the VTAM facilities available to our host CICS/VSE system.

4.1.5.1 VTAM XCA Major Node

An External Communication Adapter (XCA) major node is required to enable VTAM to use a 3172. Figure 7 describes our 3172 XCA major node.

- Note that the **CUADDR** matches the virtual address which we have ADDED in our VSE/ESA IPL procedure in Figure 5 on page 13.

```
*****
*          XCA MAJOR NODE                      *
*****
BOEXCA  VBUILD TYPE=XCA
*
* DEFINITION FOR 3172
*
BOE3172  PORT  CUADDR=960,          PORT ADDRESS          X
          MEDIUM=RING,            IBM TOKEN RING LAN    X
          ADAPNO=0,                RELATIVE ADAPTER NUMBER X
          TIMER=60                  (SEC) VTAM WAIT AFTER CHANNEL ACTIVATE
G3172    GROUP DIAL=YES              SWITCHED ATTACHMENT
L317201  LINE  ISTATUS=ACTIVE,CALL=INOUT,ANSWER=ON
P317201  PU    ISTATUS=ACTIVE
L317202  LINE  ISTATUS=ACTIVE,CALL=INOUT,ANSWER=ON
P317202  PU    ISTATUS=ACTIVE
L317203  LINE  ISTATUS=ACTIVE,CALL=INOUT,ANSWER=ON
P317203  PU    ISTATUS=ACTIVE
L317204  LINE  ISTATUS=ACTIVE,CALL=INOUT,ANSWER=ON
P317204  PU    ISTATUS=ACTIVE
L317205  LINE  ISTATUS=ACTIVE,CALL=INOUT,ANSWER=ON
P317205  PU    ISTATUS=ACTIVE
L317206  LINE  ISTATUS=ACTIVE,CALL=INOUT,ANSWER=ON
P317206  PU    ISTATUS=ACTIVE
L317207  LINE  ISTATUS=ACTIVE,CALL=INOUT,ANSWER=ON
P317207  PU    ISTATUS=ACTIVE
L317208  LINE  ISTATUS=ACTIVE,CALL=INOUT,ANSWER=ON
P317208  PU    ISTATUS=ACTIVE
L317209  LINE  ISTATUS=ACTIVE,CALL=INOUT,ANSWER=ON
P317209  PU    ISTATUS=ACTIVE
L31720A  LINE  ISTATUS=ACTIVE,CALL=INOUT,ANSWER=ON
P31720A  PU    ISTATUS=ACTIVE
```

Figure 7. VSE/VTAM XCA Major Node for 3172

4.1.5.2 VSE/VTAM SWNET Major Node

A SWNET major node is created to define the physical and logical units to which VTAM may communicate through the 3172 XCA major node. Figure 8 on page 17 shows the definitions we used.

Note the following parameters:

- **IDBLK** is 05D. This is the IDBLK used for a PS/2 running Comms Manager/2.
- **IDNUM** is assigned locally and will be used later in our OS/2 Comms Manager definitions.
- **PATH** The last 12 digits of the DIALNO operand contain the MACADDR of our PS/2 server.
- **LOCADDR=0** implies Independent logical unit definition.
- **DLOGMODE=LU62PS** is the name of the log mode entry which describes the session parameters for the APPC conversations between CICS OS/2 and CICS/VSE. See Figure 9 on page 18 for details of the log mode entry.

The last four LU definitions, IPFXL03A, IPFXL03B, IPFXL03C, and IPFXL03D in our example are used for 3270 sessions on the PS/2. These are not required to make our application function, but are included here as examples. 3270 sessions on our CICS OS/2 server facilitated the setup of this system.

```

*
* 3172 RELATED SWITCHED MAJOR NODE FOR PS/2 CICS/OS2 SERVER
*
BOEXCASW      VBUILD TYPE=SWNET,MAXGRP=20,MAXNO=20
*
* CICS OS/2 SERVER DEFINITION
*
IPFCPX03 PU   ADDR=01,                                X
              LANSW=YES,          LAN CAPABLE          X
              IDBLK=05D,          IDENTIFICATION BLOCK
              IDNUM=E0003,        IDENTIFICATION NUMBER
              DISCNT=NO,          VTAM DOES NOT HANG UP   X
              ISTATUS=ACTIVE,PACING=1,VPACING=2,         X
              PUTYPE=2,           SNA CLUSTER CONTROLLER X
              MAXDATA=265,        MAX NUMBER OF BYTES HANDLED BY PU X
              MAXOUT=1,           MAX PIUS IN A BATCH     X
              MAXPATH=1,          MAX NUMBER OF DIAL PATHS X
              SAPADDR=4           SERVICE ACCESS POINT ADDRESS

*
          PATH DIALNO=0104400010101003
*
IPFXL030 LU  LOCADDR=0,DLOGMOD=LU62PS,              (APPC LU) X
              ISTATUS=ACTIVE,MODETAB=CICSIPTM
IPFXL03A LU   LOCADDR=02,DLOGMOD=SP3273ES,              X
              ISTATUS=ACTIVE,MODETAB=IESINCLM,          X
              USSTAB=VTMUSSTR,MDLTAB=VTMMDL,MDLENT=VSELU2A, X
              SSCPFM=USSSCS
IPFXL03B LU   LOCADDR=03,DLOGMOD=SP3273ES,              X
              ISTATUS=ACTIVE,MODETAB=IESINCLM,          X
              USSTAB=VTMUSSTR,MDLTAB=VTMMDL,MDLENT=VSELU2A, X
              SSCPFM=USSSCS
IPFXL03C LU   LOCADDR=04,DLOGMOD=SP3273ES,              X
              ISTATUS=ACTIVE,MODETAB=IESINCLM,          X
              USSTAB=VTMUSSTR,MDLTAB=VTMMDL,MDLENT=VSELU2A, X
              SSCPFM=USSSCS
IPFXL03D LU   LOCADDR=05,DLOGMOD=SP3273ES,              X
              ISTATUS=ACTIVE,MODETAB=IESINCLM,          X
              USSTAB=VTMUSSTR,MDLTAB=VTMMDL,MDLENT=VSELU2A, X
              SSCPFM=USSSCS

```

Figure 8. VSE/VTAM SWNET Major Node for 3172

4.1.5.3 VSE/VTAM Log Mode Entry

Log mode entries are required to define valid session parameters and the rules by which LUs will communicate.

Figure 9 on page 18 shows the logmode entry we used, LU62PS, as well as the standard logmode entry, SNASVCMG, which is distributed with VSE/VTAM for LU 6.2 sessions. This latter is used automatically by CICS/VSE for session control.

In practice, CICS systems will negotiate PSERVIC values as part of the process of binding sessions. For an explanation of valid PSERVIC parameters for LU 6.2 sessions see the description of the BIND Request Unit in *SNA Network Product Formats, LY43-0081*. For an explanation of the other keywords on the MODEENT statement refer to *VTAM Resource Definition Reference, SC31-6438*.

```

CICSIPMT MODETAB
LU62PS  MODEENT  LOGMODE=LU62PS,      +
        TYPE=X'00',                    +
        FMPROF=X'13',                  +
        TSPROF=X'07',                  +
        PRIPROT=X' B0',                 +
        SECPROT=X' B0',                 +
        COMPROT=X'50B1',                +
        RUSIZES=X'8989',                +
        SSNDPAC=X'00',                  +
        SRCVPAC=X'00',                  +
        PSNDPAC=X'00',                  +
        PSERVIC=X'060200000000000002F00'
SNASVCMG MODEENT  LOGMODE=SNASVCMG,    +
        FMPROF=X'13',                  +
        TSPROF=X'07',                  +
        PRIPROT=X' B0',                 +
        SECPROT=X' B0',                 +
        COMPROT=X'78A5',                +
        RUSIZES=X'8989',                +
        SSNDPAC=X'00',                  +
        SRCVPAC=X'00',                  +
        PSNDPAC=X'00',                  +
        PSERVIC=X'0602000000000000122F00'
        MODEEND
        END

```

Figure 9. LOGMODE Entries

4.1.5.4 VTAM Application Major Node

The VTAM application major node describes, among other things, some of the characteristics of CICS/VSE. Figure 10 on page 19 shows the definitions for our CICS/VSE.

Note the following parameters:

- **PARSESS=YES.** Tells VTAM that this CICS can have parallel sessions with an Independent LU.
- **EAS=4000.** Tells VTAM that this CICS will have an estimated 4000 concurrent LU to LU sessions. This includes parallel sessions.
- **APPC=NO.** Tells VTAM that CICS does not use the VTAM APPCCMD macro instructions to converse via LU 6.2. CICS generates its own conversation streams.

BOEAPPL	VBUILD	TYPE=APPL	
DBDCCICS	APPL	AUTH=(PASS,ACQ)	
CICSSA22	APPL	AUTH=(PASS,ACQ,VPACE), PARSESS=YES,ACBNAME=CICSSA22,	C
		EAS=4000,MODETAB=CICSIPMT,APPC=NO,	C
		SONSCIP=YES,VPACING=3	
POWER	APPL	AUTH=(ACQ)	DEFAULT RJE NAME
BOEPWNJE	APPL	AUTH=(PASS,ACQ),VPACING=3,	ACB NAME FOR PNET
		MODETAB=NJEMOTB,	C
		DLOGMOD=NJEMOD	C
IESWAITT	APPL	AUTH=(NOACQ)	

Figure 10. Application Major Node

4.2 CICS/VSE Customization

This section covers the definitions in CICS/VSE to enable CICS OS/2 to communicate with CICS/VSE.

There are two ways to define CICS resources in CICS/VSE:

- Macro Definition
- Resource Definition Online (RDO) using the CEDA transaction.

In CICS/VSE, the Macro definition is used for SIT (System Initialization Table), DFHCNV (Data Conversion Table), and File Control Table (FCT).

We used CEDA to define:

- Connections
- Sessions
- Programs

4.2.1 CICS/VSE Resource Definition Using Macro

The next four subsections describe the macro resource definitions necessary to run our application. The following resources are discussed:

- DFHSIT (System Initialization Table)
- prerequisites for DFHTCP (Terminal Control Program)
- DFHCNV (Data Conversion Table)
- DFHFCT (File Control Table)

4.2.1.1 System Initialization Table (SIT)

The DFHSIT used in our implementation can be found in Figure 11 on page 20.

Critical parameters in the System Initialization Table are:

- **APPLID**. Represents this CICS. The name must match the:
 - ACBNAME operand of the APPL macro in the VSE VTAM Application Major Node
 - LU Name in the CM/2 Partner LU Definition
- **SYSDINT**. CICS System ID for this CICS. This ID must match the:

- System ID (CICS OS/2 -> TCS -> Host CICS)
- **EXEC = YES.** Supports command-level programs, so that CICS/VSE supports CICS OS/2 applications.
- **ISC = YES.** Includes the CICS intercommunication group of programs.

```

* $$ JOB JNM=DFHSITEZ,CLASS=A,DISP=D,NTFY=YES
* $$ LST CLASS=Q,DISP=H
// JOB DFHSITEZ ASSEMBLE
// LIBDEF *,CATALOG=PRD2.CONFIG
// OPTION CATAL,LIST
// EXEC ASSEMBLY
*****
*
* 5686-028 (C) COPYRIGHT IBM CORP. 1984, 1990
*
*****
      TITLE 'DFHSITEZ -- FOR CICS/OS2 TO VSE/ESA'
      PUNCH ' CATALOG DFHSITEZ.OBJ REP=YES'
      DFHSIT TYPE=CSECT,
          MXT=50,           MAX NO. OF ALL CONCURRENT TASK
          .
          .
          .
          APPLID=CICSSA22,    CICS NAMED AS CICSSA22
          .
          EXEC=YES,         EXEC LEVEL SUPPORT
          .
          FCT=EZ,          INCLUDE APPLICATION FILES
          .
          ISC=YES,         INTERSYSTEM COMMUNICATION
          .
          SYSDINT=CICS,    CICS SYS ID FOR ISC
          .
          .
          ZCP=S$,          ALL ACCESS METHODS
          DUMMY=DUMMY      TO END MACRO
      END DFHSITBA
/*
// EXEC LNKEDT
/*
/&
* $$ EOJ

```

Figure 11. DFHSIT Parameters

4.2.1.2 Terminal Control Program (DFHTCP)

The DFHSG PROGRAM=TCP macro instruction, used during CICS system generation and customization to create the terminal control program, should contain the following settings:

- **ACCMETH=VTAM.** To support VTAM
- **CHNASSY=YES.** To support SNA chain assembly
- **VTAMDEV=LUTYPE6.** To support LU6.2 Links

The supplied TCP (Suffix = S\$) in CICS/VSE was generated with these parameters. You don't have to regenerate this program.

4.2.1.3 Data Conversion Program (DFHCNV)

The job shown in Figure 12 is executed to assemble and catalog the phase for data conversion. A data conversion table is used for ASCII to EBCDIC data conversion for the COMMAREA in our sample application. You can compare this table with the record layout of the field named **WS-COMMAREA** in C.1, "Front-end Program Source (File Name TECCPROG.CCP)" on page 133. This field is what our program passes as the COMMAREA when we EXEC CICS LINK to the backend program.

```
* $$ JOB JNM=DFHCNV,CLASS=A,DISP=D
* $$ LST CLASS=Q,DISP=H
// JOB DFHCNV ASSEMBLE
// LIBDEF *,CATALOG=PRD2.CONFIG
// OPTION CATAL,LIST
// PHASE DFHCNV,*
// EXEC ASSEMBLY
*****
* 5686-028 (C) COPYRIGHT IBM CORP. 1984, 1990 *
*****
        TITLE 'DFHCNV -- USER DEFINED CONVERSION TABLE'
        DFHCNV TYPE=INITIAL
        DFHCNV TYPE=ENTRY,RTYPE=PC,RNAME=CICSVSE
        DFHCNV TYPE=SELECT,OPTION=DEFAULT
        DFHCNV TYPE=FIELD,OFFSET=0000,DATATYP=CHARACTER,          X
                DATALEN=2
        DFHCNV TYPE=FIELD,OFFSET=0002,DATATYP=BINARY,            X
                DATALEN=2
        DFHCNV TYPE=FIELD,OFFSET=0004,DATATYP=CHARACTER,          X
                DATALEN=75
        DFHCNV TYPE=FIELD,OFFSET=0079,DATATYP=BINARY,            X
                DATALEN=2
        DFHCNV TYPE=FIELD,OFFSET=0081,DATATYP=CHARACTER,          X
                DATALEN=50, LAST=YES
        DFHCNV TYPE=FINAL
        END
/*
// EXEC LNKEDT,PARM='AMODE=31,RMODE=24'
/*
/&
* $$ EOJ
```

Figure 12. DFHCNV Coding for Sample COMMAREA

A TYPE=ENTRY, RTYPE=PC, with the corresponding TYPE=SELECT and TYPE=FIELD statements, should be made for each CICS mainframe program that will be linked via DPL from CICS OS/2.

4.2.1.4 File Control Table (DFHFCT)

The statements shown in Figure 13 on page 22 are added to the CICS/VSE file control table. These statements describe our application's VSAM file and its alternate index. We used the VSE Interactive Interface to create the VSAM cluster and alternate index for our application file.

Please see C.9, "Sample VSAM Load Program" on page 159 for a sample program used to load a dummy record into the file, and C.11, "Build VSAM Alternate Index" on page 161 for an example of a job to build an alternate index.

```

* $$ JOB JNM=DFHFCTEZ,CLASS=A,DISP=D,NTFY=YES
* $$ LST CLASS=Q,DISP=H
// JOB DFHFCTEZ ASSEMBLE
// LIBDEF *,CATALOG=PRD2.CONFIG
// OPTION CATAL,LIST
// EXEC ASSEMBLY
*****
*                                     *
*   5686-028 (C) COPYRIGHT IBM CORP. 1984, 1990   *
*                                     *
*****
        TITLE 'DFHFCTEZ -- WITH CICS/OS BACKEND FILES'
        PUNCH ' CATALOG DFHFCTEZ.OBJ REP=YES'
        DFHFCT TYPE=INITIAL,SUFFIX=EZ
        SPACE 3
*-----*
*   START OF TECHDEMO ENTRIES FOR CICS/OS2 TO VSE APPL FILES   *
*-----*
TECHBAS  DFHFCT TYPE=DATASET,DATASET=TECHBAS,                *
        ACCMETH=VSAM,                                        *
        SERVREQ=(READ,UPDATE,ADD,BROWSE,DELETE),          *
        RSL=PUBLIC,                                        *
        BUFND=2,BUFNI=1,STRNO=1,                          *
        RECFORM=(VARIABLE,UNBLOCKED)                      *
TECHALT  DFHFCT TYPE=DATASET,DATASET=TECHALT,                *
        ACCMETH=VSAM,                                        *
        SERVREQ=(READ,UPDATE,ADD,BROWSE,DELETE),          *
        RSL=PUBLIC,                                        *
        BUFND=2,BUFNI=1,STRNO=1,                          *
        RECFORM=(VARIABLE,UNBLOCKED)                      *
*-----*
*           LOCAL ENTRIES SHOULD BE PLACED ABOVE THIS BOX     *
*-----*
        SPACE 3
        DFHFCT TYPE=FINAL
        END  DFHFCTBA
/*
// EXEC LNKEDT
/*
/&
* $$ EOJ

```

Figure 13. DFHFCT Entries for Application File

4.2.2 CEDA Transaction/RDO

We used the CEDA transaction to define the following resources to CICS:

- Connections
- Sessions
- Programs

All these resources are defined in the group CICSOS2 in CICS/VSE. The following command shows the contents of the group CICSOS2:

- **CEDA DISPLAY GROUP(CICSOS2)**

After defining the resources we used:

- **CEDA INSTALL GROUP(CICSOS2)**

To activate CICSOS2

- **CEDA ADD GROUP(CICSOS2) LIST(VSELIST)**

To make CICSOS2 a permanent entry in the list which is started when CICS starts.

Screen images of the CEDA sessions are reproduced in this section, to enable you to quickly reproduce the definitions in your installation.

NAME	TYPE	GROUP
CICI	CONNECTION	CICSOS2
CICI	SESSIONS	CICSOS2
CICVSE	PROGRAM	CICSOS2

Figure 14. Group Definition of CICSOS2

4.2.2.1 Connection Definition

The connection shown in Figure 15 was defined to CICS/VSE.

CEDA View	
Connection	: CICI
Group	: CICSOS2
CONNECTION IDENTIFIERS	
Netname	: IPFXLO30
INDsys	:
REMOTE ATTRIBUTES	
REMOTESystem	:
REMOTENAME	:
CONNECTION PROPERTIES	
ACcessmethod	: Vtam IRc INdirect
Protocol	: Appc Lu61
SInglesess	: No Yes
Datastream	: User 3270 SCs STRfield Lms
RECORDformat	: U Vb
OPERATIONAL PROPERTIES	
AUTOconnect	: Yes No Yes All
INSERVICE	: Yes No

Figure 15. Connection for CICS OS/2

The key parameters are:

- **CONNECTION.** The name of the connection. It must match the:
 - Local System ID in the CICS OS/2 SIT as in Figure 35 on page 42
 - CONNECTION name in CICS/VSE SESSIONS definition as in Figure 16 on page 24
- **GROUP.** The name of the group to which this definition belongs
- **NETNAME.** The name of the independent LU. It must match the:
 - LU name in the CM/2 Optional SNA Features Local LU definition as in Figure 25 on page 34
 - LU Alias in CICS OS/2's TCS entry for the host CICS as in Figure 36 on page 43
- **ACCESSMETHOD.** Defines VTAM as the access method
- **PROTOCOL.** Defines APPC to be used for CICS ISC

- **SINGLESESS.** Defines NO in order to use APPC parallel sessions

4.2.2.2 Session Definition

A session definition is used to define the logical links and the session characteristics between CICS/VSE and the independent LU (IPFXL030). Figure 16 shows the key session parameters for ILU IPFXL030, our CICS OS/2 server.

```

CEDA View
Sessions      : CICI
Group        : CICSOS2
SESSION IDENTIFIERS
Connection    : CICI
SESSName     :
NETnameq     :
MODename     : LU62PS
SESSION PROPERTIES
Protocol     : Appc                Appc | Lu61
Maximum      : 00012 , 00004      0-32767
RECEIVEPfx   :
RECEIVECount : No                No | 1-999
SENDPfx      :
SENDCount    : No                No | 1-999
SENDSize     : 04096              1-30720
RECEIVESize  : 04096              1-30720
OPERATOR DEFAULTS
OPERId       :
OPERPriority  : 000                0-255
OPERRs1     : 0
OPERSecurity : 1
USERId       :
SESSION USAGES
Transaction  :
SESSPriority : 000                0-255
OPERATIONAL PROPERTIES
Autoconnect  : Yes                No | Yes | All
INservice    :                    No | Yes
Buildchain   : Yes                Yes | No
USERArealen : 000                0-255
IOarealen   : 00000 , 00000      0-32767
RELreq       : No                No | Yes
Discreq      : Yes                No | Yes
NEPclass     : 000                0-255
RECOVERY
RECOvoption  : Sysdefault         Sysdefault | None

```

Figure 16. Session Definition for CICS OS/2 Connection

The key parameters are:

- **SESSION.** The name of the session.
- **GROUP.** The name of the group to which this definition belongs.
- **CONNECTION.** Defines the name of the connection associated with this session. It must match the:
 - CONNECTION name in CICS/VSE's CONNECTION definition as in Figure 15 on page 23
 - Local System ID in the CICS OS/2 SIT as in Figure 35 on page 42
- **MODENAME.** The logmode entry name for IPFXL030. It must match the:

- MODEENT name in the VTAM MODETAB as in Figure 9 on page 18
- **PROTOCOL.** APPC is required for CICS ISC
- **MAXIMUM.** This parameter has two values. The first value defines the maximum number of sessions supported. This MAXIMUM value specified here must match the:
 - Session Count in CICS OS/2's TCS entry for the host CICS as in Figure 36 on page 43
 - Mode Session Limit in CM/2's optional SNA Features Modes as in Figure 27 on page 36

In our application environment, a maximum of 12 sessions was ample to support three skinny clients running our application concurrently. Check CICS/VSE shutdown statistics for session usage to refine this number. The second value defines the maximum number of sessions which can be contention winners on the CICS/VSE end of the connection. The sum of this number and the maximum number of contention winners defined to CM/2 should equal the total for MAXIMUM sessions. In our case, specifying that four sessions should be contention winners at the host means we should specify that eight sessions be contention winners at the CICS OS/2 server.

- **SENDSIZE.** The RU size for sending data. This value should match the:
 - Maximum COMMAREA Size we expect to send
 - Session Buffer Size in the CICS OS/2 TCS entry for the host CICS as in Figure 36 on page 43
 - Maximum RU Size in CM/2's optional SNA Features Mode definition as in Figure 27 on page 36
 - RUSIZES in VTAM's MODEENT definition in Figure 9 on page 18

The send and receive RUSIZES for the VTAM MODEENT must be specified in a special notation. The first half byte of each byte represents the mantissa, the second represents the exponent of the number 2. So for example, a 4096 byte send length and a 640 byte receive length would be specified as RUSIZE=X'89A6'. Send being 8 times (2 to the power of 9) and receive being 10 times (2 to the power of 5). For further details, please refer to *VTAM Resource Definition Reference, SC31-6438*.
- **RECEIVESIZE.** The RU size for receiving data. This value should match the:
 - Maximum COMMAREA Size we expect to receive
 - Session Buffer Size in the CICS OS/2 TCS entry for the host CICS as in Figure 36 on page 43
 - Maximum RU Size in CM/2's optional SNA Features Mode definition as in Figure 27 on page 36
 - RUSIZES in VTAM's MODEENT definition in Figure 9 on page 18
- **AUTOCONNECT.** YES allows CICS to establish a session automatically with the session partner IPFXL030 during CICS initialization
- **BUILDCHAIN.** Enables SNA Chain Assembly
- **RELREQ.** NO means that CICS/VSE is not to release the LU upon request by another VTAM application.

- **DISCREQ.** YES means that CICS/VSE will honor disconnect requests for a VTAM device, and issue a VTAM CLSDST macro to terminate the VTAM session with that LU.

4.2.2.3 Program Definition

The program CICSVSE is the backend program for our application. It handles the VSAM file requests. This program is defined as Remote in CICS OS/2 as in Figure 38 on page 45. It is called from the front-end program running on CICS OS/2 via an EXEC CICS LINK command.

```

OBJECT CHARACTERISTICS
CEDA View
PROGRAM      : CICSVSE
Group       : CICSOS2
Language    : CObol           CObol | Assembler | C | Pl1 | Rpg
REload     : No              No | Yes
RESident   : No              No | Yes
RS1        : Public          0-24 | Public
Status     : Enabled         Enabled | Disabled
REMOTE ATTRIBUTES
REMOTESystem :
REMOTENAME   :
Transid     :
Executionset : Fullapi       Fullapi | Dplsubset

```

Figure 17. Application Program Definition

Chapter 5. Define and Customize the CICS OS/2 Systems

This chapter describes the steps required to prepare the CICS OS/2 environment to run our distributed application.

The chapter is divided into three sections:

- LAN Adapter Protocol Support (LAPS) Installation and Customization
- OS/2 Communication Manager Customization
- CICS OS/2 Installation and Customization

The installation of the IBM Communication Manager for OS/2 is not covered because it is a standard installation and you can follow the steps in the appropriate books.

5.1 LAPS Installation and Customization

Before CICS OS/2 can communicate over the Token-Ring to either a client or our host CICS/VSE system, the LAPS program must be installed and customized. This is the program that physically manages the PS/2 LAN adapter.

To install LAPS from the product diskette, do the following:

1. Put the LAPS diskette into your diskette drive (for example A)
2. At the OS/2 prompt, type: **A:laps** and press the Enter key.

The window in Figure 18 on page 28 lists the features which we selected for our CICS OS/2 server machine.

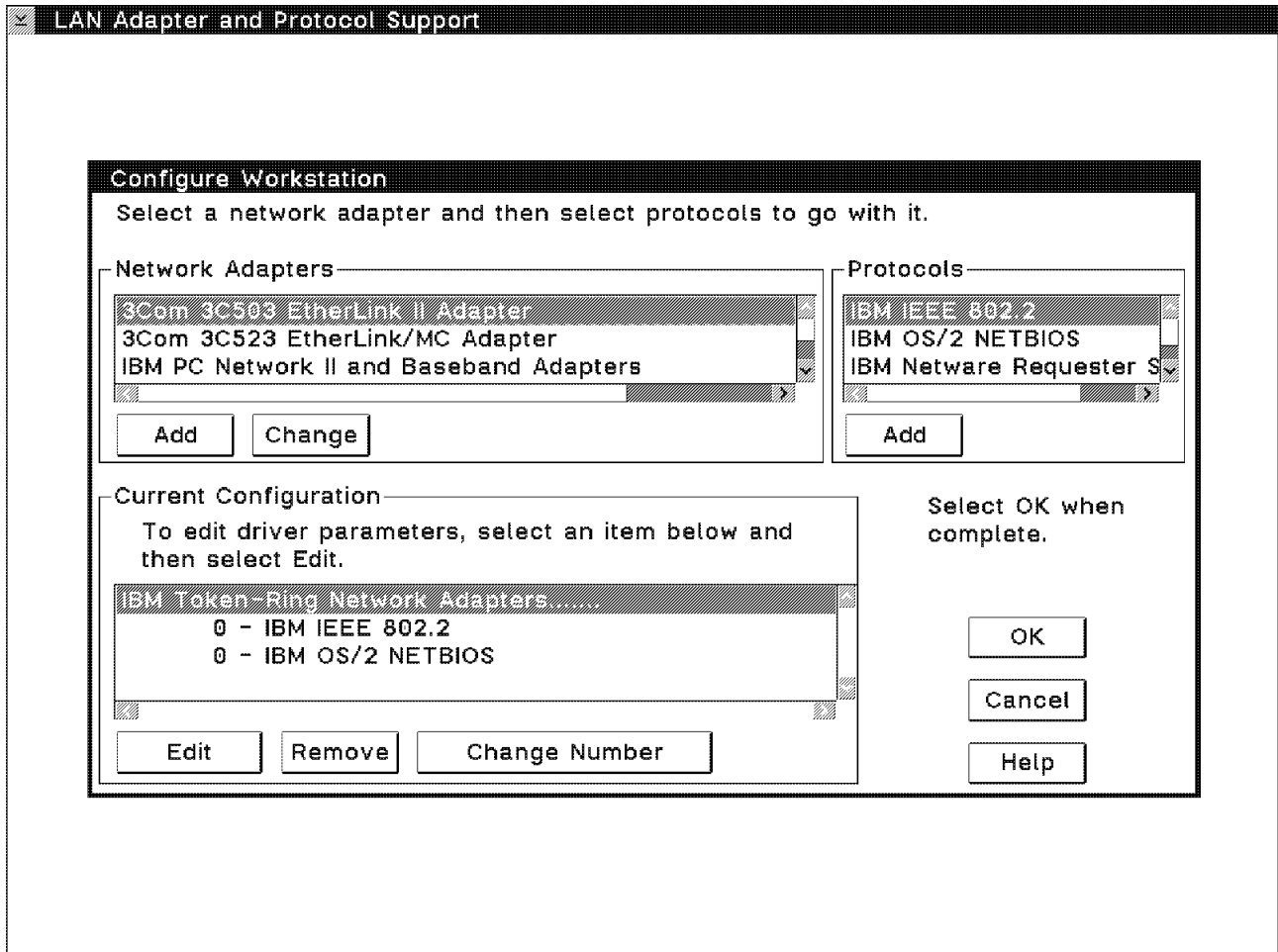


Figure 18. LAPS Configuration Menu for the CICS OS/2 Server

In this panel you have to select the Network Adapter (refer to the PS/2 hardware setup) and then select one, or more, protocols to go with it. For our installation we selected the IBM Token-Ring Network Adapters with two protocols, IBM IEEE 802.2 and IBM OS/2 NETBIOS. The IBM IEEE 802.2 protocol allows us to communicate with our host CICS/VSE using APPC over the Token-Ring. The NETBIOS protocol allows us to communicate with our skinny client systems over the Token-Ring. After the selection is made (clicking the item with mouse pointer and "pushing" the Add button), you must edit the driver parameter. The following window will be displayed:

Parameter	Value
Early release	NO
Adapter Mode	PRIMARY
Network adapter address	400010101003
Shared RAM address	
Maximum number of queued transmits	12
Number of receive buffers	2
Receive buffer size	256
Number of adapter transmit buffers	1
Transmit buffer size	

Figure 19. LAPS Token-Ring Adapter Customization

The only requested parameter to set up is the **Network Adapter Address** (for the others, the default values are good). This field specifies the locally administered address of the PS/2 Token-Ring adapter and **must match** the last six bytes of the **PATH** macro's **DIALNO** parameter in the VTAM Switched major node described in Figure 8 on page 17.

- 400010101003 is the Token-Ring MACADDRESS assigned to our CICS OS/2 server PS/2.

After you have saved all your responses, re-boot the PS/2.

If you wish to change the definitions of LAPS again, you have to invoke LAPS.EXE which can be found in the C:IBMCOM directory.

5.2 Communication Manager Customization

This section contains guidance information about the Communication Manager definitions that are important when configuring CICS OS/2 intercommunication.

For APPC (LU6.2) the following definitions are required in the CM configuration files:

- One **DLC Adapter Parameters** definition, which specifies Data Link Control characteristics of the Token-Ring adapter.
- One **Local Node Characteristics** definition, which specifies characteristics that are common to all APPC users on the workstation.
- At least one **Connection** definition.
- A **Local Logical Unit** definition for each LU alias specified in a TCS entry that defines a remote CICS system to CICS OS/2.

- A **Partner Logical Unit** definition for each remote CICS system.
- One or more **MODE** definitions to specify sets of session properties that are used in binding APPC sessions.
- A **Transaction program** definition for every local transaction that can be invoked in an inbound request from a remote system.

In addition to the customization required to support the communication links, the Workstation Profile, Translation Table File Name defaults to **ACSGTAB.DAT**. This table enables EBCDIC to ASCII conversion. It is a sample conversion table and is supplied with the Communication Manager in CMLIB. If ACSGTAB.DAT is inappropriate, you may define your own table and specify the Translation Table File Name, in the name field of the **Change Workstation Information** panel (CM/2 Configuration definitions -> Options -> Change Workstation Information).

The following are guidelines for the relevant parameters in the Communications Manager Profile List Sheet in the C/M configuration definition. They describe parameters appropriate for defining APPC APIs on a Token-Ring.

5.2.1 Configure DLC Token-Ring

Token Ring or Other LAN Types DLC Adapter Parameters

Adapter

Free unused links

Send alert for beaconing

Maximum number of link stations (1 - 255)

Maximum I-field size (265 - 16393)

Percent of incoming calls (%) (0 - 100)

C&SM LAN ID

Connection network name (optional) .

Window count

Send window count (1 - 8)

Receive window count (1 - 8)

Figure 20. Token-Ring DLC Adapter Parameters

You can take most defaults here. You should enter a name for the following parameter:

- **C&SM LAN ID**. To identify the ring - used by system management products.

5.2.2 Defining Local Node Characteristics

The screenshot shows a dialog box titled "Local Node Characteristics". It contains the following fields and options:

- Network ID: DEIBMIPF
- Local node name: CICSSRVR
- Node type section with three radio buttons:
 - End node to network node server
 - End node - no network node server
 - Network node
- Your network node server address (hex): [Empty text box]
- Local node ID (hex): 05D E0003
- Buttons: OK, Options..., NetWare(R)..., Cancel, Help

Figure 21. Local Node Characteristics

The key fields are:

- **Network ID.** DEIBMIPF is the name of the network to which this PU belongs. This must match the NETID parameter of the VTAM start list. See A.2, "VTAM Start List" on page 116.
- **Node type.** Our PS/2 is an end node accessed by the host through a 3172 gateway. Our 3172 gateway is **not** configured as a network node server.
- **Local node ID.** The values 05D E0003 are used for the XID exchange. They match IDBLK and IDNUM values specified on our PU definition statement in the VTAM Switched Network major node as in Figure 8 on page 17.

Local Node Options

The screenshot shows a dialog box titled "Local Node Options". It contains the following fields and options:

- Local node alias name: CICSSRVR
- Optional comment: [Empty text box]
- Activate Attach Manager at start up
- Buttons: OK, Cancel, Help

Figure 22. Local Node Options Alias Name

The key field is:

- **Activate Attach Manager.** This is a feature of APPC which manages incoming requests to begin a communication session. In our case, the Attach Manager is started automatically when the Communications Manager is started.

5.2.3 Define SNA Host Connections

Change a Connection to a Host

Link name: HOST0001

LAN destination address (hex): 400020201003

Partner network ID: DEIBMIPF

Partner node name: CICSSA22 (Required for partner LU definition)

Local PU name: CICSSRVR

Node ID (hex): 05D E0003

Use this host connection as your focal point support

APPN support

Optional comment: [Empty text area]

Buttons: OK, Define Partner LUs..., Cancel, Help

Figure 23. Connection Definition Panel

The key fields are:

- **Link name.** HOST0001 is the link name of the connection to the host.
- **LAN destination address.** 400020201003 is the LAN MACADDRESS of the 3172 gateway to our host as specified in the 3172 ICP configuration in Figure 6 on page 14.
- **Partner network ID.** DEIBMIPF is the NETID of the host VTAM as specified in the VTAM start list.
- **Partner node name.** CICSSA22 is the VTAM APPLID of our host CICS. It is specified in the VTAM APPL major node and in the CICS/VSE SIT.
- **Node ID.** 05D E0003 are the IDBLK and IDNUM parameters as specified in the VTAM Switched major node.

5.2.4 Defining Optional SNA Features

Figure 24 on page 33 lists the features which may be included in the local configuration of Comms Manager.

On our CICS OS/2 server we defined options for the first four features:

- Local LUs
- Partner LUs
- Modes
- Transaction program definitions

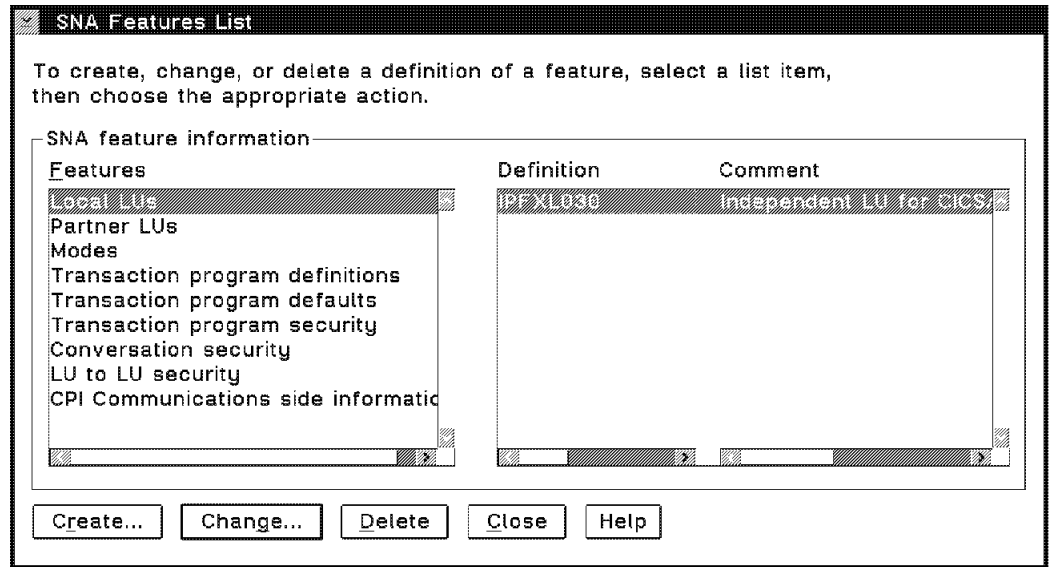


Figure 24. List of Optional SNA Features

Local LU Definition

Change a Local LU

LU name

Alias

NAU address

Independent LU

Dependent LU NAU (1 - 254)

Host link

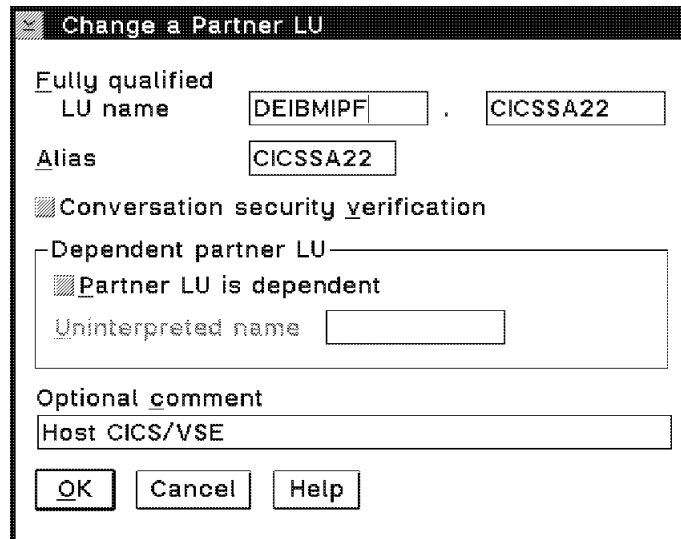
Optional comment

Figure 25. Local LU Definition Panel

The key fields are:

- **LU name.** IPFXL030 is the name by which the Logical Unit is known throughout the SNA network. It is defined in the VTAM switched network major node in Figure 8 on page 17.
- **Alias.** IPFXL030 is the alias name by which transaction programs may refer to the LU.
- **IPFXL030.** Defined as an independent LU. For more information about independent LUs please refer to Chapter 4, "Define and Customize the VSE/ESA Host" on page 11.

5.2.5 Defining the Partner LU



Change a Partner LU

Fully qualified
LU name DEIBMIPF . CICSSA22

Alias CICSSA22

Conversation security verification

Dependent partner LU

Partner LU is dependent

Uninterpreted name

Optional comment
Host CICS/VSE

OK Cancel Help

Figure 26. Partner LU Definition

The key fields are:

- **Fully qualified LU name.** The network ID and the partner LU name.
DEIBMIPF is the ID of the network on which the partner LU resides.
CICSSA22 is the name of the partner LU residing on the network whose name is in the previous field.
- **Alias.** CICSSA22 is the alternative name or alias name by which transaction programs may refer to the LU.

5.2.6 Defining the Logon Mode

Change a Mode Definition

Mode name LU62PS

Class of service #INTER

Mode session limit 12 (0 - 32767)

Minimum contention winners 8 (0 - 32767)

Receive pacing window 8 (0 - 63)

RU size

Default RU size

Maximum RU size 4096 (256 - 16384)

Optional comment

Mode for CICS/OS2

OK Cancel Help

Figure 27. Mode Definition Panel

The key fields are:

- **Mode name.** LU62PS is the name of the mode that contains the session properties for the LU6.2 session. It must match the Mode Name in the TCS of CICS OS/2 which describes our host CICS/VSE as in Figure 36 on page 43.
- **Class of service.** #inter is for interactive communications.
- **Mode session limit.** The maximum number of sessions that can be active at the same time for the LUs using this mode.
- **Minimum contention winners.** The minimum number of sessions in which you want a logical unit (LU) in this mode to win in a contention with a partner LU. We have specified eight winners for the PS2 end, and in the host CICS/VSE session definitions in Figure 16 on page 24, we have specified four winners for the host.
- **Receive pacing window.** The receiving node rate sets the pace for message transmission.
- **Maximum RU size.** The largest RU size we expect to be transferred on sessions using this mode. This number matches the maximum RUSIZES value specified in the VTAM MODENT statement for mode entry LU62PS in Figure 9 on page 18. It also matches the Session Buffer Size in our CICS OS/2 TCS entry which describes the CICS/VSE host in Figure 36 on page 43.

5.2.7 Defining Transaction Programs

Selecting Transaction program definitions from the SNA Features List allows you to define transaction programs and to see what has already been defined. When Communications Manager receives a request from a remote system to start a transaction, it must be told via these menus which program to start locally. CICS OS/2 supplies an OS/2 program called FAACLPIN.EXE for this purpose. Comms Manager can also pass a parameter string to this called program. Thus Comms Manager can pass the name of the CICS OS/2 transaction to the FAACLPIN

program. FAACLPIN can then call CICS OS/2 and pass the name of the transaction to be executed.

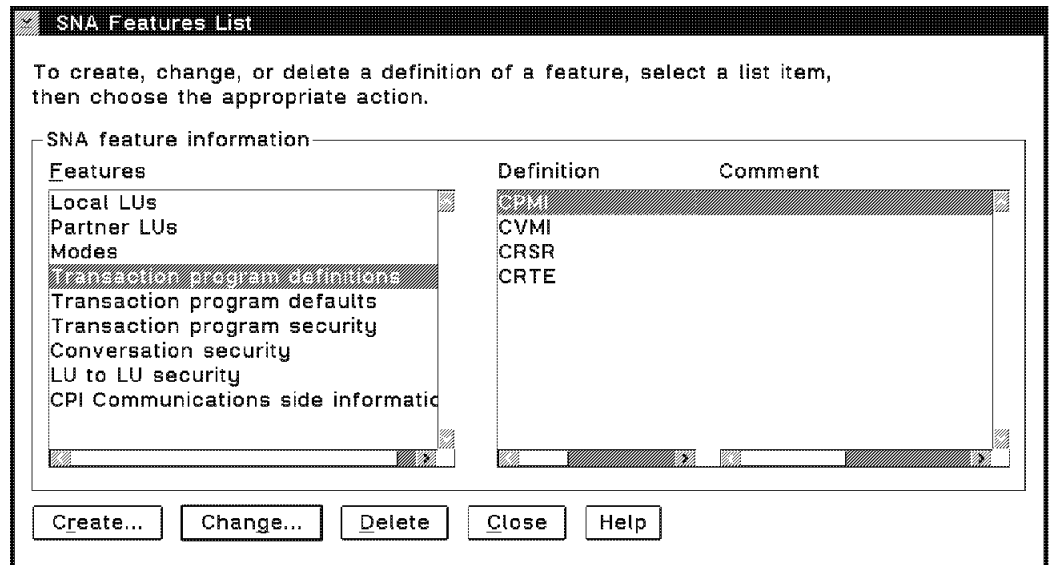


Figure 28. Transaction Program Definition Main Menu

CPMI Transaction Program

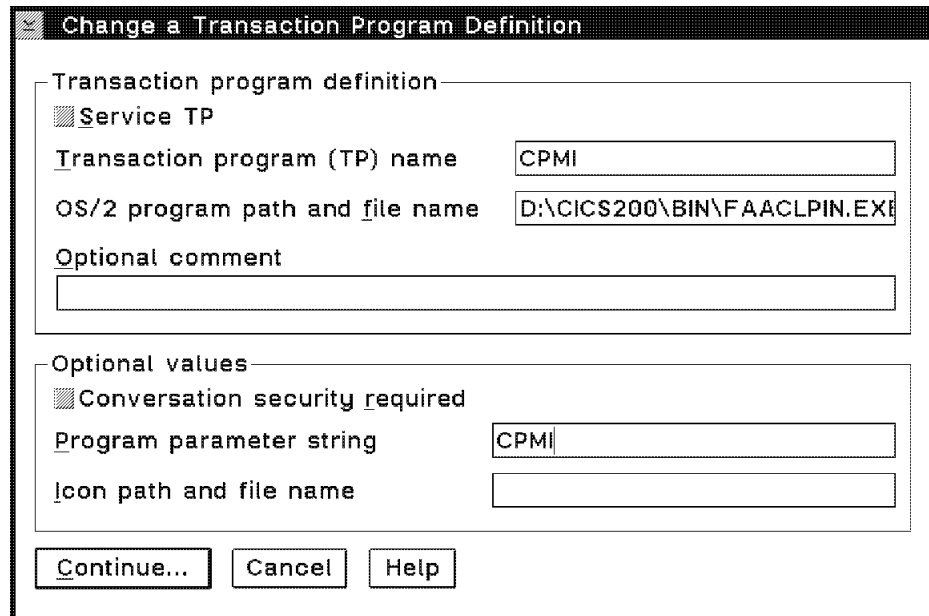


Figure 29. CPMI Transaction Program Definition

The CPMI mirror transaction deals with incoming function shipping from remote CICS systems. It recreates the original request, issues it on the local system and passes the acquired data back to the remote system.

The key fields are:

- **Transaction program (TP) definition.** CPMI is the name of the transaction which Comms Manager may receive.
- **OS/2 program path and file name.** C:\CICS200\BIN defines the path to:

- **FAACLPI~~N~~.EXE**. The OS/2 program which Comms Manager starts to pass the transaction request to CICS OS/2. FAACLPI~~N~~.EXE passes on the request to CICS OS/2 for the appropriate CICS transaction.
- **Program parameter string**. This field is **case sensitive**. CPMI is the parameter passed to the OS/2 program FAACLPI~~N~~ which in turn passes it to CICS OS/2.

Additional TP Parameters for CPMI

Figure 30. CPMI Transaction Program Definition (Additional TP Parameters)

The key fields are:

- **Background**. Defines that the transaction program runs in background.
- **Non-queued, Attach Manager started**. Defines that multiple instances of the transaction can run at one time. This is the default and the recommended option.

CVMI Transaction Program

Figure 31. CVMI Transaction Program Definition

The CVMI mirror transaction deals with incoming function shipping from remote CICS systems. It recreates the original request, issues it on the local system and passes the acquired data back to the remote system. The CVMI mirror transaction is for requests from host CICS systems.

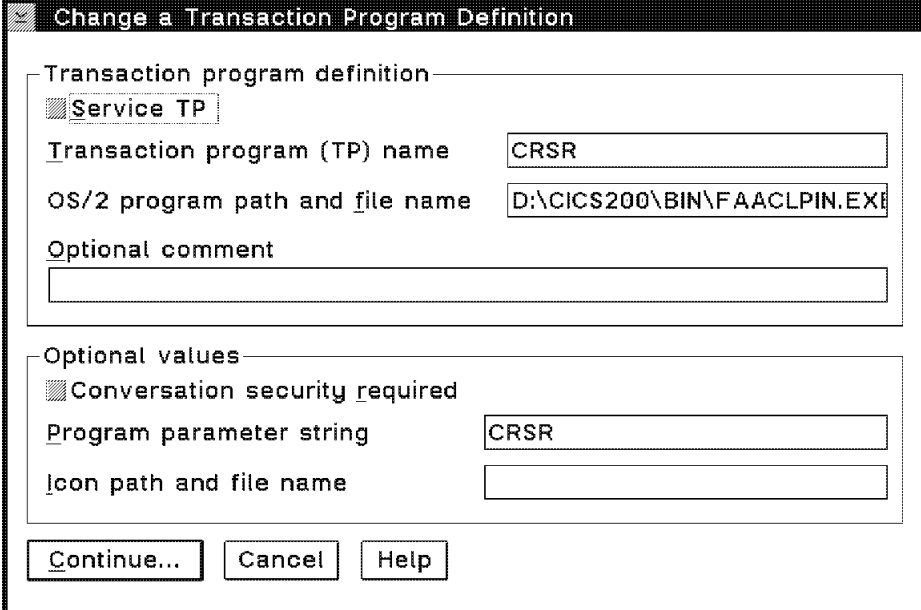
The key fields are:

- **Transaction program (TP) name.** CVMI is the name of the transaction
- **OS/2 program path and file name.** C:\CICS200BIN defines the path to:
- **FAACLPIN.EXE.** The OS/2 program to be started for CICS OS/2. FAACLPIN.EXE passes on any request to the appropriate CICS component.
- **Program parameter string.** This field is **case sensitive**. CVMI is the parameter to be passed by FAACLPIN to CICS OS/2 when invoked.

Additional TP Parameters for CVMI

- The parameters are the same as shown in Figure 30 on page 38.

CRSR Transaction Program



The screenshot shows a dialog box titled "Change a Transaction Program Definition". It is divided into two main sections: "Transaction program definition" and "Optional values".

Transaction program definition:

- Service TP
- Transaction program (TP) name:
- OS/2 program path and file name:
- Optional comment:

Optional values:

- Conversation security required
- Program parameter string:
- Icon path and file name:

At the bottom of the dialog are three buttons: "Continue...", "Cancel", and "Help".

Figure 32. CRSR Transaction Program Definition

The purpose of the CRSR transaction is to delete all shipped terminal definitions on remote systems.

The key fields are:

- **Transaction program (TP) name.** CRSR is the name of the transaction
- **OS/2 program path and file name.** C:\CICS200BIN defines the path to:
- **FAACLPIN.EXE.** The OS/2 program to be started for CICS OS/2. FAACLPIN.EXE passes on any request to the appropriate CICS component.
- **Program parameter string.** This field is **case sensitive**. CRSR is the parameter to be passed by FAACLPIN to CICS OS/2 when invoked.

Additional TP Parameters for CRSR

- The parameters are the same as shown in Figure 30 on page 38.

CRTE Transaction Program

Figure 33. CRTE Transaction Program Definition

The CRTE transaction is for transaction routing. We used CRTE to test that the connection was properly established between CICS/VSE and CICS OS/2 in order to run a transaction in the remote CICS.

CRTE can be invoked from the host CICS or the CICS OS/2 side.

- **Transaction program (TP) name.** CRTE is the name of the transaction
- **OS/2 program path and file name.** C:\CICS200\BIN defines the path to:
- **FAACLPIN.EXE.** The OS/2 program to be started for CICS OS/2. FAACLPIN.EXE passes on any request to the appropriate CICS component.
- **Program parameter string.** This field is **case sensitive**. CRTE is the parameter to be passed by FAACLPIN to CICS OS/2 when invoked.

Additional TP Parameters for CRTE

- The parameters are the same as shown in Figure 30 on page 38.

5.3 CICS OS/2 Server Customization

CICS OS/2 must be properly customized. In our application, a CICS OS/2 program will link to a program on the mainframe CICS/VSE system. To cause CICS OS/2 to dynamically link to a host program, the program must be defined as remote to CICS OS/2.

The CEDA transaction must be used to update the following CICS OS/2 tables in order for our application to function on two platforms:

- System Initialization Table (SIT)
- Terminal Control Table System Entry (TCS)

- Program Control Table (PCT)
- Processing Program Table (PPT)

5.3.1 System Initialization Table (SIT)

The SIT supplies the information required to initialize our system. The SIT holds initial conditions for system startup including the initial transaction, the size of storage areas, and so on as shown in Figure 34 and Figure 35 on page 42.

FAASIT2		System Initialization Table-1										More : +
Group Name :		FAASYS										
Description :		SAMPLE SYSTEM INITIALIZATION										
System Sizes												
CWA size :		0										
Maximum TWA size :		0										
Number of trace entries . . :		200										
Task Control												
Maximum number of tasks . . :		12										(1-99)
Minimum free tasks :		10										(0-99)
Task Classes :		1	2	3	4	5	6	7	8	9	10	
Maximum tasks in Class . . :		1	1	1	1	1	1	1	1	1	1	(1-99)
Default Process Priority . . :		86										(0-255)
CICS System Priority :		0										(0-255)

Figure 34. System Initialization Table-1

The key fields are:

- **Maximum number of tasks** should be equal to minimum free tasks plus the number of the CICS OS/2 video terminals. The default is two video terminals.
- **Minimum free tasks** should be equal to the number of concurrent nonfacility tasks. These are tasks which do not use permanent terminals as their principal facility. Client terminals are considered nonfacility tasks.

Regarding the other values in the System Initialization Table-1, the default values are acceptable for our purposes.

FAASIT3		System Initialization Table-2		More : -
Group Name	:	FAASYS		
System Communications				
Local System ID.	:	CICI		
Local System Appl ID	:	CICSITSO		
NETBIOS Listener Adapter . .	:	0	(0, 1 or B)	
Maximum NETBIOS Systems. . .	:	10	(0-254)	
Default Remote System ID . .	:	CICS		
Miscellaneous				
Load PNA Support	:	N	(Y or N)	
PNA Model Terminal	:	MPNA		
Initial Transaction ID	:	CLOG		
Dump on Abend.	:	N	(Y or N)	
Date Format.	:	MMDDYY	(ddmmyy,mmddy,ymmdd)	
External File Manager Name :				
User Conversion Table. . . .	:			

Figure 35. System Initialization Table-2

The key fields are:

- **Local System ID.** CICI is the local name of the workstation. It must match the:
 - **Connection Name** in the CICS/VSE Connection definition in Figure 15 on page 23.
 - **Sessions and Connection Name** in the CICS/VSE Sessions Definition in Figure 16 on page 24.
- **Local System Appl ID.** CICSITSO is the Appl ID which NETBIOS uses to listen for skinny clients. Each client will be customized to talk to this Appl ID via an entry in the client’s FAARQ.INI file as in 5.4.3, “Tailoring the Client Configuration” on page 46.
- **NETBIOS Listener Adapter.** The adapter number used by both NETBIOS and IEEE 802.2 protocols specified in LAPS configuration as in Figure 18 on page 28.
- **Maximum NETBIOS Systems.** Maximum number of skinny clients we expect ever to have.
- **Default Remote System ID.** If a CICS OS/2 server terminal or client issues a transaction which is not defined locally, CICS OS/2 will attempt to **transaction route** it to the system specified by this parameter. **CICS**, in our case, is the name for our host CICS/VSE system specified by the SYSIDNT parameter in the CICS/VSE SIT as in Figure 11 on page 20 and System ID in the CICS OS/2 TCS as in Figure 36 on page 43.

5.3.2 TCS Terminal Control Table (System Entry)

CICS OS/2 requires a terminal control table system entry (TCS) for any remote CICS system with which it may communicate. The following entry describes our CICS/VSE host from CICS OS/2’s point of view.

FAATCS2		Terminal Control Table (System Entry)-1	
System ID.	:	CICS	
Group Name	:	ITSO	
Connection Type.	:	APPC	(APPC or NETB)
Connection Priority.	:	086	(0-255)
Description.	:	CICS/VSE	CICSSA22
Session Details			
Session Count.	:	12	(1-99)
Session Buffer Size.	:	04096	(512-40000)
Attach Security.	:	L	(L=Local, V=Verify)
Partner Code Page.	:	00037	
APPC Details			
Mode name.	:	LU62PS	
LU alias	:	IPFXL030	
Partner LU Alias	:	CICSSA22	
NETBIOS Details			
NETBIOS Adapter.	:		(0 or 1)
Remote System Appl ID.	:		

Figure 36. Terminal Control Table (System Entry)

The key fields are:

- **System ID.** CICS is the name of the remote system with which CICS OS/2 will communicate.
This must match the SYSIDNT field in the SIT of the CICS/VSE host system as in Figure 11 on page 20.
- **Group Name.** ITSO is the name of the group to which this entry belongs.
- **Connection Type.** APPC for an LU6.2 connection. NETB would be valid only when describing a connection to another CICS OS/2 server.
- **Connection Priority.** 86 is the default.
- **Session Count.** The maximum number of concurrent sessions that can be active. This value should match the following parameters:
 - **Mode Session Limit** in CM/2's Mode definition Figure 27 on page 36.
 - **Maximum(first parameter)** in CICS/VSE's Session Definition Figure 16 on page 24.
- **Session Buffer Size.** 4096 bytes is the maximum buffer size for each session. Session buffer size should be the largest COMMAREA size we expect to send or receive in our sample application.
This value should match the:
 - **Sendsize and Receivesize** in CICS/VSE Session definitions in Figure 16 on page 24.
 - **Max RU Size** in CM/2's Mode definition as in Figure 27 on page 36.
 - **RUSIZES** in VTAM's MODEENT for LU62PS as in Figure 9 on page 18.
- **Attach Security.** L means Local security. User ID and password are not passed to the remote system. It assumes that the user is authorized.
- **Partner Code Page.** 37 is for U.S. EBCDIC used by our CICS/VSE host.

- **Mode Name.** LU62PS is the mode name that contains the session properties for the LU6.2 sessions. It matches an entry in CM/2's optional SNA features modes list as in Figure 27 on page 36.
- **LU alias.** The Alias of the local LU defined to the Communication Manager. Please refer to Figure 25 on page 34. It is used by CICS OS/2 instead of the real local LU name.
- **Partner LU Alias.** The name is used by CICS OS/2 instead of the fully qualified partner LU name to refer to the remote CICS. CICS OS/2 uses this name to obtain the SNA Network Name of the remote system. All Inter-System Communications take place between the local LU specified by the LU Alias, and the Partner LU specified by this Partner LU Alias. This name must match the:
 - **Alias.** in OS/2 Communication Manager's Partner LU definition as in Figure 26 on page 35.
- **NETBIOS Details.** These details are left blank because the connection we are describing with this TCS entry is an APPC connection, not NETBIOS.

5.3.3 Program Control Table (PCT)

The PCT holds control information used by CICS OS/2 to identify and initialize transactions. Transactions can be run locally or at a remote CICS system by specifying a system ID. The following PCT entry describes the transaction which we used in our sample application.

FAAPCT2	Program Control Table-1	
Transaction Code	TECH	
Group Name	CREDIT	
Program Name	TECCPROG	
Task Class	N	(1-10 or N)
Secure	N	(Y or N)
Can Be Purged.	Y	(Y or N)
Use Alternate Screen Size. . .	N	(Y or N)
Priority	0	(0-255)
System ID.		
Remote Transaction Code. . .		
Description.	CREDIT DEMO TRANSACTION	

Figure 37. PCT Program Control Table for TECH

The key fields are:

- **Transaction Code.** TECH is the transaction name for our application.
- **Group Name.** CREDIT is the name of the group to which the TECH transaction belongs.
- **Program Name.** TECCPROG is the first program that will be invoked by the TECH transaction. Note that this program is not described by any local PPT entry. In this case CICS OS/2 assumes the program resides locally. In fact,

we have compiled and linked it into the CICS200BIN directory as TECCPROG.DLL.

5.3.4 Processing Program Table (PPT)

Programs and mapsets to be used in CICS OS/2 do not necessarily need an entry in the Processing Program Table. If a program is called, for which there is no PPT entry, then CICS OS/2 tries to locate an executable version of it in an OS/2 directory. CICS OS/2 may have access to this directory via the OS/2 **LIBPATH** statement in CONFIG.SYS. If an executable version of the program is found, then it is loaded as if there had been a PPT entry. Similarly, when a CICS INQUIRE PROGRAM STATUS command is issued for a program which is not described by a PPT entry, CICS OS/2 tries to locate it. If it finds an executable version of it locally, CICS OS/2 returns a status of ENABLED.

In our application example, the program called TECCPROG, specified in the PCT in Figure 37 on page 44 has no corresponding PPT entry on CICS OS/2. The fact that we have created an executable version of the program called TECCPROG.DLL in the CICS200BIN directory is enough for CICS OS/2 to find and load it. However, this TECCPROG will attempt to link to another program called **CICSVSE** (our backend). This CICSVSE program is located on our host CICS/VSE system. CICS OS/2 requires an entry in its PPT to describe each remote program.

The following panel shows the CICS OS/2 PPT entry for our host backend application program **CICSVSE**.

CICSVSE is defined as Remote and resides in CICS/VSE.

```

FAAPPT2                Processing Program Table-1

  Program Name . . . . . : CICSVSE
  Group Name . . . . . : TECHDEMO
  Program(P)/Map(M)/Data(D). . . : P                (P, M, or D)
  Resident(P,T,N)/Remote(R). . . : R                (P, T, N, or R)
  Remote (R) SYSID . . . . . : CICS                (Remote only)
  Remote (R) PROGRAM . . . . . :                   (Remote only)
  Remote (R) TRANSID . . . . . :                   (Remote only)

  Description. . . . . : CICSVSE BACKEND FOR TECCPROG

```

Figure 38. Remote Program Entry for Backend Program

Note that the:

- **Remote (R) SYSID** is the name of a valid TCS entry as in Figure 36 on page 43.

5.4 CICS OS/2 Client Customization

CICS OS/2 client installation and customization is a relatively simple process which entails the following steps:

- Enabling NETBIOS
- Copying the client software to the client workstation
- Tailoring the client configuration file
- Creating folders for the client control panel and Transaction Routing (TR) emulator

5.4.1 Enabling NETBIOS

In our configuration two clients were installed. Both were OS/2 Version 2.1 with Communications Manager/2. We installed the LAN Adapter and Protocol Support (LAPS) program. Please see 5.1, "LAPS Installation and Customization" on page 27 for details of LAPS installation.

5.4.2 Install Client Software

The client software for OS/2 clients is located in the CICS200CLIENTSOS2 directory on the CICS OS/2 server.

- The contents of this directory were copied to a diskette.
- A directory with this name was created on each client.
- The diskette contents were then copied to each client machine's directory.
- Each client workstation's CONFIG.SYS file was updated by adding the CICS200CLIENTSOS2 directory name to the **SET PATH** and **LIBPATH** statements.

5.4.3 Tailoring the Client Configuration

The file named FAARQ.INI was edited. The **ServerName =** statement was changed from CICSOS2 to **CICSITSO** to reflect the name of our CICS OS/2 server. This was specified in our server's SIT entry as the **Local System Appl ID** in Figure 35 on page 42.

The FAARQ.INI statements were compiled to create the FAARQ.SYS file using the following command:

```
FAARQPO1
```

5.4.4 Folders for Client Control and TR Emulator

To use CICS OS/2 from a client workstation the following must be done:

1. A NETBIOS session must be established between the client and server machines. For OS/2 clients, this is done by invoking the Client Control Status panel with the program named FAARQPOP.EXE.
2. A Transaction Routing (TR) emulator must be started on the NETBIOS session using the program FAARQPOT.EXE.

We used icons in the OS/2 Desktop to call each of these programs. Two empty Program Templates were dragged to the desktop and the following were set in the notebooks:

- **Client Control Panel** notebook contained:

- C:CICS200CLIENTSOS2FAARQPOP.EXE for **Path and file name** and
- C:CICS200CLIENTSOS2 for **Working Directory**
- **TR Emulator** notebook contained:
 - C:CICS200CLIENTSOS2FAARQPOT.EXE for **Path and file name** and
 - C:CICS200CLIENTSOS2 for **Working Directory**

Chapter 6. Overview of our Sample Application

Our BMS based sample application has four major components:

- A front-end program called TECCPROG which runs on CICS OS/2
- A set of BMS maps used on CICS OS/2 by the front-end
- A backend program called CICSVSE which runs on a host CICS/VSE system
- A VSAM file and alternate index which reside on the host CICS/VSE system and are accessed by the backend program

The front-end program gathers information from the user about the function he wants to perform. BMS is used to interface with the user's screen. The results of this interaction are used to create a COMMAREA. The front-end program then issues an EXEC CICS LINK with a COMMAREA to the backend program, that is DPL is used to communicate with CICS/VSE.

The backend performs the appropriate I/O request on the VSAM file, puts the data in the COMMAREA and RETURNS control to the front-end.

With BMS, the front-end then displays the data returned in the COMMAREA.

We tested an alternative front-end program which was written to use OS/2 Presentation Manager. See chapter 6.3, "Front-end Application Implementation on OS/2 (PM based)" on page 52 for comments on its implementation.

6.1 Front-end Application Implementation on CICS OS/2 (BMS based)

After having set up the environment described in the preceding chapters, the following steps were executed on the CICS OS/2 system to implement our application:

- Install the MicroFocus COBOL Compiler
- Translate BMS maps
- Translate, compile and link the front-end

6.1.1 Install MicroFocus COBOL

The MicroFocus COBOL compiler can be ordered separately or as part of the MicroFocus COBOL Workbench or Toolset products. We installed the Workbench on our PS/2. Tools such as the Animator can be very valuable in a development environment.

6.1.2 Translate BMS Maps

The source for the BMS maps used by our application was placed in a file called **NEW1MS.MAP** in an OS/2 directory called **CREDIT** on our PS/2's **D:** disk. This mapset, documented in C.4, "BMS Maps (File Name NEW1MS.MAP)" on page 152, was translated with the following command:

- **CICSMAP D:CREDITNEW1MS.MAP**

This command produces two important outputs:

- A mapset copybook file called **NEW1MS.CBL** (included later when we compile our front-end).
- The executable maps in the CICS OS/2 user mapset master file (FAAMSFSC.BTR).

6.1.3 Translate, Compile and Link Front-end

The source for the front-end application program was placed in a file called **TECCPROG.CCP** in an OS/2 directory called **CREDIT** on our PS/2's **D:** disk.

This program, shown in C.1, "Front-end Program Source (File Name TECCPROG.CCP)" on page 133, tries to determine which backend programs are available. In the environment we created, the only available backend is the CICSVSE program. Our CICS OS/2 PPT had an entry describing CICSVSE as remote in Figure 38 on page 45.

The front-end program was translated, compiled and linked with the following command:

- **CICSTCL D:CREDITTECCPROG.CCP**

During the LINK step of the CICSTCL procedure we were asked to 'Enter new file spec' for the DOSCALLS.lib.
Respond with **C:OS2DOSCALLS**

COBOL directives tell the compiler how to handle a source program. The directives used for this compile are those described in C.5, "COBOL Directives (File COBOL.DIR)" on page 157. Note that the compiler locates this directives file via the OS/2 environment variable called COBDIR. COBDIR is set to the proper directory by modifying the distributed CICSENV.COMD file as illustrated in C.6, "CICS OS/2 Tailor Environment Variables (File CICSENV.COMD)" on page 158.

6.2 Application Implementation on CICS/VSE

On the CICS/VSE host system, the following activities were required to implement our application in addition to the setup activity described in the preceding chapters:

- Translate, compile and link the backend
- Create the VSAM base file
- Create the VSAM alternate index

6.2.1 Translate, Compile and Link Backend

The source code for the backend program was put in an ICCF member called **CICSVSE** which you can review in C.2, "Backend Program Source (Program CICSVSE)" on page 145. The VSE Interactive Interface was used to create the job stream in C.3, "Compile Backend Program Job" on page 151 which translates, compiles and link-edits program CICSVSE.

This program has been identified to CICS/VSE via the CEDA transaction as in Figure 17 on page 26 and to CICS OS/2 via the PPT entry in Figure 38 on page 45.

6.2.2 Create VSAM Base File

The VSE Interactive Interface was also used to create the VSAM base file. Refer to the **SERVER-CUSTOMER-FIELD** in the CICS/VSE program source in C.2, “Backend Program Source (Program CICSVSE)” on page 145 for the record layout. The file should be defined as key sequenced. The primary key field is ACCOUNTO which starts in position 0 for a length of five bytes.

This file is identified to CICS/VSE via the **TECHBAS** entry in the File Control Table as in Figure 13 on page 22.

The jobstream in C.7, “Create VSAM Base File” on page 158 was created by the VSE II Resource Definition and File and Catalog Management dialogues to create the base file.

The jobstream in C.8, “Load VSAM Base File” on page 159 was created by the VSE II Resource Definition and File and Catalog Management dialogues to load the first record into the file. This could also be done with the sample program in C.9, “Sample VSAM Load Program” on page 159.

6.2.3 Create VSAM Alternate Index

Our application can also access the data via an alternate key. This is the SURNAMEO field. The jobstream in C.10, “Define VSAM Alternate Index” on page 160 was built by the VSE II to define the alternate index.

The jobstream in C.11, “Build VSAM Alternate Index” on page 161 was built by the VSE II to create the alternate index from the base file.

6.2.4 Sample of BMS Application Screen

Here is an example of the way one of our BMS front-end application screens appeared:

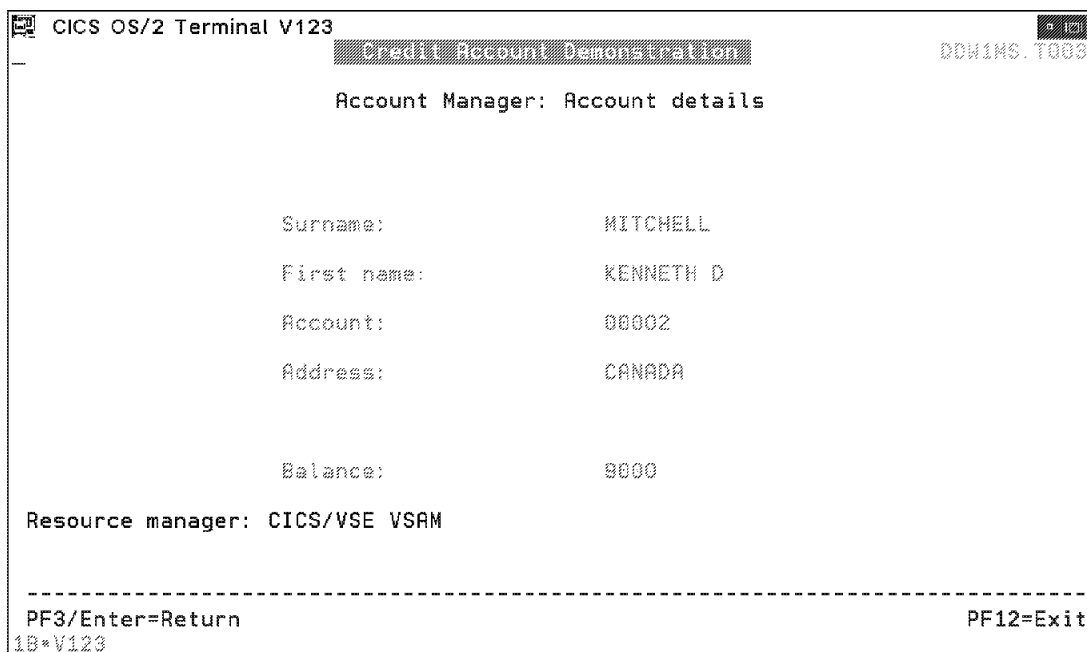


Figure 39. Application BMS Screen

6.3 Front-end Application Implementation on OS/2 (PM based)

An alternative to the BMS based front-end program was tested in our environment. We used an OS/2 program running on one of our CICS OS/2 client machines to provide a Presentation Manager front-end to the user. Here is an example of the way it appeared:

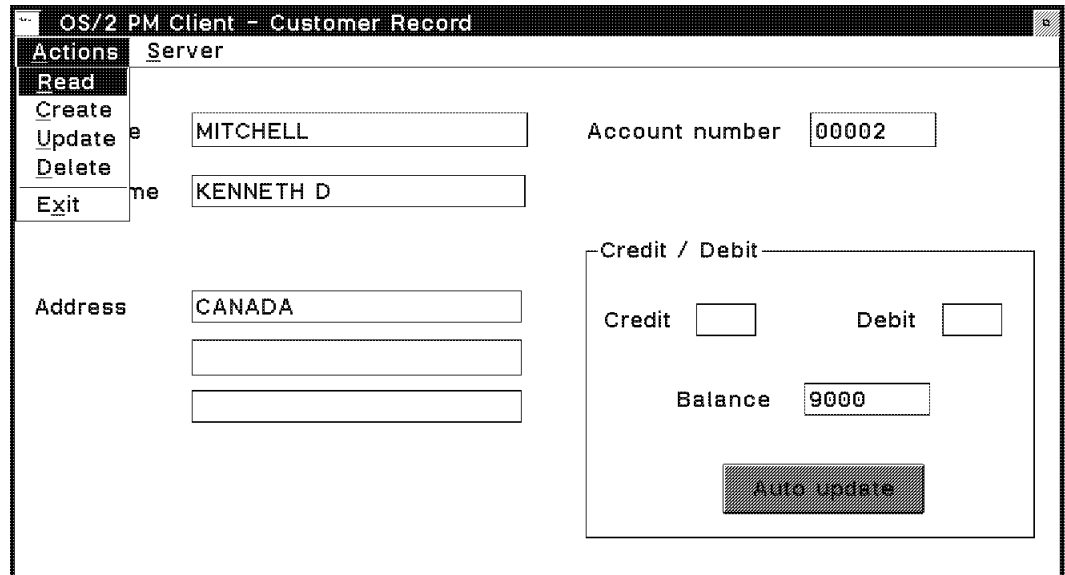


Figure 40. Application PM Window

We made **no changes** to our environment to implement the Presentation Manager OS/2 front-end.

Based on interaction with the user, the PM front-end program builds a COMMAREA then issues an **ECI** call to the CICS OS/2 client function. The CICS OS/2 client forwards the call to the CICS OS/2 server. The server identifies the program being called as remote, because of the PPT definition in Figure 38 on page 45, and calls our host backend program.

The program source for this PM based front-end is available on the IBMVSE disk which contains the User Examples Project from the IBM Boeblingen Lab. Registered users of TALKLINK (a service of IBMLink*) may have access to it.

Part 3. Implementing CICS/VSE to CICS/6000 Client/Server Solutions

Chapter 7. CICS/6000 to CICS/VSE Overview

The next five chapters document the parameters used to configure and customize the environment described in Figure 41 on page 59 to make our sample application run in a distributed fashion, that is, with our front-end program running on CICS/6000 using Distributed Program Link to a backend program on CICS/VSE.

7.1 CICS/6000

CICS/6000 is a member of the CICS family which enables CICS applications to run on an AIX platform. In addition to the standard Inter System Communication (**ISC**) facilities, CICS/6000 also provides an **External Presentation Interface (EPI)** by which non-CICS AIX applications can call CICS programs or transactions. These non-CICS programs may use a GUI for presentation services.

- By using the CICS **EPI**, the non-CICS program appears to CICS/6000 as one or more 3270 terminals. It may start CICS transactions and send and receive 3270 data streams.
- The CICS/6000 program may then invoke **ISC** facilities to communicate with CICS/VSE programs. The CICS/VSE program then returns to the CICS/6000 program, which in turn sends data to a non-CICS program where it may be presented to the user via a GUI.

7.2 The CICS/6000 Server

The server version of AIX CICS/6000 provides a CICS environment under AIX which supports CICS ISC to a CICS/VSE host and communicates to CICS/6000 client workstations via TCP/IP. In that way the CICS/6000 server acts as a gateway to CICS/VSE for CICS/6000 clients. The CICS/6000 server communicates with CICS/VSE via SNA LU6.2, using IBM Encina Peer to Peer Gateway and AIX SNA Services/6000.

AIX programs on the CICS/6000 server may use the EPI to invoke CICS programs and transactions. In addition, a 3270 BMS style interface is available to invoke CICS transactions.

CICS programs running on the server are command level programs written in Micro Focus COBOL/6000 for Unix or C using minimum BMS. With that the CICS/6000 server can be used as an application server for CICS/6000 clients in a TCP/IP network.

7.3 The CICS/6000 Client

The client component of CICS/6000 provides terminal and printer access to the CICS/6000 server using the **cicsterm** and **cicstermp** commands.

It also provides a mechanism to create native AIX applications that can interact with CICS/6000. This mechanism could be used, for example, to create Motif** or X-Windows GUI front ends for CICS/6000 applications. These graphical front ends could run on a different machine from the CICS/6000 server and communicate with the server using **EPI** interfaces. The clients communicate via

TCP/IP to the CICS/6000 server machine. The CICS/6000 server may in turn use ISC to communicate with a CICS/VSE host.

7.4 Our Distributed Application

The application used to demonstrate the implementation of CICS/6000 to CICS/VSE in this book is composed of two programs:

- A front-end CICS program which handles the interface to the user via BMS maps.
- A backend program running on CICS/VSE which handles VSAM file operations.

These two programs can reside on the same or different platforms.

The front-end program LINKs to the backend passing a COMMAREA, that is uses DPL to link to CICS/VSE. The backend program updates this COMMAREA with data from the VSAM file and RETURNS control to the front-end program.

The source code of the sample application is provided in Appendix C, "Sample Application Job Streams and Definitions" on page 133 together with the appropriate job streams for setting up the host environment. Chapter 12, "Implementing an Application in CICS/6000" on page 103 describes the implementation aspects of CICS/6000.

CICS **D**istributed **P**rogram **L**ink style of communication is used between the front-end and backend programs (refer to Chapter 1, "Client/Server Computing with CICS/VSE" on page 3).

Chapter 8. CICS/6000 to CICS/VSE Implementation Environment

This chapter describes the hardware and software used in our project.

Our environment consists of:

- a VSE/ESA host running CICS/VSE
- a CICS/6000 server workstation
- a CICS/6000 client workstation

CICS/VSE and CICS/6000 server are connected to each other via a Token-Ring LAN running SNA protocols. The CICS/6000 server and client communicate via TCP/IP on the same Token-Ring LAN. The network is illustrated in Figure 41 on page 59.

8.1 Hardware

- An IBM 9221 Model 150 as the host with:
 - 64MB main memory
 - Token-Ring connection via channel attached IBM 3172
- An IBM RISC/6000 Model 560 attached to the IBM Token-Ring LAN as CICS/6000 server
- An IBM RISC/6000 Model 320 attached to the IBM Token-Ring LAN as CICS/6000 client

8.2 Software

- VM/ESA R2.1 with GCS and ACF/VTAM V3.4.1 in the IBM 9221
- VSE/ESA Version 1.3.3 guest under VM in the IBM 9221 with
 - ACF/VTAM V3.4.0
 - CICS/VSE V2.2
- Interconnect Controller Program (ICP) Version 3.2 in the IBM 3172 Model 3
- CICS/6000 server which operates under AIX Version 3.2.5 and consists of
 - IBM AIX CICS/6000 V1.1.0.0
 - AIX DCE & Encina Base Services/6000 V1.2.0.0 including
 - AIX DCE Cell Directory Server/6000
 - AIX DCE Security Server/6000
 - IBM Encina Server for AIX/6000 V1.1.1.0 including
 - IBM Encina Structured File Service
 - IBM Encina Peer to Peer Gateway
 - AIX SNA Services/6000 V1.2
 - Micro Focus COBOL/6000 for Unix V3.1
- CICS/6000 client which operates under AIX Version 3.2.5 and consists of
 - IBM AIX Client for CICS/6000

– AIX DCE & Encina Base Services/6000 V1.2.0.0

A detailed list of the Licensed Program Products (LPPs) required for CICS/6000 (server and client) is provided in Table 1 on page 61 and Table 2 on page 61 in 9.1, “Planning a CICS/6000 Installation” on page 61.

8.3 Network Configuration

Figure 41 on page 59 illustrates the network described above including the MAC Addresses used in the appropriate definitions in VTAM, AIX SNA Services/6000 and the Interconnect Controller Program as well as the IP addresses within the TCP/IP network.

The CICS/6000 server and client are attached to two different TCP/IP networks since they are located in two separate buildings; physically they are connected via several IBM Token-Ring LANs forming one logical Token-Ring via IBM 8209 bridges.

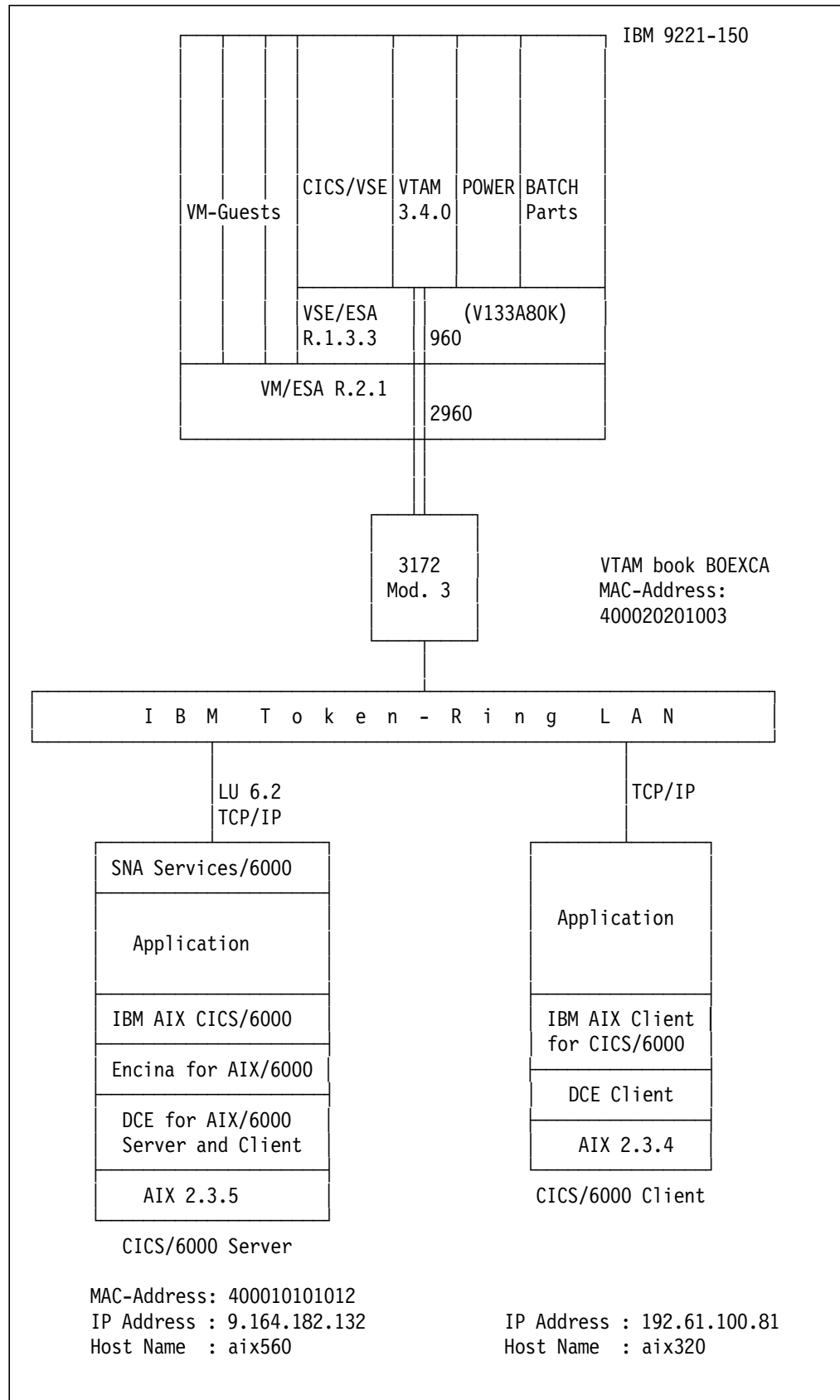


Figure 41. CICS/6000 to CICS/VSE Network Diagram

Chapter 9. Installing and Configuring the IBM AIX CICS/6000 Server

The installation and customization steps for both, IBM AIX Client for CICS/6000 and IBM AIX CICS/6000 Server are described in detail in *AIX CICS/6000 Installation and Configuration: A Guide to Implementation, GG24-4091* and *AIX CICS/6000 Planning and Installation, GC33-0816*. The emphasis of the implementation guidelines of this book is on setting up SNA LU6.2 connections between the CICS/6000 server and CICS/VSE.

Therefore, in chapters 9 and 10 the installation and customization steps for the CICS/6000 and client are only briefly reviewed. It is strongly recommended to use the two documents mentioned above if more detailed information is required regarding these tasks.

9.1 Planning a CICS/6000 Installation

Table 1 shows the Licensed Program Products (LPPs) required to install a stand-alone IBM AIX CICS/6000 server:

Product	LPP Name	Description
DCE	dcebase	Libraries for the DCE LPPs
	dcecds	Cell Directory Server
	dcesec	Security Server
Encina	encExec	Base, transactional RPCs
	encExecmEn_US	Messages
	encServ	Storage and maintenance
	encServmEn_US	Messages
	encSfs	Sturctured File Server
	encSfsmEn_US	Messages
	ppcExec	Access to CICS ISC
	ppcExecmEn_US	Messages
CICS	cics6000base	Libraries for the CICS/6000 LPPs
	cics6000	Server (run time + development)
	cics6000clt	Client (run time + development)
	cics6000mEn_US	Messages and help

Table 1. LPPs Required for an IBM AIX CICS/6000 Server

Table 2 shows the LPPs needed to enable **ISC** communication between a CICS/6000 server and other CICS platforms.

Product	LPP Name	Description
Encina	ppcGate	Gateway to CICS SNA
	ppcGatemEn_US	Messages
SNA Services/6000	sna.lu0	lu0 support
	sna.sna	lu6.2 support

Table 2. LPPs for IBM AIX CICS/6000 ISC Support

Communication Software: CICS/6000 requires Transmission Control Protocol/Internet Protocol (TCP/IP) to communicate with AIX clients and AIX SNA Services/6000 to use CICS ISC functions. TCP/IP is also required by DCE.

9.2 Preliminary AIX Configuration

Some preliminary work has to be done within AIX before the CICS/6000 server can be installed and configured:

- Check the amount of allocated page space on your machine
- Create the appropriate file systems
- Configure the TCP/IP subsystem
- Set some environment variables
- Create groups and users.

Detailed information regarding these steps is provided in *AIX CICS/6000 Installation and Configuration: A Guide to Implementation, GG24-4091*, Topic 4.2 Preliminary AIX Settings.

9.3 CICS/6000 Installation

CICS/6000 requires the Distributed Computing Environment (DCE) LPPs in order to define a working **DCE cell**. The cell must contain at least one Encina file and log server. To enable CICS/6000 for using ISC functions to communicate with CICS/VSE, the cell must also contain an **Encina Peer to Peer Gateway**.

Our installation consists of one IBM AIX CICS/6000 Server, named **aix560**, and one IBM AIX Client for CICS/6000, named **aix320**. The CICS/6000 server acts as a gateway to CICS/VSE. Table 1 on page 61 and Table 2 on page 61 show the LPPs required for the server machine. Chapter 10, "Installing and Configuring the IBM AIX CICS/6000 Client" on page 65 provides information on how to install and customize the AIX client for CICS/6000.

Before CICS/6000 can be installed, an AIX user called **cics** and AIX groups named **cics** and **cicsterm** must have been created. In addition the user **root** must be a member of group **cics**.

To install the LPPs from tape through SMIT use the following procedure:

1. Log in as 'root'.
2. Insert the tape into the tape drive.
3. To display the appropriate SMIT panel, enter '**smitty install_latest**'.
4. Enter the name of the tape drive.
5. Move the cursor to the '**SOFTWARE to install**' field in the SMIT panel and press F4 to display a list of the LPPs you may want to install.
6. Select the appropriate LPPs.
7. Start the installation.

Once all of the products listed in Table 1 on page 61 are successfully installed you are ready to configure DCE and Encina.

For detailed information refer to *AIX CICS/6000 Planning and Installation*, GC33-0816.

9.4 CICS/6000 Server Configuration

9.4.1 Configuring DCE for CICS/6000

CICS/6000 requires a working DCE cell which must contain at least one security server and one cell directory server.

All necessary components within the cell have to be created and the DCE groups and directories that CICS/6000 needs have to be added. This is achieved by the following four steps (the first two are part of the DCE setup, the last two are for CICS/6000):

1. Configure DCE's Security Server
2. Configure DCE's Cell Directory Server (CDS)
3. Authenticate yourself as the DCE administrator
4. Configure the DCE cell for use by CICS/6000

A detailed description of DCE cell configuration is provided in *AIX CICS/6000 Installation and Configuration: A Guide to Implementation*, GG24-4091, Topic 4.4 DCE Cell Configuration.

9.4.2 Configuring Encina for CICS/6000

The Encina Structured File Server (SFS) manages data for CICS/6000 and provides a variety of file services, including a VSAM-like interface.

The Encina log server provides logging functions for both CICS/6000 and SFS and is the base for recovery functions if transactions fail.

CICS/6000 permits a high degree of flexibility in the distribution of machines and CICS/6000 components. To set up one particular environment perform the following steps:

1. Create AIX Logical Volumes for Encina
2. Create AIX Users for Encina Servers
3. Configure the Default Encina Log and File Servers which in turn requires the following:
 - Create default Encina servers
 - Cold start the log server
 - Configure a log server for an SFS
 - Cold start the SFS

A detailed description of how to configure Encina for CICS/6000, can be found in *AIX CICS/6000 Installation and Configuration: A Guide to Implementation*, GG24-4091, Topic 4.5 Encina Configuration.

9.4.3 Creating and Starting a CICS/6000 Region

When the DCE cell and Encina are successfully configured, you can create a CICS/6000 region and link the Encina server(s) to that region.

To create and start a CICS/6000 region perform the following steps:

1. Create the region
2. Configure the Encina servers for the region
3. Create a CICS user
4. Start the CICS/6000 region

Details on how to create and start a CICS/6000 region are provided in *AIX CICS/6000 Installation and Configuration: A Guide to Implementation, GG24-4091*, Topic 4.6 Creating a CICS/6000 Region, details on how to create CICS/6000 users are presented in Chapter 5 Creating Users for CICS/6000 of the same manual.

If one or more CICS/6000 regions are running you can access one using the command:

```
# cicsterm
```

The resulting panel shows available CICS Regions and you are able to connect to the one you wish, *CICS6K* in our example.

9.5 CICS/6000 Server Installation Summary

Table 3 shows the installation variables we used for our installation.

Description	Value
AIX Hostname	aix560
Cell Name	cell01
CICS/6000 Region name	CICS6K
SFS Logical Volume	sfs_SFS_SERV
SFS/log Logical Volume	log_SFS_SERV
LOG Logical Volume	log_LOG_SERV
CICS/log Logical Volume	log_CICS6K
DCE Principal for SFS	cics/sfs/aix560
DCE Principal for LOG	cics/log/aix560
DCE Principal for CICS region	cics/CICS6K
Initial DCE password	-dce-
DCE cell_admin password	-dce-
CICS/6000 Users name	WSR1,WSR2,WSR3
CICS/6000 Users password	WSR1,WSR2,WSR3

Table 3. Summary of CICS/6000 Installation Variables

Chapter 10. Installing and Configuring the IBM AIX CICS/6000 Client

This chapter provides a brief description of how to install and configure the DCE, Encina and CICS/6000 components required to run IBM AIX Client for CICS/6000 on another RISC/6000 system in the same DCE cell as the CICS/6000 server.

Our DCE cell consist of two machines:

- the DCE server machine, referred to as *server* (aix560), and
- the DCE client machine, referred to as *client* (aix320).

The DCE server contains a "complete" DCE configuration, that is it contains all the servers shown in Table 1 on page 61.

In our implementation all services for the client (aix320) reside on the server (aix560), that is:

- SFS services
- Cell directory services
- Security services
- Logging services

The client machine will use the services of the Security Server and the Cell Directory Server, residing on the DCE server machine.

10.1 Planning an AIX CICS/6000 Client Installation

Table 4 shows the LPPs needed to install an IBM AIX Client for CICS/6000:

Product	LPP Name	Description
DCE	dcebase	Libraries for the DCE LPPs
Encina	encExec	Base, transactional RPCs
	encExecmEn_US	Messages
CICS	cics6000base	Libraries for the CICS/6000 LPPs
	cics6000clt	Client (run time + development)
	cics6000mEn_US	Messages and help

Table 4. LPPs Required for IBM AIX Client for CICS/6000

10.2 Preliminary AIX Configuration

Before installing and customizing a CICS/6000 client the following preliminary AIX configuration steps are required:

- Configure TCP/IP subsystem
- Set some environment variables
- Create groups and users

A detailed description of these steps can be found in *AIX CICS/6000 Installation and Configuration: A Guide to Implementation, GG24-4091*, Topic 8.1 Preliminary AIX Settings.

10.3 Configuring DCE for CICS/6000 Client Support

After successful installation of the LPPs shown in Table 4 on page 65, you have to configure DCE client support for the CICS/6000 clients.

First, the client machine's clock needs to be synchronized with the clock in the server (aix560). This is done by entering:

```
# setclock aix560
```

To get a client up and running perform the following steps:

1. Make the DCE client a part of the existing cell by entering:
smitty mkdceclient.
2. Move the cursor to the '**CLIENTS to configure**' field, in the resulting panel, press '**F4**' to obtain a list of the possible DCE clients. Select the following entries:
 - RPC EnpPoint Mapper (rpc)
 - Security Client (sec_cl)
 - CDS Client (cds_cl)
3. In the '**CELL name**' field enter cell name, '*cell01*' in our example.
4. In the fields '**SECURITY server**' and '**CDS server**' enter the server's name, '*aix560*' in our example.
5. Press Enter to continue.
6. Enter the password for the DCE principal '**cell_admin**' when you are prompted.
7. Configure DCE for the CICS/6000 client by entering '**cicssetupclients**' at an AIX prompt. This command creates the client's DCE principal and account and places the password in a keytab file.

More details are provided in *AIX CICS/6000 Installation and Configuration: A Guide to Implementation, GG24-4091*, Topic 8.3 Configuring DCE for CICS/6000 Client Support.

10.4 Accessing the CICS/6000 Region

Now that the DCE clients have been installed and configured, you can log in as a DCE user and access your CICS/6000 region using the commands:

```
# dce_login
```

```
# cicsterm
```

From the list of available CICS regions choose the one to which you want to connect, '*CICS6K*' in our example.

Chapter 11. Connecting the CICS/6000 Server to the Host

This chapter describes how to connect the CICS/6000 server to VSE/ESA via ACF/VTAM and to CICS/VSE. From all tasks we performed this turned out to be the most complex one due to:

- the number of components involved
- the lack of experience and documentation for this type of communication.

The implementation and customization of the connection consists of three parts which in turn involve a number of steps:

1. The definitions required for the host which consist of customizing:

- ACF/VTAM
- CICS/VSE

These definitions are provided in 11.3, "CICS/6000 Connection Customization" on page 73.

2. The definitions required for CICS/6000 which consist of customizing:

- the AIX environment
- the Token-Ring adapter of the RISC/6000
- DCE
- AIX SNA Services/6000
- CICS/6000 for ISC

These definitions are provided in 11.2, "VSE/ESA Host Customization" on page 69.

3. The operating instructions for starting the connection including hints and tips on how to monitor CICS/6000 to CICS/VSE communications.

These instructions are provided in 11.4, "Interconnection Operations" on page 95, 11.5, "Monitoring the Encina PPC Gateway" on page 97 and 11.6, "Operational Hints" on page 97.

A summary of the customization process is given in 11.7, "CICS/6000 to CICS/VSE Customization Summary" on page 100.

Before we turn to the customization tasks we briefly review the ISC aspects of CICS/6000 to CICS/VSE communication.

11.1 Introduction to CICS/6000 Intercommunication

The CICS/6000 support provided for the CICS Inter System Communication (ISC) functions is designed to simplify distributed system operation. In general, this support extends the standard CICS functions (such as reading and writing to files and queues) in such a way that the application or user is unaware that the resource they are using may be situated on a remote system. CICS/6000 supports the following ISC functions:

- **Distributed Program Link (DPL):** extends the use of the EXEC CICS LINK command to allow a CICS application program to link to a program that resides on a different CICS system.

- **Function shipping:** allows an application program to access files, transient data queues, and temporary storage queues belonging to another CICS system.
- **Asynchronous processing:** extends the EXEC CICS START command to allow an application to initiate a transaction to run at a certain time on another CICS system. As with standard EXEC CICS START calls, the transaction requested in the START command runs independently from the application issuing the START command.
- **Distributed Transaction Processing(DTP):** provides additional EXEC CICS command interface calls to allow two applications running on different systems to pass information between each other. These EXEC CICS commands map to the LU6.2 mapped conversation verbs defined in the SNA architecture. This is the only intercommunication function that is available both between CICS systems and CICS and a non-CICS system.

11.1.1 Network Protocol Considerations

When two systems communicate, they need to agree on a set of rules they will use to interpret the data sent between them. These rules are known as *network protocols* and they are defined in what is called a network architecture.

CICS/6000 can communicate with remote systems either via Transmission Control Protocol/Internet Protocol (TCP/IP) connections or across a System Network Architecture (SNA) network. Which of the two methods is used depends almost entirely on the type of the remote system. Non-UNIX CICS platforms, such as CICS/MVS, CICS/ESA, CICS/VSE, CICS/400, or CICS OS/2, use SNA. UNIX-based CICS platforms, such as CICS/6000 will most likely choose TCP/IP (although CICS/6000 can also communicate via SNA using AIX SNA Services/6000).

11.1.2 Using SNA Communication

A CICS/6000 region requires the following components (LPPs) for communication to SNA-connected CICS platforms:

- Encina Peer-to-Peer Communication (PPC) Executive for AIX/6000
- Encina PPC Gateway for AIX/6000
- AIX SNA Services/6000

When CICS/6000 sends an inter-system request to an SNA-connected system, the PPC Executive contacts the PPC Gateway across TCP/IP. The PPC Gateway then routes the inter-system request to the SNA-connected system.

In general the PPC Gateway and AIX SNA Services/6000 are installed independently from CICS/6000, that is on another machine. In this case you need to make sure that these products are available, that is that they can be accessed from your CICS region.

In our sample configuration all three components are installed on the same machine, the CICS/6000 server (aix560).

11.1.3 Using TCP/IP Communication

CICS/6000 regions, whether they are UNIX-based or not, can communicate with each other via TCP/IP since the underlying Encina PPC Executive is able to emulate SNA LU6.2 on TCP/IP connections if necessary.

11.2 VSE/ESA Host Customization

VSE/ESA customization involves three steps:

1. **Network attachment** to connect (in our configuration) the IBM 9221 to the IBM Token-Ring LAN (see Figure 41 on page 59). These definitions are already described in Chapter 4, "Define and Customize the VSE/ESA Host" on page 11.
2. **ACF/VTAM definitions** to provide for SNA LU6.2 communications.
3. **CICS/VSE definitions** to enable communication to and from CICS/6000.

11.2.1 ACF/VTAM Customization

The RISC/6000 is defined to ACF/VTAM via:

- a VTAM switched major node and
- an independent LU6.2 logical unit.

11.2.1.1 VSE/VTAM SWNET Major Node

A SWNET major node is created to define the physical and logical units to which VTAM may communicate through the 3172 XCA major node. Figure 42 on page 70 shows the definitions we used.

Note the following parameters:

- **IDBLK** is 071. These are the first three hexadecimal digits of AIX SNA Services **Control Point** profile's attribute 'xid_node_id'.
- **IDNUM** is 00000. These are the last five hexadecimal digits of AIX SNA Services **Control Point** profile's attribute 'xid_node_id'.
- **PATH** The last six bytes of the DIALNO operand contain RISC/6000 server's ALTERNATE TOKEN RING address.
- **LOCADDR=0** implies Independent Logical Unit definition.
- **DLOGMODE=LU62PS** is the name of the log mode entry which describes the session parameters for APPC conversations between CICS/6000 and CICS/VSE.

The log mode name **LU62PS** must match the mode name of an AIX SNA Services **LU6.2 Mode** profile, which must be referenced in the AIX SNA Services Connection Profile mode list. We are using the same log mode entry as for the CICS OS/2 to CICS/VSE connection (see Figure 9 on page 18).

In addition, the LU name **must** be the same as the CICS/6000 region name, *CICS6K* in our case.

```

*
* 3172 RELATED SWITCHED MAJOR NODE FOR RS/6000
*
BOEXCASW      VBUILD TYPE=SWNET,MAXGRP=20,MAXNO=20
*
* PU AND LU DEFINITION FOR RS/6000
*
IPFCPX12 PU   ADDR=04,          STATION ADDRESS FOR RS/6000      X
              LANSW=YES,       LAN capable                          X
              IDBLK=071,       IDENTIFICATION BLOCK                X
              IDNUM=00000,     IDNUM SET BY ITSC CONVENTION        X
              DISCNT=NO,       VTAM does not hang up                    X
              ISTATUS=ACTIVE,PACING=0,VPACING=0, NO PACING          X
              PUTYPE=2,        SNA cluster controller                X
              MAXDATA=521,     FROM 265 TO 2057                          X
              MAXOUT=7,        RECOMMENDED BY DOC                          X
              MAXPATH=1,       ADDED FROM LAB DEFN                          X
              LANACK=(01.0,1), LAN ACKNOWLEDGEMENT VALUES          X
              LANINACT=02.0,   timer for inactive link station        X
              LANCON=(05.0,1), LAN TIMER AND RETRY COUNT            X
              LANSDDW=(2,1),   SEND WINDOW & WINDOW STEP          X
              LANRESP=(02.0,2), TIMER FOR CONNECTED STATE          X
              SAPADDR=4        service access point address
*
* PATH STATEMENT FOR DIAL OUT
*
      PATH DIALNO=0204400010101012
*
* LU DEFINITION FOR RS/6000
*
CICS6K LU     LOCADDR=0,DLOGMOD=LU62PS,          (APPC LU)      X
              ISTATUS=ACTIVE,MODETAB=CICSIPMT

```

Figure 42. CICS/6000 VSE/VTAM SWNET Major Node for 3172

11.2.2 CICS/VSE Customization

This section covers the definitions required in CICS/VSE to enable CICS/6000 to communicate with CICS/VSE. The following definitions are common to both, CICS OS/2 and CICS/6000 and are described in 4.2, "CICS/VSE Customization" on page 19:

- System Initialization Table (DFHSIT)
- Terminal Control Program (DFHTCP)
- Data Conversion Program (DFHCNV)
- File Control Table (DFHCNV)

We used the 'CEDA DEF' transaction to define following resources to CICS:

- Connections
- Sessions

Regarding the application, the backend program is the same as in the CICS OS/2 to CICS/VSE case (see 6.2, "Application Implementation on CICS/VSE" on page 50). The appropriate entries for the front-end program are described in 12.3, "CICS Table Entries" on page 105.

All CICS/VSE resources are defined in group *CICS6000*. After definition we used the CICS '**CEDA**' transaction to activate them and to make them permanent in the CICS startup list:

- **CEDA INSTALL GROUP(CICS6000)**
- **CEDA ADD GROUP(CICS6000) LIST(VSELIST)**

11.2.2.1 Connection Definition

Autoconnect=No allows for manual control when the connection between CICS/VSE and the RISC/6000 system is activated by entering a '**CEMT SET CONN(sysid) ACQ**' (Sysid is *ISCO* in our case) transaction. '**Singlesess**' must be set to '**No**' to provide parallel session (independent LU) support. Figure 43 shows our connection definition.

```

OBJECT CHARACTERISTICS
CEDA VIEw
  Connection      : ISCO
  Group          : CICS6000
CONNECTION IDENTIFIERS
  Netname        : CICS6K
  INdsys         :
REMOTE ATTRIBUTES
  REMOTESystem   :
  REMOTENAME     :
CONNECTION PROPERTIES
  AAccessmethod  : Vtam          Vtam | IRc | INdirect
  Protocol       : Appc          Appc | Lu61
  Singlesess    : No           No | Yes
  Datastream     : User         User | 3270 | SCs | STRfield | Lms
  RECORDformat   : U            U | Vb
OPERATIONAL PROPERTIES
  Autoconnect  : No           No | Yes | All
  INService      : Yes          Yes | No
SECURITY
  SEcurityname   :
  ATTachsec      : Local        Local | Identify | Verify
  Bindpassword   :              PASSWORD NOT SPECIFIED

PF 1 HELP      3 END          6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Figure 43. Connection for CICS/6000

The key parameters are:

- **CONNECTION** specifies the name of the connection.
- **NETNAME** specifies the name of the independent LU. It must match the:
 - Local LU NAME in the AIX SNA Services/6000 Local LU Profile definition as shown in Figure 52 on page 81
 - Name of the independent LU defined to VTAM as shown in Figure 42 on page 70.
- **SINGLESESS** specifies 'NO' in order to use APPC parallel sessions.

11.2.2.2 Session Definition

A session definition is used to define the logical links and session characteristics between CICS/VSE and the independent LU (CICS6K). '**AUTOconnect=Yes**' should be specified. This will bind only the contention winners when the connection is started (AUTOconnect=All would cause the CICS/VSE to bind all sessions). The **MAXimum** parameter is specified as 'XXX,YYY'. 'XXX' is the maximum number of sessions, and 'YYY' the minimum number of contention winners. YYY should be greater than 0 and less than XXX. This will reserve some sessions for contention winners on the AIX side. The values will be negotiated when the connection is started, but they should be consistent with what is specified in the corresponding AIX SNA Services LU6.2 Mode profile (see Figure 53 on page 82).

Figure 44 shows our session definition.

```

OBJECT CHARACTERISTICS
CEDA View
Sessions      : ISCO
Group          : CICS6000
SESSION IDENTIFIERS
Connection   : ISCO
SESSName      :
NETnameq      :
MOdename     : LU62PS
SESSION PROPERTIES
Protocol    : Appc          Appc | Lu61
MAximum     : 00008 , 00004  0-32767
RECEIVEPfx    :
RECEIVECount  : No          No | 1-999
SENDPfx       :
SENDCount     : No          No | 1-999
SENDSIZE      : 00256       1-30720
RECEIVESIZE   : 00256       1-30720
OPERATOR DEFAULTS
OPERId        :
OPERPriority   : 000         0-255
OPERRr1       : 0           0-24,...
OPERSecurity   : 1         1-64,...
USERId        :
SESSION USAGES
Transaction    :
SESSPriority   : 100        0-255
OPERATIONAL PROPERTIES
Autoconnect  : Yes        No | Yes | All
INservice     :            No | Yes
Buildchain    : Yes        Yes | No
USERArealen   : 000        0-255
IOarealen     : 00000 , 00000  0-32767
RELreq        : No         No | Yes
Discreq       : Yes        No | Yes
NEPclass      : 000        0-255
RECOVERY
RECOvoption   : Sysdefault  Sysdefault | None

PF 1 HELP      3 END          6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL

```

Figure 44. CICS/6000 Session Definition

The key parameters are:

- **SESSION.** Specifies the name of the session.
- **CONNECTION.** Defines the name of the connection associated with this session. It must match the CONNECTION name in CICS/VSE's CONNECTION definition as in Figure 43 on page 71.
- **MODENAME.** The logmode entry name for the independent LU named **CICS6K.** It must match the MODEENT name in the VTAM MODETAB (see Figure 9 on page 18).
- **PROTOCOL.** APPC is required for CICS ISC.
- **AUTOCONNECT.** 'YES' allows CICS to automatically establish a session with the session partner *CICS6K.*

In the CICS/VSE region, the terminal you are using for any CICS/6000 routed transaction should be specified with UCTRAN off, since CICS/6000 transactions are largely case sensitive.

11.3 CICS/6000 Connection Customization

Before starting this step verify that all LPPs shown in Table 2 on page 61 are installed in your cell. Otherwise they have to be installed to enable CICS ISC functions. Note that AIX SNA Services/6000 must be installed **before** Encina PPC Gateway Services/6000.

CICS/6000 server customization involves the following five tasks:

1. Customize the AIX environment
2. Customize the Token-Ring adapter of the RISC/6000
3. Customize DCE
4. Customize AIX SNA Services/6000
5. Customize the CICS/6000 server for ISC

11.3.1 AIX Environment Customization

CICS/6000 needs to be operational and AIX SNA Services/6000 and the Encina PPC Gateway must be installed at this point.

Update your */etc/environment* file to include the statements shown in Figure 45. This will set some variables required later by the Encina PPC Gateway.

```
ENCINA_CDS_ROOT=./cics/
ENCINA_GWY_SERVER=./cics/ppc/gateway/GWY1
ENCINA_PRINCIPAL=../../ce1101/MyGtw
ENCINA_AUTHN=1
ENCINA_MIN_AUTHN=1
```

Figure 45. AIX Environment Customization

GWY1 is the name of our gateway

ce1101 is the name of our DCE cell

MyGtw is the name of the DCE principal under which the gateway will run

ENCINA_AUTHN=1 instructs ENCINA to send unauthenticated RPCs to the CICS/6000 region

ENCINA_MIN_AUTHN=1 instructs ENCINA to accept unauthenticated RPCs from the CICS/6000 region

Reboot AIX to activate these environment variables.

11.3.2 Token-Ring Adapter Customization

For the IBM Token-Ring LAN connection to VSE/ESA, AIX must define the hardware address of its Token-Ring adapter. We use SMIT to change the address to an ALTERNATE TOKEN RING address, that is a locally administered address:

1. Enter '**smitty devices**'.
2. Move the cursor to the '**Communication**' field and press 'Enter'.
3. Select the '**Token Ring Adapter**' field.
4. In the Token Ring Adapter panel select '**Adapter**'.
5. Select field '**Change / Show Characteristics of a Token Ring Adapter**' and move the cursor to the appropriate item and press 'Enter' to obtain the following panel:

```
Change / Show Characteristics of a Token Ring Adapter

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

Token Ring Adapter          'Entry Fields'
Description                 tok0
Status                     Token-Ring High-Perfor>
Location                   Available
Receive data transfer OFFSET 00-04
TRANSMIT queue size        '92'          + #
RECEIVE queue size         '30'          + #
STATUS BLOCK queue size    '30'          + #
RING speed                 '10'          + #
Receive ATTENTION MAC frame 16                +
Receive BEACON MAC frame   no                 +
Enable ALTERNATE TOKEN RING address yes          +
ALTERNATE TOKEN RING address '0x400010101012'
Apply change to DATABASE only no                 +

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

Figure 46. Token-Ring Adapter Customization

6. Enable ALTERNATE TOKEN RING address if it is not enabled.
7. Move the cursor to field '**ALTERNATE TOKEN RING address**' and enter an address (400010101012 in our example).
8. Press 'Enter' to make the changes effective.

If you intend to have a host connection via NCP and your NCP Release is lower than 6, the Token-Ring hardware address must contain only numeric characters (0-9).

11.3.3 DCE Customization

In order to connect to CICS/VSE (or other CICS platforms) via SNA, DCE requires the Encina PPC Gateway. This gateway runs within the DCE cell we created (refer to 9.4.1, “Configuring DCE for CICS/6000” on page 63) using its own principal (principals are members of a DCE cell and represent client and user processes, subsystems and so on).

While performing its function, that is connecting to SNA hosts, the Encina PPC Gateway dynamically updates the password of its principal. Consequently, **the name ‘cell-admin’ or some other common name must not be used for Encina PPC Gateway’s principal.**

Figure 47 shows how we defined the DCE principal under which the Encina PPC Gateway will run. We chose ‘MyGtw’ as the principal name for our gateway using the AIX *rgy_edit* command to create the appropriate DCE keytab file (for details on DCE keytab file refer to chapter 5: Creating a DCE User, in *AIX CICS/6000 Installation and Configuration: A Guide to Implementation, GG24-4091*).

```
# rgy_edit
rgy_edit> domain principal
rgy_edit> add MyGtw
rgy_edit> domain account
rgy_edit> add MyGtw -g cics_admin -o none -pw MyGtw -mp -dce-
rgy_edit> ktadd -p MyGtw -pw MyGtw
rgy_edit> exit
#
```

Figure 47. DCE Customization for Encina PPC Gateway

11.3.4 AIX SNA Services/6000 Customization

Before we customized AIX SNA Services/6000 on our system we applied the following PTFs on top of our version 1.2 of the LPP:

1. U426472
2. U426473
3. U428048

The customization of AIX SNA Services/6000 can be divided into three parts:

1. The customization of profiles describing its **physical** characteristics, that is:
 - a. the SNA Token-Ring Attachment Profile
 - b. the SNA Token-Ring Logical Link Profile
 - c. the SNA Token-Ring Physical Link Profile
2. The customization of profiles describing its **logical** characteristics, that is:

- a. the SNA LU6.2 Logical Connection Profile
 - b. the SNA LU6.2 Local LU Profile
 - c. the SNA LU6.2 Mode List Profile
 - d. the SNA LU6.2 Mode Profile
 - e. the SNA LU6.2 TPN List Profile
 - f. the SNA LU6.2 TPN Profile
 - g. the SNA LU6.2 RTPN List Profile
 - h. the SNA LU6.2 RTPN Profile
3. The customization of:
- a. the SNA Node Profile
 - b. the SNA Control Point Profile

We used SMIT to create or change the appropriate SNA profiles in the following way:

1. Enter '**smitty sna**'.
2. Move the cursor to the '**Configure SNA Profiles**' field and press 'Enter'.
3. Select the '**Advanced SNA Configuration**' field.
4. Select the appropriate components or items in the following panels to create or modify their profiles.

11.3.4.1 Change Profiles for SNA Physical Units

1. Token-Ring Attachment Profile

This profile defines the physical characteristics of the communications environment. It includes data link control, transmission media and adapter assignment. Once started, an attachment provides connectivity and SNA data transport across the physical link.

The profile in Figure 48 on page 77 shows the profile we used to connect the RISC/6000 running the CICS/6000 server to CICS/VSE via the Token-Ring adapter.

```

Change / Show SNA Token Ring Attachment Profile

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

' TOP'                                     ' Entry Fields'
CURRENT profile name                       VSE62
NEW PROFILE name                           ''
CONTROL POINT profile name                 ' VSE62'
LOGICAL LINK profile name                  ' VSE62'
PHYSICAL LINK profile name                 ' VSE62'
STOP ATTACHMENT on inactivity?             no +
  If yes, inactivity TIMEOUT (0-10 minutes) '5' #
RESTART on deactivation?                    yes +
LU address REGISTRATION?                   no +
  If yes, LU address REGISTRATION PROFILE name ' LDEFAULT' +
CALL type                                   call
  If listen,
    AUTO-LISTEN?                            yes +
    MINIMUM SAP address (hex 04-ec)          '04'
    MAXIMUM SAP address (hex 04-ec)          'EC'
  If call, ACCESS ROUTING                   link_address
    If link_name, REMOTE LINK name           ''
    If link_address,
      Remote LINK address                     ' 400020201003'
      Remote SAP address (hex 04-ec)         '04'
' BOTTOM'

F1=Help      F2=Refresh      F3=Cancel      F4=List
F5=Reset     F6=Command     F7=Edit       F8=Image

```

Figure 48. VSE62 Attachment Profile

We configured the attachment profile with **CALL type=call** and physical **Remote LINK address**, which is the Token-Ring MAC Address of the IBM 3172 (refer to the network diagram in Figure 41 on page 59).

This configuration doesn't allow for connections to multiple systems. In **call** mode, only one single Token-Ring address may be used to communicate with the Token-Ring adapter on the RISC/6000 system, since AIX/SNA Services only contacts that particular address after having started the attachment profile. Other devices with different Token-Ring addresses cannot establish communication with the RISC/6000 system using this attachment.

If **listen** mode is specified, AIX SNA Services can accept Token-Ring connections from any number of other devices at the same time. The listen attachment requires that a non-RISC/6000 communication partner initiates connection to the RISC/6000; this allows multiple CICS systems on different platforms, to connect to CICS/6000 at the same time.

The key parameters are:

- **PROFILE name.** The system uses this name to refer to the set of characteristics described in this profile.
- **CONTROL POINT profile name.** This field provides the name of the Control Point Profile that defines the node ID of the physical unit associated with this attachment (refer to 11.3.4.3, "Change Profiles for SNA Nodes" on page 87 for the SNA Control Point Profile).

- **LOGICAL LINK profile name.** This field provides the name of the Logical Link Profile which defines the characteristics of the data link protocol used. (refer to Figure 49 on page 78 for the SNA TR Logical Link Profile).
- **PHYSICAL LINK profile name.** This field provides the name of the Physical Link Profile which defines the characteristics of the physical port for the network. (refer to Figure 50 on page 79 for the SNA TR Physical Link Profile).
- **Remote LINK address.** When *CALL type* is *call* and *ACCESS ROUTING* is *link_address*, this field specifies the network address of the remote station (the 3172 MAC address in our example).

2. Token-Ring Data Link Control Logical Link

This profile is required by the Attachment Profile and contains information regarding the data link level. For a valid configuration, at least one Logical Link must exist.

Figure 49 shows the Logical Link Profile we used:

```

Change / Show SNA Token Ring Logical Link Profile

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

' TOP'                                     ' Entry Fields'
CURRENT profile name                       VSE62
NEW PROFILE name                           ''
TRANSMIT window count (1-127)             '10' #
DYNAMIC window increment (1-127)          '1' #
RETRANSMIT count (1-30)                   '8' #
RECEIVE window count (1-127)              '127' #
RING ACCESS priority                        0 +
RETRY limit                                '20' #
DROP LINK on inactivity?                   yes
INACTIVITY timeout (1-120 seconds)         '48' #
RESPONSE timeout (1-40, 500 msec intervals) '2' #
ACKNOWLEDGE timeout (1-40, 500 msec intervals) '1' #
FORCE DISCONNECT timeout (1-600 seconds)   '120' #
DEFINITION of maximum I-FIELD size        system_defined +
  If user_defined, max. I-FIELD SIZE (265-30729) '30729' #
TRACE link?                                no +
  If yes, TRACE SIZE                        short +
' BOTTOM'

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

```

Figure 49. VSE62 Token-Ring Logical DLC Profile

The key parameter is:

- **PROFILE name.** The system uses this name to refer to the set of characteristics described in this profile. The profile name appears in the Attachment profile (see Figure 48 on page 77) for the attachment that uses this profile.

3. Token-Ring Data Link Control Physical Link

This profile is required by the Attachment Profile and contains information regarding the physical level. For a valid configuration, at least one Physical Link must exist. Figure 50 on page 79 shows the Physical Link Profile we used:

```

Change / Show SNA Token Ring Physical Link Profile

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

                                'Entry Fields'
CURRENT profile name              VSE62
NEW PROFILE name                  ''
DATALINK device name             'tok0'
LOCAL LINK name                   ''
Maximum number of LOGICAL LINKS (1-255) '32' #
Local SAP address (hex 04-ec)     '04' X

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

```

Figure 50. VSE62 Token-Ring Physical DLC Profile

The key parameters are:

- **PROFILE name.** The system uses this name to refer to the set of characteristics described in this profile. The profile name also appears in the Attachment Profile (see Figure 48 on page 77) for the attachment that uses this profile.
- **DATALINK device name.** This field contains the name that the local system (AIX) uses for the Token-Ring attachment.

11.3.4.2 Change Profiles for SNA LU6.2 Units

1. SNA LU6.2 Logical Connection Profile

To use a connection, you must describe its characteristics to AIX SNA Services/6000 Connection Profile.

This is the 'key' profile for CICS/6000 to AIX SNA Services/6000 connection. It defines a **LOCAL LU** and **REMOTE LU** pair, an **ATTACHMENT**, and a set of session characteristics.

This profile (named 'LU62VSE' in our example) will be referenced in the CICS/6000 **Connection Definition (CD)** for connections defined as **RemoteSysType=SNA**, in Figure 61 on page 91.

Figure 51 on page 80 shows the Connection Profile we used:

```

Change / Show SNA LU6.2 Logical Connection Profile

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

' TOP'                                     ' Entry Fields'
CURRENT profile name                       LU62VSE
NEW PROFILE name                           ''
ATTACHMENT profile name                    ' VSE62'
LOCAL LU profile name                      ' CICS6K'
NETWORK name                               ' DEIBMIPF'
STOP CONNECTION on inactivity?             no +
  If yes, TIMEOUT (0-10 minutes)          ' 0' #
REMOTE LU name                             ' CICSSA22'
SECURITY Accepted                           none +
  If conversation or already_verified,
  CONVERSATION SECURITY ACCESS LIST profile ' CONVDEFAULT'
  (If no name entered, /etc/passwd used)
REMOTE TPN LIST profile name               ' RDEFAULT'
MODE LIST profile name                     ' LU62VSE'
INTERFACE type                             extended
  If extended, SESSION CONCURRENCY        parallel
Node VERIFICATION?                         no +
' BOTTOM'

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

```

Figure 51. LU62VSE Connection Profile

The key parameters are:

- **PROFILE name.** The system uses this name to refer to the remote LU associated with the profile and to refer to the set of characteristics described.
- **ATTACHMENT profile name.** This field provides the name of the Attachment Profile that describes the characteristics of the attachment to the remote LU (see Figure 48 on page 77).
- **LOCAL LU profile name.** This field provides the name of the Local LU Profile that defines the characteristics of the local LU (see Figure 52 on page 81).
- **NETWORK name.** This field provides the network name associated with the remote LU for this connection. This name, combined with the remote LU name forms a fully qualified remote LU name ('DEIBMIPF.CICSSA22' in our case).
- **REMOTE LU name.** This field provides the name of the remote LU representing the other half of this connection. The name must match the VTAM APPLID name of CICS/VSE.
- **MODE LIST profile name.** This field specifies the name of the Mode List Profile that contains the Mode Profile applying to this connection (see Figure 53 on page 82).
- **SESSION CONCURRENCY.** Must be 'parallel' to enable CICS ISC.

At least one LU6.2 Connection Profile must exist if CICS/6000 wants to use a remote connection via SNA protocols.

Note: The terminology used in AIX SNA Services/6000 differs from the one used on other platforms, ACF/VTAM and CM/2, for example.

2. SNA LU6.2 Local LU Profile

This profile is required by the SNA LU6.2 Logical Connection Profile. It defines the characteristics of the local LU, that is CICS/6000. At least one Local LU Profile is required for each CICS/6000 region that uses an SNA connection.

Figure 52 shows the Local LU Profile we used:

```

Change / Show SNA LU6.2 Local LU Profile

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

                                'Entry Fields'
CURRENT profile name             CICS6K
NEW PROFILE name                 ''
TPN LIST profile name           'LU62VSEt'
NETWORK name                     'DEIBMIPF'
Local LU NAME                    'CICS6K'
INDEPENDENT LU?                  yes          +
  If no,
    Local LU ADDRESS (1-255)     '1'          #
    SSCP ID                       ''

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

```

Figure 52. CICS6K Local Logical Unit

The key parameters are:

- **PROFILE name.** The system uses this name to refer to the local LU associated with the profile and to refer to the set of characteristics that you described.
- **TPN LIST profile name.** This field specifies the name of the TPN List Profile that lists the TPN Profiles that can use this LU (see Figure 55 on page 84). This applies only to LUs of type 6.2.
- **NETWORK name.** This field provides the network name associated with the local LU. This name, combined with the local LU name forms a fully qualified local LU name.
If you are attaching to an existing network, use the name of that network.
- **Local LU NAME.** This field provides the name of the local LU. The name must match the VTAM LU definition (see Figure 42 on page 70) and CICS/VSE connection definition (see *Netname* in Figure 43 on page 71).

An **SSCP ID** is not specified for *independent* type 6.2 LUs, because they are not controlled by an SNA host.

3. SNA LU6.2 Mode List Profile

This profile is required by the SNA LU6.2 Logical Connection Profile. It contains a list of Mode profiles that describe the characteristics of LU6.2 sessions associated with the connection.

Figure 53 shows our Mode List Profile:

```

                                Add SNA LU6.2 Mode List Profile

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

'TOP'                                'Entry Fields'
* PROFILE name                        'LU62VSE'
  Add profile names to list:
    Name 1                             'LU62PS'
    Name 2                             ''
    Name 3                             ''
    Name 4                             ''
    Name 5                             ''
    Name 6                             ''
    Name 7                             ''
    Name 8                             ''
    Name 9                             ''
    Name 10                            ''
    Name 11                            ''
'MORE...53'

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

Figure 53. LU62VSE Mode List Profile

The key parameters are:

- **PROFILE name.** The system uses this name to refer to the set of characteristics described in this profile.
- **Add profile name to list.** These fields contain the names of Mode Profiles available for use on the session which is associated with this list.

An entry which is identical to the Log Mode specified in the VSE/ESA ACF/VTAM definition (see Figure 9 on page 18) should be the first one in this list; this entry is the default.

4. SNA LU6.2 Mode Profile

This profile is required by the SNA LU6.2 Mode List Profile. It names an SNA mode, and defines the attributes of that mode. The mode defines the characteristics of sessions between the local and remote LUs specified in the corresponding Connection Profile. The name of the defined mode can be the same as the name of the Mode Profile that defines it (however, this is not required).

At least one (default) Mode Profile must exist, specifying the name of the mode table entry defined in the VTAM independent LU (usually specified in the DLOGMOD parameter of the LU definition, see Figure 42 on page 70). The names of the host VTAM mode table (see Figure 9 on page 18) and the AIX SNA Services/6000 mode (**LU62PS** in our case) **must** be the same. Additional Mode Profiles may be defined as required.

Figure 54 on page 83 shows our Mode Profile:

```

Change / Show SNA LU6.2 Mode Profile

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

                                'Entry Fields'
CURRENT profile name             LU62PS
NEW PROFILE name                 ''
MODE name                        'LU62PS'
Maximum number of SESSIONS (1-999) '8' #
Minimum contention WINNERS (0-499)  '4' #
Minimum contention LOSERS (0-500)   '4' #
Auto ACTIVATIONS limit (0-500)     '0' #
RECEIVE pacing (0-63)              '3' #
SEND pacing (0-63)                 '3' #
Maximum RU SIZE (256,288,...,3840) '2816' #
RECOVERY level                     no_reconnect +

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

```

Figure 54. LU62PS Mode Profile

The key parameters are:

- **PROFILE name.** The system uses this name to refer to the set of characteristics described in this profile.
- **MODE name.** This field associates the set of session characteristics that are described with a mode name used by SNA. This mode name **must** be defined consistently across the network.

A CICS/6000 transaction may specify a **MODE name** (not a Mode Profile Name) either in the **SNAModeName** attribute of the Transaction Definition (TD) for the executing transaction, or for DTP, by specifying a PROFILE option on an EXEC CICS ALLOCATE command.

5. SNA LU6.2 Transaction Program Name (TPN) List Profile

This profile is required by the SNA LU6.2 Local LU Profile. It consists of a list of TPN Profiles.

Figure 55 on page 84 shows the TPN List Profile we used:

```

                          Add SNA LU6.2 TPN List Profile

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

'TOP'                                     'Entry Fields'
* PROFILE name                            'LU62VSEt'
Add profile names to list:
  Name 1                                   'TDEFAULT'
  Name 2                                   ''                                         +
  Name 3                                   ''                                         +
  Name 4                                   ''                                         +
  Name 5                                   ''                                         +
  Name 6                                   ''                                         +
  Name 7                                   ''                                         +
  Name 8                                   ''                                         +
  Name 9                                   ''                                         +
  Name 10                                  ''                                         +
  Name 11                                  ''                                         +
'MORE...53'

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

```

Figure 55. LU62VSEt TPN List Profile

The key parameters are:

- **PROFILE name.** The system uses this name to refer to the set of characteristics described in this profile.
- **Add profile names to list.** These fields contain the names of TPN profiles available for use on the session associated with this list (refer to Figure 52 on page 81).

This profile should include all the profiles requested in the **TPNSNAProfile** entries of the CICS/6000 region's transaction definition (TD). This list normally contains only one member. We used the system supplied name, 'TDEFAULT'. Refer to the TD of our sample transaction in B.4, "Application Definition Examples" on page 131.

6. SNA LU6.2 Transaction Program Name (TPN) Profile

This profile is required by the SNA LU6.2 TPN List Profile. It describes the characteristics of all CICS/6000 transactions that may be started by a remote SNA host, that is by CICS/VSE in our case. The name of this profile must match the TD attribute **TPNSNAProfile**. Because CICS/6000 is responsible for starting transactions, only the conversation type, PIP data, PIP subfields, sync level, and the security fields are relevant to CICS/6000 configuration.

Figure 56 on page 85 shows our TPN Profile:

```

Change / Show SNA LU6.2 TPN Profile

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

' TOP'                                     ' Entry Fields'
CURRENT profile name                      TDEFAULT
NEW PROFILE name                          ''
Transaction program name is in HEXADECIMAL? no +
TRANSACTION program name                  ''
PIP data?                                  no +
    If yes, SUBFIELDS (0-99)              '0' #
CONVERSATION type                          either
RECOVERY level                             no_reconnect +
SYNC level                                 either
Full PATH to TPN executable               '/usr/lpp/sna'
MULTIPLE INSTANCES supported?             no +
User ID                                     '0' #
SERVER synonym name                       ''
RESTART action                             once +
COMMUNICATION type                         signals +
    If IPC, communication IPC queue key    '0' #
Standard INPUT file/device                 '/ dev/null'
Standard OUTPUT file/device                '/ dev/console'
Standard ERROR file/device                 '/ dev/console'
SECURITY Required                          none +
    If access,
        RESOURCE SECURITY ACCESS LIST profile ' RSRCDDEFAULT'
        (If no name entered, /etc/passwd used)
' BOTTOM'

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

```

Figure 56. TDEFAULT TPN Profile

The key parameter is:

- **PROFILE name.** The system uses this name to refer to the set of characteristics described in this profile.

In the profile, **CONVERSATION type** and **SYNC level** should both be set to **'either'**. This enables both sides to initiate the binding process.

7. SNA LU6.2 Remote Transaction Program Name (RTPN) List Profile

This profile is required by the SNA LU6.2 Logical Connection Profile. At least one RTPN List Profile is required which in turn has to contain the name of at least one RTPN Profile.

Figure 57 on page 86 shows the RTPN List Profile we used:

```

Change SNA LU6.2 RTPN List Profile

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

' TOP'                                     ' Entry Fields'
CURRENT profile name                       RDEFAULT
NEW PROFILE name                           ''
DELETE profile names from list (F4 to list) +
Add profile names to list:
Name 1                                     ' RDEFAULT'
Name 2                                     '' +
Name 3                                     '' +
Name 4                                     '' +
Name 5                                     '' +
Name 6                                     '' +
Name 7                                     '' +
Name 8                                     '' +
Name 9                                     '' +
' MORE...54'

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

```

Figure 57. RDEFAULT Remote TPN List Profile

The key parameters are:

- **PROFILE name.** The system uses this name to refer to the list of profile names specified in this profile.
- **Add profile names to list.** These fields contain the names of transaction program name profiles available for use on the session associated with this remote transaction program name list.

This profile is **not used by CICS/6000** connections, but is required by AIX SNA Services/6000.

8. SNA LU6.2 Remote Transaction Program Name (RTPN) Profile

This profile is required by the SNA LU6.2 RTPN List Profile. It describes the characteristics of remote transactions.

Figure 58 on page 87 shows our RTPN Profile:


```

Change / Show SNA LU6.2 RTPN Profile

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

CURRENT profile name          'Entry Fields'
NEW PROFILE name              RDEFAULT
RTPN name is in HEXADECIMAL?  ''
RTPN name                      'rtpn'
PIP data?                      no           +
CONVERSATION type             mapped       +
RECOVERY level                no_reconnect +
SYNC level                    none          +

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

```

Figure 58. RDEFAULT Remote TPN Profile

The key parameter is:

- **PROFILE name.** The system uses this name to refer to the set of characteristics described in this profile.

This profile is **not used by CICS/6000** connections, but is required by AIX SNA Services/6000.

11.3.4.3 Change Profiles for SNA Nodes

1. SNA Node Profile

This profile is required to start the SNA subsystem. The name of the SNA profile *must* be 'sna'.

Figure 59 on page 88 shows our SNA Node Profile:

```

Change / Show SNA Node Profile

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

                                'Entry Fields'
CURRENT profile name              sna
NEW PROFILE name                  ''
Total active open CONNECTIONS (1-5000)  '200'      #
Total SESSIONS (1-5000)            '200'      #
Total CONVERSATIONS (1-5000)       '200'      #
SERVER synonym name                'sna'
RESTART action                     once         +
Perform ERROR LOGGING?             no          +
Standard INPUT file/device         '/ dev/null'
Standard OUTPUT file/device        '/ dev/console'
Standard ERROR file/device         '/ dev/console'

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

```

Figure 59. SNA Profile

2. SNA Control Point Profile

This profile is required by the SNA Token-Ring Attachment Profile. It supplies PU information to configure the workstation as a node in a particular network.

At least one SNA Control Point Profile is required in a valid configuration. If more Control Point Profiles exist and are active at the same time on the same or different networks, then XID data and control point name (if needed) must be unique within the network.

Figure 60 on page 89 shows our SNA Control Point Profile Definitions:

```

Change / Show SNA Control Point Profile

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

CURRENT profile name          'Entry Fields'
NEW PROFILE name              VSE62
XID node ID                   ''
NETWORK name                  '07100000'
CONTROL POINT name            'DEIBMIPF'
                                'IPFCPX12'

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

```

Figure 60. VSE62 SNA Control Point Profile

The key parameters are:

- **PROFILE name.** The system uses this name to refer to the set of characteristics described in this profile.
- **XID node ID** is a hexadecimal value that provides the node ID of the physical unit, and is exchanged with the remote physical unit when a connection is first established.

It is composed of two fields:

- The first three hexadecimal digits must correspond to the *IDBLK=* parameter in the VTAM PU definition statement.
- The last five hexadecimal digits must correspond to the *IDNUM=* parameter in the VTAM PU definition statement.

Refer to Figure 42 on page 70 to cross-check these values.

11.3.5 CICS/6000 Customization for Inter System Communications

In order to communicate to or from the CICS/VSE system, you need to:

1. Specify, in the Region Definition (RD), the name of the network to which your region is attached.
2. Provide CICS/6000 Communication Definition (CD) entries that describe the connection to a remote system in an AIX file (stanza) for:
 - the SNA- connected system (CICS/VSE)
 - the Encina PPC Gateway

11.3.5.1 Region Definition (RD) Attributes for Intercommunication

Region Definitions are referred to by a key which represents the region name. CICS/6000 uses the region name as the LU name. Each LU belongs to a particular network. An LU can be uniquely identified by combining its LU name with the network name. In our example the region name is *CICS6K* and the network name *DEIBMIPF* resulting in a full LU identifier of *DEIBMIPF.CICS6K*.

The CICS/6000 region name must be the same as the **local_lu_name** defined to AIX SNA Services/6000. Therefore, the CICS/6000 region name **must** be specified in **upper case** and it must conform to your host LU naming conventions.

The RD attributes required for intercommunication are:

- **LocalSysId** = "ISC0"

This attribute contains the four character **SYSID** of the CICS/6000 region. This allows CICS/6000 applications to distinguish the local region from connected systems.

- **LocalNetworkName** = "DEIBMIPF"

This attribute contains the network name to which the local region is attached.

Note: The region name must be exactly the same (including case) as the name specified in the VTAM independent LU definition on the host (see B.2, "Definition Examples for CICS/6000 Region" on page 128).

11.3.5.2 Communications Definitions (CD) for SNA-Connected Systems

An AIX stanza file containing a **RemoteSysType** of **SNA** is required to define the SNA system, CICS/VSE, to which CICS/6000 will communicate. Since all SNA-connected systems must be routed through a gateway (the Encina PPC Gateway), each CD for SNA must name the local definition of the gateway (see 11.3.5.4, "Some Notes about CICS/6000 to CICS/VSE ISC" on page 94).

For more details about CICS/6000 resource definitions, refer to 12.3, "CICS Table Entries" on page 105.

We used SMIT to make the following CDs:

```

                                Add Communication

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

' TOP'                                ' Entry Fields'
* Communication Identifier            ' SNA1'
* Model Communication Identifier      ""
* Region name                          ' CICS6K'                                +
Add to database only OR Add and Install  Add                                +
Group to which resource belongs         ' tech'
Activate the resource at cold start?    yes                                +
Resource description                    ' CICS/VSE Definition'
* Number of updates                    0
Protect resource from modification?    no                                +
Timeout on allocate (in seconds)       ' 0'
Transaction Level Security Key Mask    ' none'
Resource Level Security Key Mask       ' none'
Remote system connection type(TCP, SNA or GWY)  SNA
Remote region name                      ' CICSSA22'
Gateway connection                      ' GWY1'
Network Name to which remote region is attached  ' DEIBMIPF'
SNA Profile name describing link to remote sys.  ' LU62VSE'
Code page of remote region for transaction routing  ' IBM-037'
DCE cell name of remote region(TCP)/gateway(SNA)  '/.:/'
DCE principal under which this gateway runs      ''
Inbound request security level           local                                +
UserId for link security on inbound requests     ''
Transmission encryption level            none                                +
Whether UserIds are sent on outbound requests   ' sent'
Set connection in service?               yes                                +

' BOTTOM'
F5=Reset          F6=Command          F7=Edit          F8=Image
F9=Shell          F10=Exit             Enter=Do

```

Figure 61. CD for SNA Connected System

The CD attributes required to define an SNA connection are:

- **AllocateTimeout= 0**
This attribute describes how many seconds a front end transaction waits for a backend transaction to be started over an SNA connection.
- **RemoteSysType= SNA**
This attribute must be **SNA** if this is a System Network Architecture connection.
- **RemoteLUName = "CICSSA22"**
This attribute specifies the LU name of the remote system. It must match the application identifier of the CICS/VSE region.
- **GatewayName = "GWY1"**
CICS/6000 requires that all SNA connections are routed through a PPC Gateway. This attribute identifies the CD stanza that defines the gateway.
- **RemoteNetworkName = "DEIBMIPF"**

This attribute specifies the name of the network to which the remote system is attached.

- **SNACConnectName** = "LU62VSE"

This attribute specifies the AIX SNA Services/6000 **Connection Profile** that describes the connection to a remote system via SNA.

- **RemoteCodePageTR** = "IBM-037"

This attribute specifies the code page of the remote region that CICS/6000 is to use when converting transaction routing data. For transaction routing activity to mainframe (EBCDIC) systems **IBM-037** can be used.

- **DCECell** = "/./"

This is the DCE cell name of the Encina PPC Gateway. If the gateway is running in the same cell as your CICS/6000 region, the value should be "/./".

A complete list of our CD definitions is provided in B.3, "Definition Examples for CICS/6000 Communication" on page 130.

11.3.5.3 Communications Definitions (CD) for the Encina PPC Gateway

Another CD stanza file is required to define the Encina PPC Gateway. This file is pointed to by the gateway name ('*GWY1*') specified in the **GatewayName** attribute of Figure 61 on page 91.

We used SMIT to make the following CDs:

```

                                Add Communication

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

' TOP'                                ' Entry Fields'
* Communication Identifier            ' GWY1'
* Model Communication Identifier      ""
* Region name                         ' CICS6K'                +
Add to database only OR Add and Install  Add                +
Group to which resource belongs        ' tech'
Activate the resource at cold start?    yes                +
Resource description                   ' Gateway Definition'
* Number of updates                   0
Protect resource from modification?     no                +
Timeout on allocate (in seconds)        ' 0'
Transaction Level Security Key Mask     ' none'
Resource Level Security Key Mask        ' none'
Remote system connection type(TCP, SNA or GWY)  GWY
Remote region name                     ' GWY1'
Gateway connection                     ""
Network Name to which remote region is attached ""
SNA Profile name describing link to remote sys. ""
Code page of remote region for transaction routing ' IBM-850'
DCE cell name of remote region(TCP)/gateway(SNA) '/.:/cics/ppc'
DCE principal under which this gateway runs ""
Inbound request security level          local                +
UserId for link security on inbound requests ""
Transmission encryption level          none                +
Whether UserIds are sent on outbound requests ' sent'
Set connection in service?             yes                +

' BOTTOM'
F5=Reset          F6=Command          F7=Edit          F8=Image
F9=Shell          F10=Exit           Enter=Do

```

Figure 62. CD for the Encina PPC Gateway

The CD attributes required to define a gateway are:

- **RemoteSysType=GWY**
This attribute must be **GWY** to specify that this stanza defines a gateway.
- **RemoteLUName = "GWY1"**
This attribute must be the same as the one specified in the **ppcgwy -n** argument when the gateway is started (see: 11.4, "Interconnection Operations" on page 95).
- **DCECell="/.:/cics/ppc"**
This attribute specifies the name of the Encina CDS root (that is ENCINA_CDS_ROOT) followed by **/ppc** (assuming that the gateway is running in the same cell as your CICS/6000 region).

Note: The gateway CD is used only for configuration purposes. Therefore, the gateway CD name ('GWY1) must not be used as a SYSID in intercommunication requests.

11.3.5.4 Some Notes about CICS/6000 to CICS/VSE ISC

1. Using Encina PPC Gateway for SNA Connections

The actual APPC partners in this communication process are CICS/VSE and the Encina PPC Gateway. Encina PPC Gateway is the APPC application, but it receives all configuration dependent information from definitions made in the CICS/6000 resource definitions (RD and CD), and AIX SNA Services/6000.

The Encina PPC Gateway dynamically registers itself at AIX SNA Services/6000 when CICS/6000 connects to it. CICS/6000 to Encina PPC Gateway communication is done via DCE's RPCs through the Encina PPC Executive.

CICS/6000 requires an active SNA connection between AIX SNA Services/6000 and the remote LU in order to be successfully registered at the Encina PPC Gateway

As described in 11.3.5.3, "Communications Definitions (CD) for the Encina PPC Gateway" on page 92, the gateway is defined via a CD stanza with a **RemoteSysType** of 'GWY'. The gateway CD provides the name ('GTWY1') by which it is referred to by the SNA connection CD.

Here are some other items to be considered when using the gateway:

- The gateway DCE name is specified in the gateway's CD **RemoteLUName** attribute. This name is specified by using the **-n** option at gateway startup (see 11.4, "Interconnection Operations" on page 95). The gateway uses this name and the value of the **ENCINA_CDS_ROOT** environment variable to register itself in the DCE CDS name space.

For example, if **ENCINA_CDS_ROOT=/:/cics**

then, **/:/** is the DCE Cell Name,

and **/cics/ppc/gateway** is the DCE CDS directory name.

Note: This example is cell dependent.

- The gateway 'advertises' the names of the SNA host via the DCE CDS for use by other potential systems.

2. Configuring for Inter-System Data Conversion

CICS/VSE to which CICS/6000 connects stores data in EBCDIC, not in ASCII as AIX does. This means that data, which is transferred between two systems, has to be converted in the appropriate way.

CICS products are able to handle most of the necessary data conversion tasks automatically. Some of these tasks require no special setup by the user, for example, handling file names in function shipping requests. For other differences, such as the ones regarding record structures in files, the user is responsible to provide definitions that identify the type of conversion to be done.

In general, conversion is always handled by the system where the data is stored.

If the conversion is not handled by the system, exits or user-replaceable exits or conversion programs are available.

Our sample application stores data in VSAM files controlled by CICS/VSE on the VSE/ESA host. Therefore, the CICS DFHCNV macro is used to handle data conversion.

For a description of how to code and generate the DFHCNV macro see, 4.2.1, “CICS/VSE Resource Definition Using Macro” on page 19, Data Conversion Program (DFHCNV).

For more information about Inter-System Data Conversion on a CICS/6000 system see, *AIX CICS/6000 Intercommunication*.

For more information about Inter-System Data Conversion refer to *Communicating with CICS OS/2, SC33-0736*.

3. Configuring CICS/6000 for Transaction Availability

There are two aspects of making a transaction available for intercommunication:

a. Over an SNA Connection

In order to service requests for FS, TR and DTP transactions for SNA-connected systems, the transactions have to be made available to the **Encina PPC Gateway**.

The CICS/6000's Transaction Definitions (TD) attribute **TPNSNAPProfile** names a Local Transaction Profile used by AIX SNA Services/6000 ('**TDEFAULT**' in our example) to determine the characteristics of the transaction to be started. If a profile is not specified with the TPNSNAPProfile attribute, then that transaction is not available for incoming requests.

This means you need to make sure that:

- CICS-supplied transactions which have to service requests for FS, TR and DTP (for example CVMI, CSMI, CPMI,...) have the **TPNSNAPProfile** attribute set to '**TDEFAULT**', which is a supplied TPN profile in AIX SNA Services/6000.
- The TPN profile called '**TDEFAULT**' is a member of the TPN List required by the Local LU Profile. Refer to our example transaction definition ('TECH') in B.4, “Application Definition Examples” on page 131.

b. Making a Backend DTP Transaction Available for TCP/IP and SNA Connections

CICS/6000 doesn't distinguish between transactions that are the backend partner of a DTP pair of transactions, and transactions that are implicitly routed. If a backend DTP transaction is used, the Transaction Definitions (TD) attribute **IsBackEndDTP** has to be set to '**yes**', otherwise your backend program will not start.

11.4 Interconnection Operations

We used the following sequence of operations to start CICS/6000 to CICS/VSE communication. This is not the only possible sequence in particular with regard to the startup and registration of the Encina PPC Gateway.

1. Activate VTAM including the PU and LU for CICS/6000:

```
d net,id=ipfcpx12,e
```

for VTAM start refer to A.2, “VTAM Start List” on page 116

2. Start AIX SNA Services/6000 using SMIT
3. Start the SNA Attachment ('VSE62) using SMIT

4. Start the AIX SNA Services/6000 Connection:

- On CICS/VSE enter:

```
CEMT SET CONN(ISC0) ACQUIRED
```

(This may be automatic if you shutdown/restart CICS/VSE.)

5. Verify SNA status, using SMIT or use the AIX 'lssrc' (get status of a subsystem) command:

```
# lssrc -ls sna
"sna"Program, Process ID <nnnn> active
VSE62          Attachment    - active
LU62VSE       Connection    - active
```

6. Start the Encina PPC Gateway

This requires 'root' authority. ENCINA_AUTHN must be set to 1 before starting the gateway (see 11.3.1, "AIX Environment Customization" on page 73).

To actually start the gateway enter:

```
# ppcgwy -Z0 -n GWY1 -p MyGtw -k /krb5/v5srvtab -f <any.out>
```

Where:

GWY1	is the name used to register the gateway in the CDS
MyGtw	is the gateway's DCE principal.
/krb5/v5srvtab	is the gateway's DCE keytab file.
any.out	is the name of the file the gateway uses to store its configuration.

If *-f <any.out>* is not specified, then the Encina PPC Gateway will use *./configFile* as the default configuration file.

Check the connection startup messages on your terminal. Successful startup is indicated by a '1 A Ready...' message.

7. Start your CICS/6000 region as described in 9.4.3, "Creating and Starting a CICS/6000 Region" on page 64.

8. Use the AIX 'tail' command for the */var/cics_regions/<reg_name>/console.msg* file to check the CICS startup messages (our *reg_name* is 'CICS6K'). Close to the end of the message file there will be messages like:

```
ERZ3039I... Configuring APPC Gateway with details of local system
ERZ3040I... Configuring APPC Gateway 'GWY1' with details of system
              'SNA1'
ERZ3041I... APPC Gateway configuration complete
```

9. Use 'tail' again to check in the *any.out* file from the *ppcgwy* command for messages regarding the transactions (TPNs) that are registered in the CICS/6000 startup. These messages might look as shown in Figure 63.

```
AddSysId .CICS6K PPC_CONN_TCP /./cics/CICS6K/ts <null> <null> ;
AddSysId .CICS6K.DEIBMIPF.CICSSA22 PPC_CONN_SNA LU62VSE <null> <null> ;
AddSysIdMap .CICS6K.DEIBMIPF.CICSSA22 .CICS6K ;
AddSNAPPrograms .CICS6K.DEIBMIPF.CICSSA22 TDEFAULT 4 CVMI CSMI CPMI CRTE ;
```

Figure 63. PPC's configFile

The 'AddSysId' messages identify the connections between the Encina PPC Gateway and the CICS/6000 region, and the connection between the remote (CICS/VSE) logical unit and the CICS/6000 region. The 'AddSysIdMap' identifies the *logical* connection between the CICS/6000 region and TPN's it wants to have registered. These are requested by specifying a TPN profile name ('*TDEFAULT*') in the TD for the region's transactions.

10. Use CICS ISC facilities between CICS/6000 and CICS/VSE.

11.5 Monitoring the Encina PPC Gateway

Encina provides an administration tool for the gateway which is invoked via the **ppcadmin** command. The following useful services are provided by this tool:

ppcadmin list conv	lists all conversations running at the gateway
ppcadmin list sysid	lists the configuration at the gateway
ppcadmin query conv	queries a conversation that is active at the gateway
ppcadmin query stat	queries the statistics kept by the gateway
ppcadmin stop server	stop a gateway

Table 5. Encina PPC Gateway Administration Tool Services

The corresponding command syntax can be obtained by entering: '*ppcadmin help*'.

For more about CICS/6000 intercommunication, refer to *AIX CICS/6000 Intercommunication Release 1, SC33-0815*.

11.6 Operational Hints

11.6.1 VSE Host Operational Hints

As mentioned earlier, connecting the CICS/6000 server to CICS/VSE is a complex task. Therefore some hints and tips are provided in A.1, "VSE Host Operation Hints" on page 113 if connection problems arise. Tracing VTAM from the host point of view has helped us in many cases to identify and solve the problem. An example of such a trace is therefore provided in A.3, "VTAM Trace Example" on page 117.

To produce the VTAM trace, shown in A.3, "VTAM Trace Example" on page 117, we changed the **AUtoconnect** attribute in the CICS/VSE connection definition to '**Yes**', see Figure 43 on page 71.

We then produced a trace using the following sequence of operations:

1. VTAM and AIX SNA Services were active. PU *IPFCPX12* was in *CONC* state.
2. We started trace (using similar instructions as shown in Figure 71 on page 115).
3. The activation of AIX SNA Services/6000's connection *VSE62* resulted in establishing a connection for PU *IPFCPX12*: The PU status changed to '*ACTIVE*'.

4. When the Encina PPC Gateway was started, CICS/VSE automatically "acquired" a connection with the RISC/6000 system. This resulted in an LU-LU (that is, *CICSSA22-CICS6K*) session establishment.
5. Then the CICS/6000 region *CICS6K* was started and our sample transaction "TECH" was entered. "TECH" performs a DPL to CICS/VSE (see 12.1, "Application Description" on page 103), that is, uses the CICS/6000 to CICS/VSE connection in both directions.
6. We stopped the trace and printed the trace results.

11.6.2 CICS/6000 Operational Hints

Although the CICS/6000 *CEMT* transaction has no ISC connection functions, *CECI* can be used to inquire the connection status as follows:

CECI INQUIRE CONNECTION(sysid) (sysid is *SNA1* in our example)

This lists the netname for (remote_lu) and the serverstatus.
'SERVERSTATUS=000000073' means that the connection is *in-service*.

11.6.3 AIX SNA Services/6000 Operational Hints

Either SMIT or the following commands can be used to control AIX SNA Services/6000 resources:

- For starting AIX SNA Services/6000 resources:
 - **startsrc -s sna**
Starts the *sna* subsystem.
 - **startsrc -t attachment -o VSE62**
Starts the attachment *VSE62*.
 - **startsrc -t connection -o LU62VSE**
Starts the connection *LU62VSE*. Instead of *startsrc*, you can establish the connection by
 - entering the transaction
CEMT SET CONN(ISC0) ACQ
at the CICS/VSE console or
 - starting the Encina PPC Gateway:
ppcgwy -Z0 -n GWY1 -p MyGtw -k /krb5/v5srvtab
on AIX.
- Displaying the status of AIX SNA Services/6000 resources:
 - **lssrc -l -s sna**
Displays the status of the resources of the AIX SNA Services/6000 subsystem called '*sna*'.
 - **lssrc -l -t attachment -o VSE62**
Displays the status of attachment '*VSE62*'.
 - **lssrc -l -t connection -o LU62VSE**
Displays the status of connection '*LU62VSE*'.
- Stopping AIX SNA Services/6000 resources:

- **stopsrc -t connection -o LU62VSE**
Stops the connection 'LU62VSE'.
- **stopsrc -t attachment -o VSE62**
Stops connection 'VSE62'.
- **stopsrc -s sna**
Stops the AIX SNA Services/6000 subsystem called 'sna'.

11.7 CICS/6000 to CICS/VSE Customization Summary

11.7.1 Reference of Implementation Tasks

The tables below list the items to be customized both for the VSE/ESA host and the CICS/6000 server together with a reference to the corresponding topic in this or other chapters.

VSE/ESA Host	Items to be customized	Details to be found in
Network Attachment	IOCDS and VM considerations	4.1.1, "IOCDS and VM Definitions for the 3172" on page 12
	Attach 3172 to VSE-Machine	4.1.2, "Attach 3172 to VSE Machine" on page 12
	Define 3172 in VSE/ESA	4.1.3, "3172 Definition in VSE/ESA" on page 13
	Customize 3172	4.1.4, "3172 Customization" on page 13
VTAM	VTAM XCA Major Node	4.1.5.1, "VTAM XCA Major Node" on page 15
	VTAM SWNET Major Node	4.1.5.2, "VSE/VTAM SWNET Major Node" on page 16
	VTAM Log Mode Entry	4.1.5.3, "VSE/VTAM Log Mode Entry" on page 17
	VTAM Appl. Major Node	4.1.5.4, "VTAM Application Major Node" on page 18
CICS/VSE Resources	DFHSIT	4.2.1.1, "System Initialization Table (SIT)" on page 19
	DFHTCP	4.2.1.2, "Terminal Control Program (DFHTCP)" on page 20
	DFHCNV	4.2.1.3, "Data Conversion Program (DFHCNV)" on page 21
	DFHFCT	4.2.1.4, "File Control Table (DFHFCT)" on page 21
	Connections	11.2.2.1, "Connection Definition" on page 71
	Sessions	11.2.2.2, "Session Definition" on page 72
	Programs	4.2.2.3, "Program Definition" on page 26

Table 6. VSE/ESA Connection Customization Summary

AIX Server	Items to be customized	Details to be found in
AIX	<i>/etc/environment</i> file	11.3.1, "AIX Environment Customization" on page 73
RISC/6000	TR-Adapter Hardware Address	11.3.2, "Token-Ring Adapter Customization" on page 74
DCE	For running PPC Gateway	11.3.3, "DCE Customization" on page 75
SNA Services/6000	Physical characteristics profiles	11.3.4.1, "Change Profiles for SNA Physical Units" on page 76
	- Attachment Profile	Figure 48 on page 77
	- Token Ring DLC Logical Link	Figure 49 on page 78
	- Token Ring DLC Physical Link	Figure 50 on page 79
	Logical characteristics profiles	11.3.4.2, "Change Profiles for SNA LU6.2 Units" on page 79
	- LU6.2 Logical Connection	Figure 51 on page 80
	- LU6.2 Local Logical Unit	Figure 52 on page 81
	- LU6.2 Mode List	Figure 53 on page 82
	- LU6.2 Mode	Figure 54 on page 83
	- LU6.2 Transaction Program Name List	Figure 55 on page 84
	- LU6.2 Transaction Program Name	Figure 56 on page 85
	- LU6.2 Remote Transaction Program Name List	Figure 57 on page 86
	- LU6.2 Remote Transaction Program Name	Figure 58 on page 87
	Nodes	11.3.4.3, "Change Profiles for SNA Nodes" on page 87
	- System Network Architecture	Figure 59 on page 88
- Control Point	Figure 60 on page 89	
CICS/6000	Customization for ISC	11.3.5, "CICS/6000 Customization for Inter System Communications" on page 89

Table 7. AIX Connection Customization Summary

11.7.2 CICS/6000 to CICS/VSE Name Cross Reference

The table below provides an overview on how the names used within CICS/6000 are reflected in (that is, have to match) other subsystems.

For equivalent cross references in SNA (ACF/VTAM) refer to the individual panel descriptions provided in 11.3.4, "AIX SNA Services/6000 Customization" on page 75 and in appendix Appendix B, "Definition Examples for CICS/6000 and AIX SNA Services/6000" on page 123.

CICS/6000 resource	Items	MUST match
RD	Region Name	<ul style="list-style-type: none"> • VTAM independent LU name • Netname in CICS/VSE Connection definition • LocalLUName in AIX SNA LU6.2 Local LU Profile
	LocalNetworkName	NETWORK name in AIX SNA LU6.2 Local LU Profile
CD Type SNA	RemoteLUName	<ul style="list-style-type: none"> • CICS/VSE SIT APPLID • VTAM APPL definition • RemoteLUName in AIX SNA LU6.2 Logical Connection Profile
	GatewayName	Communication Identifier in CICS/6000 CD Type GWY
	RemoteNetworkName	<ul style="list-style-type: none"> • VTAM's network name • NETWORK Name in AIX SNA LU6.2 Connection Profile
	SNACConnectName	AIX SNA LU6.2 Connection Profile name
	DCECell	AIX /etc/environment
CD Type GWY	RemoteLUName	ppcgwy -n <i>parameter</i>
	DCECell	AIX /etc/environment
TD	TPNSNAProfile	Local TPNProfile in AIX SNA LU6.2 TPN

Table 8. CICS/6000 Cross Reference Summary

Chapter 12. Implementing an Application in CICS/6000

This chapter describes the implementation of the distributed application described in Chapter 6, "Overview of our Sample Application" on page 49 for the CICS/6000 environment.

Unlike the implementation in CICS OS/2 the front-end program (TECCPROG) handles the interface to the user in CICS/6000 only via BMS.

This chapter provides the application relevant definitions required in CICS/6000, the front-end.

The definitions required in CICS/VSE, the backend, are described in 6.2, "Application Implementation on CICS/VSE" on page 50. Application's CICS/VSE definitions are described in 6.2, "Application Implementation on CICS/VSE" on page 50.

12.1 Application Description

From a conceptual point of view there is no difference between the application running in a CICS/6000 to CICS/VSE or CICS OS/2 to CICS/VSE environment. It is invoked via the same transaction, 'TECH', in CICS/6000. TECH calls the local application, TECCPROG, which interacts with the user via BMS maps (defined in the mapset NEW1MS).

Communication to CICS/VSE is done via DPL to program CICSVSE, defined as **remote** and running on CICS/VSE. CICSVSE performs the necessary file operations to provide the data requested by the users (see also 6.2.2, "Create VSAM Base File" on page 51). The CICSVSE program handles files TECHBAS, a VSAM key sequenced file, and TECHALT, an alternate index to TECHBAS. Files are defined in CICS/VSE, see 6.2.2, "Create VSAM Base File" on page 51.

This process involves the following CICS resources:

1. In CICS/6000:

- Transaction **TECH**
starts the application
- Program **TECCPROG**
invoked by transaction TECH; provides a BMS user interface and links to backend program CICSVSE via DPL
- Mapset **NEW1MS**
Contains all the maps used by TECCPROG

2. In CICS/VSE:

- Program **CICSVSE**
invoked by program TECCPROG via DPL; handles VSAM files and must be defined in CICS/6000 as 'remote'
- File **TECHBAS**
VSAM KSDS containing the data required by the application
- File **TECHALT**

12.2 Application Development*

This is how we organized our directories in AIX and CICS/6000 to implement the sample application:

- Source for application programs and maps
/home/userxy/source
- Executables (CICS/6000 application programs)
/var/cics_regions/CICS6K/bin
- Maps (CICS/6000 application physical maps)
/var/cics_regions/CICS6K/maps/En_Us

Then we installed Micro Focus COBOL/6000 for Unix compiler (refer to *Micro Focus COBOL for UNIX***, *Getting Started, April 1993* for information on how to do this).

- Generate the BMS maps:
 - Change directory to your working directory:
cd /home/userxy/source
 - Create file *new1ms.bms* with source map definitions
See C.4, “BMS Maps (File Name NEW1MS.MAP)” on page 152.
 - Enter ‘**cicsmap new1ms**’, to generate the physical and symbolic maps.
 - Copy the physical maps to the region map directory:
cp NEW1MS.map /var/cics_regions/CICS6K/maps/En_Us
- Translate, Compile, and Link the program
 - Change directory to your working directory:
cd /home/userxy/source
 - Create source program *teccprog.ccp*.
See C.1, “Front-end Program Source (File Name TECCPROG.CCP)” on page 133.
 - Enter ‘**cicstcl -eds teccprog**’, to generate executables.
 - Copy the executables to the *production* directory:
cp teccprog.gnt /var/cics_regions/CICS6K/bin

For more about application development in AIX CICS/6000 see *AIX CICS/6000 Application Programming Guide, SC33-0814* and *AIX CICS/6000 Application Programming Reference, SC33-0886*.

12.3 CICS Table Entries

For detailed information on how to define CICS/6000 resources see *AIX CICS/6000 Customization and Operation, GC33-0816*.

We used SMIT to define the application's resources to CICS/6000:

1. From an AIX window execute the command:
@ smitty cics
2. Select '**Manage CICS/6000 Regions**' and press enter.
3. Select '**Define Resources for a CICS/6000 Region**' and press enter.
4. Select '**Manage Resource(s)**' and press enter.
5. This brings us to the resource list (see Figure 64). From here a fixed sequence of steps needs to be followed for any resources to be added.

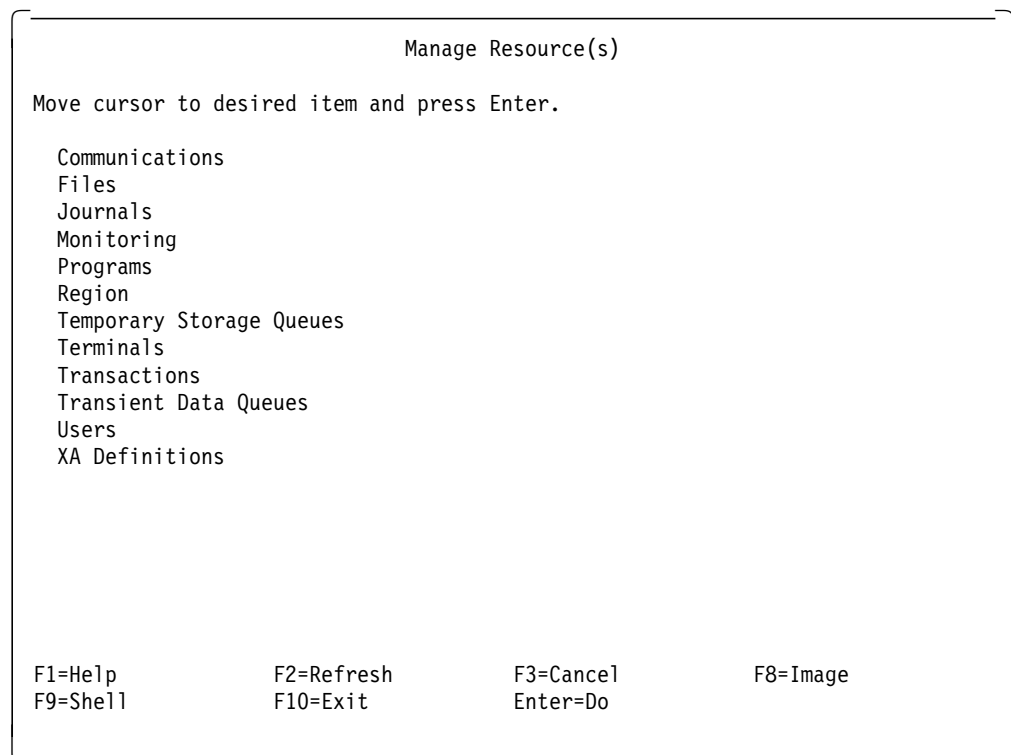


Figure 64. Manage CICS/6000 Resource(s)

Select the desired resource with the cursor and press 'Enter'.

6. Select '**Add New**' and press 'Enter'.
7. You are presented with a panel requesting the name of a model definition for the selected resource type. You may enter "" to indicate the use of the default resource entry.
8. When you press 'Enter' you receive the panel for the resource definition you selected. We will define the following resources:
 - Transaction 'TECH'
 - BMS mapset for 'TECCPROG'
 - Programs 'TECCPROG' (front-end) and 'CICSVSE' (backend)

12.3.1 Transaction Definition for Sample Transaction 'TECH'

```

                                Add Transaction

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

'TOP'                                'Entry Fields'
* Transaction Identifier              'TECH'
* Model Transaction Identifier        ""
* Region name                         'CICS6K'                                +
Add to database only OR Add and Install  Add                                +
Group to which resource belongs        'tech'
Activate the resource at cold start?    yes                                +
Resource description                   'Demo Transaction'
* Number of updates                   0
Protect resource from modification?    no                                +
Transaction enable status              enabled                              +
Remote System Identifier                ''
Remote Transaction Identifier           ''
Resource Level Security Key            'public'
Transaction Level Security Key         '1'
Type of RSL Checks                     none                                +
Type of TSL Checks                     internal                            +
Should transaction be dumped on an abend? no                                +
First program name                     'TECCPROG'
SNA TPN profile for APPC listener program ''
Is a user conversion template defined?  no                                +
Is back end of a DTP transaction?      no                                +
Enabled at shutdown?                   no                                +
Transaction Work Area Size              '0'                                #
Transaction Priority                    '0'                                #
Transaction purgeability                 purgeable                            +
Transaction deadlock timeout value (secs) '0'                                #
Effect of FORCEPURGE for InDoubt transactions wait_backout                +
Should data be converted to uppercase?  no                                +
SNA modename for this transaction      ''
Locally Queue ATI requests for this transaction? no                            +
Conversational timeout value (mins)    '0'                                #
Contexts in which transaction can START 'no_facility|output_ter>'

'BOTTOM'

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

```

Figure 65. Defining the Sample Transaction

The **First program name** can be specified in upper or lower case as CICS will internally change it to uppercase when it searches the program definitions.

12.3.2 BMS Mapset Definition for Sample Application

```

                                Add Program

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

' TOP'                                ' Entry Fields'
* Program Identifier                 ' NEW1MS'
* Model Program Identifier            ""
* Region name                         ' CICS6K'                               +
  Add to database only OR Add and Install  Add                               +
  Group to which resource belongs         ' tech'
  Activate resource at cold start?       yes                               +
  Resource description                   ' Demo Mapset'
* Number of updates                    0
  Protect resource from modifications?    no                               +
  Program enable status                  enabled                             +
  Remote system on which to run program   ''
  Name to use for program on remote system ''
  Resource Level Security Key            ' public'
  Program path name                      '/var/cics_regions/CICS>
  Program type                           map
  Is a user conversion template defined?  no                               +
' BOTTOM'

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

```

Figure 66. MEW1MS Mapset Definition

Pay attention to the following specifications:

- **Program path name**

The program path name includes the name of the physical map. COBOL mapset names are in upper case. You **must** specify *.map* extension for the COBOL mapsets.

- **Program type**

This specification tells CICS/6000 how to load the resource. For the mapset definitions it must be **map**.

12.3.3 Defining the Sample Front-end Program

```

                                Add Program

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

' TOP'                                ' Entry Fields'
* Program Identifier                 ' TECCPROG'
* Model Program Identifier           ' "'
* Region name                         ' CICS6K'                                +
Add to database only OR Add and Install
Add                                  ' Add'                                +
Group to which resource belongs      ' tech'
Activate resource at cold start?     ' yes'                                +
Resource description                 ' Front End Program'
* Number of updates                  0
Protect resource from modifications? ' no'                                +
Program enable status                ' enabled'                            +
Remote system on which to run program
Name to use for program on remote system
Resource Level Security Key          ' public'
Program path name                    ' teccprog'
Program type                          program
Is a user conversion template defined? ' no'                                +
' BOTTOM'

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

```

Figure 67. Defining the Sample Program TECCPROG

Pay attention to the following specifications:

- **Program path name**

The program path name includes the name of the executable. When the program executable is placed in the directories supplied by CICS/6000, only the program executable name is necessary. Do **not** specify the *.gnt* extension for COBOL programs.

- **Program type**

This specification tells CICS/6000 how to load the resource. For the program definitions it must be **program**.

12.3.4 Defining the Sample Backend Program

```

                                Add Program

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

' TOP'                                ' Entry Fields'
* Program Identifier                  ' CICSVSE'
* Model Program Identifier             ""
* Region name                         ' CICS6K'                               +
Add to database only OR Add and Install  Add                               +
Group to which resource belongs        ' tech'
Activate resource at cold start?       yes                               +
Resource description                   ' Back End Program'
* Number of updates                    0
Protect resource from modifications?    no                               +
Program enable status                   enabled                             +
Remote system on which to run program   ' SNA1'
Name to use for program on remote system ''
Resource Level Security Key            ' public'
Program path name                       ''
Program type                            program
Is a user conversion template defined?  no                               +
' BOTTOM'

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

```

Figure 68. Defining the Backend Program CICSVSE

Pay attention to the following specifications:

- **Remote system on which to run program**

You must enter the name of the remote region in which the program resides. You must include an entry of the remote region in the Communications Definitions (CD).

Part 4. Operation and Definition Examples

Appendix A. VSE Host Operation and Debugging

A.1 VSE Host Operation Hints

A.1.1 CEMT

Displaying the ISC Connection

- **CEMT INQUIR CONNECTION** command is used to display existing ISC connections.
- In our environment when all definitions are correct and necessary network elements are in service the following should be seen using the command:

```
I CONN
STATUS: RESULTS - OVERTYPE TO MODIFY
Conn(CICI) Net(IPFXL030)    Ins Acq
Conn(ISCO) Net(CICS6K )    Ins Acq
```

Figure 69. CEMT Inquiry Connection

A.1.2 VTAM Display Sessions

- **VTAM Display Sessions** command is also used to display existing ISC Sessions.

```
D NET,SESSIONS,LIST=ALL
1C39I COMMAND PASSED TO ACF/VTAM
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = SESSIONS
IST873I      PLU              SLU              SID              STATUS
IST874I DEIBMIPF.CICSSA22 DEIBMIPF.IPFXL030 C453D9C23F12EA03 ACTIV
IST874I DEIBMIPF.CICSSA22 DEIBMIPF.IPFXL030 C453D9C23F12EA02 ACTIV
IST874I DEIBMIPF.CICSSA22 DEIBMIPF.IPFXL030 C453D9C23F12EA01 ACTIV
IST874I DEIBMIPF.IPFXL030 DEIBMIPF.CICSSA22 CD07DA27364E9868 ACTIV
IST874I DEIBMIPF.CICSSA22 DEIBMIPF.IPFXL030 C453D9C23F12EA00 ACTIV
IST874I DEIBMIPF.CICSSA22 DEIBMIPF.IPFXL030 C453D9C23F12E9FF ACTIV
IST874I DEIBMIPF.IPFXL030 DEIBMIPF.CICSSA22 CD07DA27354E9868 ACTIV
.
.
IST874I DEIBMIPF.CICSSA22 DEIBMIPF.CICS6K    C453D9C23F12EA60 ACTIV
IST874I DEIBMIPF.CICSSA22 DEIBMIPF.CICS6K    C453D9C23F12EA5F ACTIV
IST874I DEIBMIPF.CICSSA22 DEIBMIPF.CICS6K    C453D9C23F12EA5E ACTIV
IST874I DEIBMIPF.CICSSA22 DEIBMIPF.CICS6K    C453D9C23F12EA5D ACTIV
IST874I DEIBMIPF.CICSSA22 DEIBMIPF.CICS6K    C453D9C23F12EA5C ACTIV
IST874I DEIBMIPF.CICS6K   DEIBMIPF.CICSSA22 C453D9C23F12EA5B ACTIV
IST878I NUMBER OF PENDING SESSIONS =      0
IST878I NUMBER OF ACTIVE   SESSIONS =     15
IST878I NUMBER OF QUEUED  SESSIONS =      0
IST878I NUMBER OF TOTAL   SESSIONS =     15
IST314I END
```

Figure 70. The Status of the Sessions

A.1.3 How to Trace with VTAM

- **VTAM Trace** facility may be used to find inconsistencies in customization. Here is the console log session demonstrating the use of VTAM tracing commands and trace printing.

The trace output produced from a similar session can be viewed in appendix A.3.1, "VTAM Trace to OS/2" on page 117.

```

F NET,TRACE,TYPE=BUF,ID=IPFCPX03 1
AR 015 1C39I COMMAND PASSED TO ACF/VTAM
F3 053 IST097I MODIFY ACCEPTED
F3 053 IST513I TRACE INITIATED FOR NODE IPFCPX03
F NET,TRACE,TYPE=BUF,ID=IPFXL030 1
AR 015 1C39I COMMAND PASSED TO ACF/VTAM
F3 053 IST097I MODIFY ACCEPTED
F3 053 IST513I TRACE INITIATED FOR NODE IPFXL030

F3 053 IST590I CONNECTIN ESTABLISHED FOR PU IPFCPX03 ON LINE L317209 2

F2 002 DFH4900I -99W CLS1 15:12:58 NODE IPFXL030 SYSTEM CICI MODENAME 3
LU62PS RECEIVED CNOS: MAX = 012, WIN = 008. SUCCESSFUL.

F NET,NOTRACE,TYPE=BUF,ID=IPFXL030
AR 015 1C39I COMMAND PASSED TO ACF/VTAM
F3 053 IST097I MODIFY ACCEPTED
F3 053 IST512I TRACE TERMINATED FOR NODE = IPFXL030
F NET,NOTRACE,TYPE=BUF,ID=IPFCPX03
AR 015 1C39I COMMAND PASSED TO ACF/VTAM
F3 053 IST097I MODIFY ACCEPTED
F3 053 IST512I TRACE TERMINATED FOR NODE = IPFCPX03
R RDR,PAUSEF5
AR 015 1C39I COMMAND PASSED TO VSE/POWER
F1 001 1R88I OK
F1 001 1Q47I F5 PAUSEF5 02010 FROM (SYSA) , TIME=15:14:46
F5 005 // JOB PAUSEF5
DATE 04/28/94,CLOCK 15/14/46
F5 005 // PAUSE
F5-005

5 EXEC TPRINT 4
F5 005 IST900I TRACE PRINT UTILITY STARTED
F5 005 IST907A SNAPSHOT MODE TPRINT? ENTER Y OR N
F5-005
5 N
F5 005 IST905A ENTER TRACE PRINT OPTIONS OR 'CANCEL'
F5-005
5 PRINT CLEAR=YES
F3 053 IST505I VTAM TRACE SUBTASK ENDED
F5 005 IST915I TRACE RECORDING SUSPENDED FOR ONLINE TPRINT
F5 005 IST917I RECORDS MISSING ON TRACE FILE
F3 053 IST504I VTAM TRACE SUBTASK STARTED
F5 005 IST910I TRACE PRINT UTILITY ENDED
F5 005 1I00D READY FOR COMMUNICATIONS.
F5-005
F3 062 4933D EQUAL FILE ID IN VTOC TRFILE SYS001=992 SYSWK1
VTAM.TRACE.FILE
F3-062
5
F5 005 E0J PAUSEF5
DATE 04/28/94,CLOCK 15/15/19,DURATION 00/00/33
F1 001 1Q34I F5 WAITING FOR WORK
62 DELETE

```

Figure 71. Console Log for VTAM Trace

Notes:

1 These are the commands to start the trace on both the PU (IPFCPX03) and the LU (IPFXL030). Because the LU is independent, specifying SCOPE=ALL on the PU trace will not record entries for the independent LU. SCOPE=ALL at the PU level causes trace entries to be recorded for dependent LUs only.

2 This message appears when the OS/2 Communication Manager has started and VTAM becomes aware that the physical unit is available.

3 This message appears when CICS OS/2 establishes sessions with CICS/VSE. This can occur by using CEMT to 'ACQUIRE' the sessions, or may happen automatically at the time of the first application request between the two CICS systems. In this case, the front-end application program on CICS OS/2 has issued a Distributed Program Link to the backend program to cause this session establishment. That is, the CICS/VSE host has **RECEIVED** a request to establish sessions from OS/2 Comms Manager. The case where CEMT is used to ACQUIRE the connection from the host side is shown in the console log of the trace example in Figure 72 on page 117.

4 TPRINT is the name of the program which formats and prints the trace data. In our case the DLBL info for the trace file is in the system standard labels and does not have to be specified here.

A.2 VTAM Start List

```
* $$ JOB JNM=BOESTR04,DISP=D,PRI=3, C
* $$ NTFY=YES, C
* $$ LDEST=*, C
* $$ CLASS=0
// JOB BOESTR04 CATALOG VTAM START OPTION LIST
// EXEC LIBR,PARM='MSHP'
ACCESS SUBLIB=PRD2.CONFIG
CATALOG ATCSTR04.B REPLACE=YES
SSCPID=22, *
HOSTSA=22, *
SSCPNAME=IPFV2B, *
HOSTPU=IPFVM22, *
NETID=DEIBMIPF, *
MAXSUBA=255, *
CONFIG=04, *
IOINT=0, *
SGALIMIT=1M, *
BSBUF=(28,,1), *
CRPLBUF=(60,,1), *
LFBUF=(300,288,,20), FROM 70 TO 120 *
LPBUF=(12,,6), *
SFBUF=(20,,20), *
SPBUF=(210,,32), *
VFBUF=204800, FROM 102400 TO 122880 (ADD 5P) *
VPBUF=528384, FROM 446464 TO 528384 (ADD 20P) *
XDBUF=(6,,1)
/+
/*
/&
* $$ EOJ
```

A.3 VTAM Trace Example

The following figure is an example of the trace data collected for a 'TYPE=BUF' VTAM trace of the independent LU defined in our scenario as the CICS OS/2 system. The trace data you see here shows the activity between VTAM and OS/2 Comms Manager as a result of using the CEMT transaction on the VSE/ESA host to 'ACQUIRE' the connection to this independent LU named IPFXL030. The first portion of the figure is the console log.

Note the session IDs displayed with the VTAM DISPLAY command. Also note that CICSSA22 is the primary logical unit (STATUS ACTIV-P) for these sessions as a result of using CEMT ACQUIRE.

A.3.1 VTAM Trace to OS/2

```

***** CONSOLE LOG *****
11 F3 053 IST590I CONNECTIN ESTABLISHED FOR PU IPFCPX03 ON LINE L31720A
12 F2 002 DFH4900I -99X CLS1 8:43:45 NODE IPFXL030 SYSTEM CICI MODENAME
13 LU62PS SENT CNOS: MAX = 012, WIN = 004. SUCCESSFUL.
////////////////////////////////////
14 D NET, ID=IPFXL030, E
15 AR 015 1C39I COMMAND PASSED TO ACF/VTAM
16 F3 053 IST097I DISPLAY ACCEPTED
17 F3 053 IST075I NAME = IPFXL030, TYPE = CDRSC
18 F3 053 IST486I STATUS= ACT/S, DESIRED STATE= ACTIV
19 F3 053 IST977I MDLTAB=***NA*** ASLTAB=***NA***
20 F3 053 IST861I MODETAB=CICSIPTM USSTAB=***NA*** LOGTAB=***NA***
01 F3 053 IST934I DLOGMOD=LU62PS USS LANGTAB=***NA***
02 F3 053 IST597I CAPABILITY-PLU ENABLED ,SLU ENABLED ,SESSION LIMIT NONE
03 F3 053 IST231I CDRSC MAJOR NODE = ISTDILU
04 F3 053 IST1044I ALSLIST = IPFCPX03
05 F3 053 IST082I DEVTYPE = INDEPENDENT LU / CDRSC
06 F3 053 IST654I I/O TRACE = OFF, BUFFER TRACE = ON
07 F3 053 IST171I ACTIVE SESSIONS = 0000000005, SESSION REQUESTS = 0000000000
08 F3 053 IST206I SESSIONS:
09 F3 053 IST1081I ADJACENT LINK STATION = IPFCPX03
10 F3 053 IST634I NAME STATUS SID SEND RECV VR TP NETID
11 F3 053 IST635I CICSSA22 ACTIV-P C453D9C23F12EBF4 0000 0001 0 0 DEIBMIPF
12 F3 053 IST635I CICSSA22 ACTIV-P C453D9C23F12EBF3 0000 0001 0 0 DEIBMIPF
13 F3 053 IST635I CICSSA22 ACTIV-P C453D9C23F12EBF2 0000 0001 0 0 DEIBMIPF
14 F3 053 IST635I CICSSA22 ACTIV-P C453D9C23F12EBF1 0000 0001 0 0 DEIBMIPF
15 F3 053 IST635I CICSSA22 ACTIV-P C453D9C23F12EBF0 0001 0002 0 0 DEIBMIPF
16 F3 053 IST314I END

***** END OF CONSOLE LOG *****

***** TRACE DATA *****
// JOB PAUSEF5 DATE 05/06/94,CLOCK 08/45/52
// PAUSE
EXEC TPRINT

VTAM TRACE FILE UTILITY 94.126 08:46:02 PAGE 1

00015 PHYSICAL RECORDS MISSING FROM TRACE FILE
BUF IPFXL030/CICSSA22 94.126/07:43:44.86 OUTBOUND
VTAM TH=400000000000000000000000016000000161D0000500000703880092 RH=6B8000
31001307 B0B050B3 00818585 82000602 00000000 00000000 4F000011 C4C5C9C2 .....&..AEEB.....|...DEIB
D4C9D7C6 4BC3C9C3 E2E2C1F2 F2280009 02E2D5C1 E2E5C3D4 C7090300 3BF2A4D5 MIPF.CICSSA22...SNASVCMG....2UN
1 4ECC0012 04C4C5C9 C2D4C9D7 C64BC3C9 C3E2E2C1 F2F20011 C4C5C9C2 D4C9D7C6 +...DEIBMIPF.CICSSA22..DEIBMIPF
4BC9D7C6 E7D3F0F3 F06018C4 53D9C23F 12EBF00F C4C5C9C2 D4C9D7C6 4BC9D7C6 .IPFXL030-.D.RB...O.DEIBMIPF.IPF
E5F2C22C 0A010840 40404040 404040 V2B....

```

Figure 72 (Part 1 of 3). VTAM Trace Data for CICS OS/2

```

BUF  CICSSA22/IPFXL030  94.126/07:43:44.93          INBOUND
VTAM  TH=400000000000000000000000000000000016000000161D000007005003880064  RH=EB8000
31001307  B0B050B3  00808585  80000602  00000000  00000010  23400000  28000902  .....&...EE.....
E2D5C1E2  E5C3D4C7  0903003B  F2A4D54E  CC001205  C4C5C9C2  D4C9D7C6  4BC9D7C6  SNASVCMG...2UN+...DEIBMIPF.IPF
E7D3F0F3  F0000060  18C453D9  C23F12EB  F00FC4C5  C9C2D4C9  D7C64BC9  D7C6E5F2  XLO30-.D.RB...O.DEIBMIPF.IPFV2
C2                                               B

BUF  IPFXL030/CICSSA22  94.126/07:43:45.20          OUTBOUND
USER  40000000  00
BUF  IPFXL030/CICSSA22  94.126/07:43:45.20          OUTBOUND
VTAM  TH=400000000000000000000000000000000016000000161C000050000700010008  RH=4B9101
04000600  00
BUF  CICSSA22/IPFXL030  94.126/07:43:45.20          INBOUND
VTAM  TH=400000000000000000000000000000000016000000161D000007005000000006  RH=830100
000002
BUF  IPFXL030/CICSSA22  94.126/07:43:45.23          OUTBOUND
USER  00171210  02000000  00000C00  04000000  06D3E4F6  F2D7E24B  C3C9C3E2  E2C1F2F2  .....LU62PS.CICSSA22
3BF2A4ED  89C60001  0460F9F9  E7000000  00000000  00000000  00000000  00000000  .2U.IF...-99X.....
00000000
BUF  IPFXL030/CICSSA22  94.126/07:43:45.24          OUTBOUND
VTAM  TH=400000000000000000000000000000000016000000161C000050000700020047  RH=0B95A0
2D0502FF  0003D000  000206F1  001A11C4  C5C9C2D4  C9D7C64B  C3C9C3E2  E2C1F2F2  .....1...DEIBMIPF.CICSSA22
3BF2A4ED  89C60001  0460F9F9  E7001712  10020000  0000000C  00040000  0006D3E4  .2U.IF...-99X.....LU
F6F2D7E2  62PS
BUF  CICSSA22/IPFXL030  94.126/07:43:45.25          INBOUND
VTAM  TH=400000000000000000000000000000000016000000161D000007005000000006  RH=830100
000002
BUF  CICSSA22/IPFXL030  94.126/07:43:45.29          INBOUND
VTAM  TH=400000000000000000000000000000000016000000161C00000700500001001A  RH=039101
00171210  0A000000  00000C00  04000000  06D3E4F6  F2D7E2          .....LU62PS
BUF  CICSSA22/IPFXL030  94.126/07:43:45.50          INBOUND
USER  00171210  0A000000  00000C00  04000000  06D3E4F6  F2D7E2          .....LU62PS
BUF  IPFXL030/CICSSA22  94.126/07:43:45.50          OUTBOUND
VTAM  TH=400000000000000000000000000000000016000000161D000050000700010006  RH=830100
007FFF

VTAM TRACE FILE UTILITY
94.126  08:46:02  PAGE  2

BUF  IPFXL030/CICSSA22  94.126/07:43:45.90          OUTBOUND
VTAM  TH=400000000000000000000000000000000016000000161D000050004C03890090  RH=6B8000
31001307  B0B050B3  00818989  82000602  00000000  00000000  4F000011  C4C5C9C2  .....&...AIIB.....|...DEIB
D4C9D7C6  4BC3C9C3  E2E2C1F2  F2260007  02D3E4F6  F2D7E209  03003BF2  A5D3AC1C  MIPF.CICSSA22...LU62PS...2VL...
001204C4  C5C9C2D4  C9D7C64B  C3C9C3E2  E2C1F2F2  0011C4C5  C9C2D4C9  D7C64BC9  ...DEIBMIPF.CICSSA22..DEIBMIPF.I
D7C6E7D3  F0F3F060  18C453D9  C23F12EB  F10FC4C5  C9C2D4C9  D7C64BC9  D7C6E5F2  PFXL030-.D.RB...1.DEIBMIPF.IPFV2
C22C0A01  08404040  40404040  40                                               B....
BUF  CICSSA22/IPFXL030  94.126/07:43:45.95          INBOUND
VTAM  TH=400000000000000000000000000000000016000000161D00004C005003890062  RH=EB8000
31001307  B0B050B3  00808989  80000602  00000000  00000010  23400000  26000702  .....&...II.....
D3E4F6F2  D7E20903  003BF2A5  D3AC1C00  1205C4C5  C9C2D4C9  D7C64BC9  D7C6E7D3  LU62PS...2VL...DEIBMIPF.IPFXL
F0F3F000  006018C4  53D9C23F  12EBF10F  C4C5C9C2  D4C9D7C6  4BC9D7C6  E5F2C2  030...-D.RB...1.DEIBMIPF.IPFV2B
BUF  IPFXL030/CICSSA22  94.126/07:43:45.95          OUTBOUND
VTAM  TH=400000000000000000000000000000000016000000161D0000500056038A0090  RH=6B8000
31001307  B0B050B3  00818989  82000602  00000000  00000000  4F000011  C4C5C9C2  .....&...AIIB.....|...DEIB
D4C9D7C6  4BC3C9C3  E2E2C1F2  F2260007  02D3E4F6  F2D7E209  03003BF2  A5D6C5A4  MIPF.CICSSA22...LU62PS...2VOEU
001204C4  C5C9C2D4  C9D7C64B  C3C9C3E2  E2C1F2F2  0011C4C5  C9C2D4C9  D7C64BC9  ...DEIBMIPF.CICSSA22..DEIBMIPF.I
D7C6E7D3  F0F3F060  18C453D9  C23F12EB  F20FC4C5  C9C2D4C9  D7C64BC9  D7C6E5F2  PFXL030-.D.RB...2.DEIBMIPF.IPFV2
C22C0A01  08404040  40404040  40                                               B....
BUF  IPFXL030/CICSSA22  94.126/07:43:45.99          OUTBOUND
VTAM  TH=400000000000000000000000000000000016000000161D0000500054038B0090  RH=6B8000
31001307  B0B050B3  00818989  82000602  00000000  00000000  4F000011  C4C5C9C2  .....&...AIIB.....|...DEIB
D4C9D7C6  4BC3C9C3  E2E2C1F2  F2260007  02D3E4F6  F2D7E209  03003BF2  A5E77882  MIPF.CICSSA22...LU62PS...2VX.B
001204C4  C5C9C2D4  C9D7C64B  C3C9C3E2  E2C1F2F2  0011C4C5  C9C2D4C9  D7C64BC9  ...DEIBMIPF.CICSSA22..DEIBMIPF.I
D7C6E7D3  F0F3F060  18C453D9  C23F12EB  F30FC4C5  C9C2D4C9  D7C64BC9  D7C6E5F2  PFXL030-.D.RB...3.DEIBMIPF.IPFV2
C22C0A01  08404040  40404040  40                                               B....

```

Figure 72 (Part 2 of 3). VTAM Trace Data for CICS OS/2


```

BUF  CICSSA22/IPFXL030  94.126/07:43:45.99          INBOUND
VTAM  TH=400000000000000000000000016000000161D0000560050038A0062  RH=EB8000
9  31001307  B0B050B3  00808989  80000602  00000000  00000010  23400000  26000702  .....&...II.....
D3E4F6F2  D7E20903  003BF2A5  D6C5A400  1205C4C5  C9C2D4C9  D7C64BC9  D7C6E7D3  LU62PS...2VOEU...DEIBMIPF.IPFXL
FOF3F000  006018C4  53D9C23F  12EBF20F  C4C5C9C2  D4C9D7C6  4BC9D7C6  E5F2C2  030...D.RB...2.DEIBMIPF.IPFV2B
BUF  IPFXL030/CICSSA22  94.126/07:43:46.02          OUTBOUND
VTAM  TH=400000000000000000000000016000000161D0000500062038C0090  RH=6B8000
10 31001307  B0B050B3  00818989  82000602  00000000  00000000  4F000011  C4C5C9C2  .....&...AIIB.....|...DEIB
D4C9D7C6  4BC3C9C3  E2E2C1F2  F2260007  02D3E4F6  F2D7E209  03003BF2  A5F07AE8  MIPF.CICSSA22...LU62PS...2VO.Y
001204C4  C5C9C2D4  C9D7C64B  C3C9C3E2  E2C1F2F2  0011C4C5  C9C2D4C9  D7C64BC9  ...DEIBMIPF.CICSSA22..DEIBMIPF.I
D7C6E7D3  F0F3F060  18C453D9  C23F12EB  F40FC4C5  C9C2D4C9  D7C64BC9  D7C6E5F2  PFXL030-.D.RB...4.DEIBMIPF.IPFV2
C2C0A01  08404040  40404040  40          B....
BUF  CICSSA22/IPFXL030  94.126/07:43:46.03          INBOUND
VTAM  TH=400000000000000000000000016000000161D0000540050038B0062  RH=EB8000
11 31001307  B0B050B3  00808989  80000602  00000000  00000010  23400000  26000702  .....&...II.....
D3E4F6F2  D7E20903  003BF2A5  E7788200  1205C4C5  C9C2D4C9  D7C64BC9  D7C6E7D3  LU62PS...2VX.B...DEIBMIPF.IPFXL
FOF3F000  006018C4  53D9C23F  12EBF30F  C4C5C9C2  D4C9D7C6  4BC9D7C6  E5F2C2  030...D.RB...3.DEIBMIPF.IPFV2B
BUF  CICSSA22/IPFXL030  94.126/07:43:46.06          INBOUND
VTAM  TH=400000000000000000000000016000000161D0000620050038C0062  RH=EB8000
12 31001307  B0B050B3  00808989  80000602  00000000  00000010  23400000  26000702  .....&...II.....
D3E4F6F2  D7E20903  003BF2A5  F07AE800  1205C4C5  C9C2D4C9  D7C64BC9  D7C6E7D3  LU62PS...2VO.Y...DEIBMIPF.IPFXL
FOF3F000  006018C4  53D9C23F  12EBF40F  C4C5C9C2  D4C9D7C6  4BC9D7C6  E5F2C2  030...D.RB...4.DEIBMIPF.IPFV2B
VTAM TRACE FILE UTILITY          94.126  08:46:02  PAGE 3
BUF  IPFXL030/CICSSA22  94.126/07:43:46.33          OUTBOUND
USER  40000000  00          ....
BUF  IPFXL030/CICSSA22  94.126/07:43:46.33          OUTBOUND
VTAM  TH=400000000000000000000000016000000161C000050004C00010008  RH=4B9101
04000600  00          .....
BUF  CICSSA22/IPFXL030  94.126/07:43:46.34          INBOUND
VTAM  TH=400000000000000000000000016000000161D00004C0050000000006  RH=830100
000002          ...
BUF  IPFXL030/CICSSA22  94.126/07:43:46.35          OUTBOUND
USER  40000000  00          ....
BUF  IPFXL030/CICSSA22  94.126/07:43:46.35          OUTBOUND
VTAM  TH=400000000000000000000000016000000161C000050005600010008  RH=4B9101
04000600  00          .....
BUF  CICSSA22/IPFXL030  94.126/07:43:46.36          INBOUND
VTAM  TH=400000000000000000000000016000000161D0000560050000000006  RH=830100
000002          ...
BUF  IPFXL030/CICSSA22  94.126/07:43:46.38          OUTBOUND
USER  40000000  00          ....
BUF  IPFXL030/CICSSA22  94.126/07:43:46.38          OUTBOUND
VTAM  TH=400000000000000000000000016000000161C000050005400010008  RH=4B9101
04000600  00          .....
BUF  CICSSA22/IPFXL030  94.126/07:43:46.39          INBOUND
VTAM  TH=400000000000000000000000016000000161D0000540050000000006  RH=830100
000002          ...
BUF  IPFXL030/CICSSA22  94.126/07:43:46.40          OUTBOUND
USER  40000000  00          ....
BUF  IPFXL030/CICSSA22  94.126/07:43:46.41          OUTBOUND
VTAM  TH=400000000000000000000000016000000161C000050006200010008  RH=4B9101
04000600  00          .....
BUF  CICSSA22/IPFXL030  94.126/07:43:46.44          INBOUND
VTAM  TH=400000000000000000000000016000000161D0000620050000000006  RH=830100
000002          ...
NO IO RECORDS FOUND IN TRACE FILE
NO LINE RECORDS FOUND IN TRACE FILE
NO TNST RECORDS FOUND IN TRACE FILE
NO VIT RECORDS FOUND IN TRACE FILE
END OF TRACE PRINT UTILITY
EOJ PAUSEF5          DATE 05/06/94,CLOCK 08/46/23,DURATION 00/00/31

```

Figure 72 (Part 3 of 3). VTAM Trace Data for CICS OS/2

Notes:

- 1** This is the first message from VTAM to OS/2 Comms Manager. It is a BIND request (first byte x'31') to the OS/2 System Control Point. This session will be used to control and keep track of the other four sessions which are about to be started. Notice the session ID underlined matches the last session displayed in

the console log. You will notice as well that the mode named SNASVCMG is used for this control session.

2 This is Comms Manager's response to the bind for the control session.

3 This is a request from a USER (in this case the user is host CICS) to ALLOCATE a conversation. Notice that the request includes the name LU62PS which is the mode entry we have set up to rule our CICS to CICS sessions.

4 VTAM translates the ALLOCATE from CICS into an SNA FMH-5 request and sends it to Comms Manager.

5 This is the bind request for the second session between VTAM and Comms Manager. It will use mode entry LU62PS.

6 This is Comms Manager's response to the second session bind request.

7 and **9** are the bind request and response for a third session.

8 and **11** are the bind request and response for a fourth session.

10 and **12** are the bind request and response for a fifth session.

A.4 VSE Virtual Machine Directory

```
USER V133A80K PASSWORD 0032M 64M G
ACCOUNT V133A80K V133A80K
OPTION MAXCON 150
MACHINE ESA
IPL CMS
CONSOLE 0009 3215 T
SPOOL 000C 2540 READER *
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A
SPECIAL 080 3270
SPECIAL 081 3270
SPECIAL 082 3270
SPECIAL 083 3270
SPECIAL 084 3270
SPECIAL 085 3270
SPECIAL 086 3270
SPECIAL 087 3270
SPECIAL 088 3270
SPECIAL 089 3270
SPECIAL 800 CTCA VTAM
DEDICATE 300 0300
DEDICATE 301 0301
DEDICATE 302 0302
DEDICATE 303 0303
DEDICATE 960 2960
MDISK 191 3380 001 049 DISK01 MW ALL
MDISK 192 3380 001 049 DISK02 MW ALL
LINK MAINT 190 0190 RR
LINK MAINT 19D 019D RR
LINK MAINT 19E 019E RR
MDISK 991 3380 1770 885 DISK01 MW RVSE WVSE MVSE
MDISK 992 3380 1770 885 DISK02 MW RVSE WVSE MVSE
MDISK 993 3380 885 885 DISK01 MW RVSE WVSE MVSE
MDISK 994 3380 885 885 DISK02 MW RVSE WVSE MVSE
MDISK 995 3380 50 835 DISK01 MW RVSE WVSE MVSE
MDISK 996 3380 50 835 DISK02 MW RVSE WVSE MVSE
MDISK 997 3380 0 2655 DISK01 MW RVSE WVSE MVSE
MDISK 998 3380 0 2655 DISK02 MW RVSE WVSE MVSE
```

Appendix B. Definition Examples for CICS/6000 and AIX SNA Services/6000

B.1 AIX SNA Services/6000 Profiles

* PROFILE TYPE BEING PRINTED: "CONNECTION"

LU62VSE_CONNECTION:

```
type = CONNECTION
profile_name = LU62VSE
attachment_profile_name = VSE62
local_lu_profile_name = CICS6K
network_name = DEIBMIPF
remote_lu_name = CICS6K
stop_connection_on_inactivity = no
lu_type = lu6.2
interface_type = extended
remote_tpn_list_name = RDEFAULT
mode_list_name = LU62VSE
node_verification = no
inactivity_timeout_value = 0
notify = no
parallel_sessions = parallel
negotiate_session_limits = yes
security_accepted = none
conversation_security_access_list_name = CONVDEFAULT
```

* PROFILE TYPE BEING PRINTED: "LOCAL LU"

CICS6K_LOCALLU:

```
type = LOCALLU
profile_name = CICS6K
local_lu_name = CICS6K
network_name = DEIBMIPF
lu_type = lu6.2
independent_lu = yes
tpn_list_name = LU62VSEt
local_lu_address = 1
sscp_id = *
number_of_rows = 24
number_of_columns = 80
```

* PROFILE TYPE BEING PRINTED: "ATTACHMENT"

VSE62_ATTACHMENT:

```
type = ATTACHMENT
profile_name = VSE62
control_point_profile_name = VSE62
logical_link_profile_name = VSE62
physical_link_profile_name = VSE62
logical_link_type = token_ring
restart_on_deactivation = no
stop_attachment_on_inactivity = no
station_type = secondary
physical_link_type = token_ring
remote_secondary_station_address = 1
smart_modem_command_sequence =
length_of_command_sequence = 0
call_type = call
x25_level = 1984
listen_name = IBMQLLC
autolisten = yes
timeout_value = 0
remote_link_name_ethernet =
remote_link_name_token_ring =
remote_link_address = 400020201003
selection_sequence =
length_of_selection_sequence = 0
network_type = switched
access_routing = link_address
remote_sap_address = 04
remote_sap_address_range_lower = 04
remote_sap_address_range_upper = EC
virtual_circuit_type = permanent
remote_station_X.25_address =
optional_X.25_facilities = no
logical_channel_number_of_PVC = 1
reverse_charging = no
rpoa = no
default_packet_size = no
default_window_size = no
default_throughput_class = no
closed_user_group = no
closed_user_group_outgoing = no
network_user_id = no
network_user_id_name =
data_network_identification_code =
packet_size_for_received_data = 128
packet_size_for_transmit_data = 128
window_size_for_received_data = 2
window_size_for_transmit_data = 2
throughput_class_for_received_data = 9600
throughput_class_for_transmit_data = 9600
index_to_selected_closed_user_group = 0
lu_address_registration = no
lu_address_registration_name = LDEFAULT
```

* PROFILE TYPE BEING PRINTED: "SNA"

sna_SNA:

```
type = SNA
profile_name = sna
total_active_open_connections = 200
total_sessions = 200
total_conversations = 200
server_synonym_name = sna
nmvt_action_when_no_nmvt_process = reject
restart_action = once
stdin = /dev/null
stdout = /dev/console
stderr = /dev/console
sna_error_log = no
```

* PROFILE TYPE BEING PRINTED: "REMOTE TPN"

RDEFAULT_REMOTETPN:

```
type = REMOTETPN
profile_name = RDEFAULT
tpn_name = rtpn
tpn_name_hex = 99A39795
pip_data = no
conversation_type = basic
recovery_level = no_reconnect
sync_level = none
tpn_name_in_hex = no
```

* PROFILE TYPE BEING PRINTED: "REMOTE TPN LIST"

RDEFAULT_REMOTETPNLIST:

```
type = REMOTETPNLIST
Listname = RDEFAULT
list_members = RDEFAULT
```

* PROFILE TYPE BEING PRINTED: "TPN"

TDEFAULT_TPN:

```
type = TPN
profile_name = TDEFAULT
tpn_name = tpn
tpn_name_hex = A39795
conversation_type = either
pip_data = no
sync_level = either
recovery_level = no_reconnect
full_path_to_tpn_executable = /usr/lpp/sna
multiple_instances = no
user_id = 0
server_synonym_name = TDEFAULT
restart_action = once
communication_type = signals
```

```
stdin = /dev/null
stdout = /dev/console
stderr = /dev/console
subfields = 0
communication_ipc_queue_key = 0
tpn_name_in_hex = no
security_required = none
resource_security_access_list_name = RSRCDEFAULT
```

* PROFILE TYPE BEING PRINTED: "TPN LIST"

```
LU62VSEt_TPNLIST:
  type = TPNLIST
  listname = LU62VSEt
  list_members = TDEFAULT
```

* PROFILE TYPE BEING PRINTED: "CONTROL POINT"

```
VSE62_CONTROLPOINT:
  type = CONTROLPOINT
  profile_name = VSE62
  xid_node_id = 07100000
  network_name = DEIBMIPF
  cp_name = IPFCPX12
```

* PROFILE TYPE BEING PRINTED: "MODE"

```
LU62PS_MODE:
  type = MODE
  profile_name = LU62PS
  mode_name = LU62PS
  maximum_number_of_sessions = 8
  minimum_contention_winners = 4
  minimum_contention_losers = 0
  auto_activations_limit = 0
  receive_pacing = 3
  send_pacing = 3
  maximum_ru_size = 2816
  recovery_level = no_reconnect
```

* PROFILE TYPE BEING PRINTED: "MODE LIST"

```
LU62VSE_MODELIST:
  type = MODELIST
  listname = LU62VSE
  list_members = LU62PS
```


* PROFILE TYPE BEING PRINTED: "TOKEN RING LOGICAL"

VSE62_TOKENRINGLOGICAL:

```
type = TOKENRINGLOGICAL
profile_name = VSE62
retry_limit = 20
transmit_window_count = 10
dynamic_window_increment = 1
retransmit_count = 8
receive_window_count = 127
ring_access_priority = 0
inactivity_timeout = 48
drop_link_on_inactivity = yes
response_timeout = 2
acknowledgement_timeout = 1
force_disconnect_timeout = 120
link_trace = no
trace_entry_size = short
logical_link_type = token_ring
maximum_i_field = system_defined
maximum_i_field_size = 30729
physical_link_type = token_ring
```

* PROFILE TYPE BEING PRINTED: "TOKEN RING PHYSICAL"

VSE62_TOKENRINGPHYSICAL:

```
type = TOKENRINGPHYSICAL
profile_name = VSE62
device_name = tok0
local_link_name =
local_sap_address = 04
physical_link_type = token_ring
maximum_number_of_logical_links = 32
```

B.2 Definition Examples for CICS/6000 Region

```
Region name="CICS6K"
ResourceDescription="Region Definition"
AmendCounter=2
Modifiable=no
StartType=cold
Groups=
StartupProgList=""
ShutdownProgList1=""
ShutdownProgList2=""
DefaultUserId="CICSUSER"
FileRSLCheck=external
TransientDataRSLCheck=external
TemporaryStorageRSLCheck=external
JournalRSLCheck=external
ProgramRSLCheck=external
TransactionRSLCheck=external
ESMLoad=no
ESModule=""
RuntimeProtection=none
LogProtection=none
LogicalTDQProtection=none
PhysicalTDQProtection=none
NonRecTDQProtection=none
RecTSQProtection=none
NonRecTSQProtection=none
LocalQProtectProtection=none
LocalQProtection=none
ReleaseNum="0110"
LocalSysId="ISC0"
LocalNetworkName="DEIBMIPF"
CWASize=512
MinServer=4
MaxServer=10
ServerIdleLimit=300
SafetyLevel=normal
TSLISTENINGThreads=0
DateForm=ddmmyy
ClassTableSize=3,10,17,1,5,19,7,3,2,17,1,5
MaxRegionPool=2097152
MaxTaskPrivatePool=1048576
MaxTSHPool=1048576
RegionPoolThreshold=90
TaskShPoolThreshold=90
TaskSHNumBuckets=512
LoadDataNumBuckets=512
SysDump=no
PCDump=yes
ABDump=yes
DumpName="dumps"
CoreDumpName="dir1"
Trace=all
TraceModules=0
ExternalTrace=no
TraceFileA="trace.a"
TraceFileB="trace.b"
TraceFileSize=163840
```

```

SysTraceBufferSize=16384
UserTraceDirectory="/tmp"
PublicUserTraceFile="cicspub1"
IntrospectInterval=10
IntrospectLevel=minimal
CATSDelayMinutes=10
CUBSDelayMinutes=5
CARPDelayHours=8
ProtectPurgeDelayPeriod=8
PurgeDelayPeriod=8
StatsRecord=yes
StatFile="statsfile"
SufficesSupported=yes
CheckpointInterval=1000
NonRecTSQServer="/./cics/sfs/%H"
NonRecTSQFile="%Rcicsnrectsqfil"
NonRecTSQIndex="cicsnrectsqidx"
NonRecTSQVol="sfs_%S"
NonRecTSQPrePages=5
NonRecTSQMaxRecs=1000000
LogicalTDQServer="/./cics/sfs/%H"
LogicalTDQFile="%Rcicstdqlgfile"
LogicalTDQIndex="cicstdqlgidx"
LogicalTDQVol="sfs_%S"
LogicalTDQPrePages=5
LogicalTDQMaxRecs=1000000
PhysicalTDQServer="/./cics/sfs/%H"
PhysicalTDQFile="%Rcicstdqphfile"
PhysicalTDQIndex="cicstdqphidx"
PhysicalTDQVol="sfs_%S"
PhysicalTDQPrePages=5
PhysicalTDQMaxRecs=1000000
NonRecTDQServer="/./cics/sfs/%H"
NonRecTDQFile="%Rcicstdqnofile"
NonRecTDQIndex="cicstdqnoidx"
NonRecTDQVol="sfs_%S"
NonRecTDQPrePages=5
NonRecTDQMaxRecs=1000000
LocalQProtectServer="/./cics/sfs/%H"
LocalQProtectFile="%Rcicsplqfile"
LocalQProtectIndex="cicsplqidx"
LocalQProtectVol="sfs_%S"
LocalQProtectPrePages=5
LocalQProtectMaxRecs=1000000
LocalQServer="/./cics/sfs/%H"
LocalQFile="%Rcicsnlqfile"
LocalQIndex="cicsnlqidx"
LocalQVol="sfs_%S"
LocalQPrePages=5
LocalQMaxRecs=1000000
TSQAgeLimit=20

```

B.3 Definition Examples for CICS/6000 Communication

Communication Identifier="SNA1"

```
SNA1:
GroupName="sna"
ActivateOnStartup=yes
ResourceDescription="LU6.2 to CICS/VSE"
AmendCounter=0
Permanent=no
AllocateTimeout=0
TSLKeyMask=none
RSLKeyMask=none
RemoteSysType=SNA
RemoteLUName="CICSSA22"
GatewayName="GWY1"
RemoteNetworkName="DEIBMIPF"
SNAConnectName="LU62VSE"
RemoteCodePageTR="IBM-037"
DCECell="/./"
GatewayPrincipal=""
RemoteSysSecurity=local
LinkUserId=""
RemoteSysEncrypt=none
OutboundUserIds=sent
InService=yes
```

Communication Identifier="GWY1"

```
GWY1:
GroupName="sna"
ActivateOnStartup=yes
ResourceDescription="Gateway Definition"
AmendCounter=0
Permanent=no
AllocateTimeout=0
TSLKeyMask=none
RSLKeyMask=none
RemoteSysType=GWY
RemoteLUName="GWY1"
GatewayName=""
RemoteNetworkName=""
SNAConnectName=""
RemoteCodePageTR="IBM-850"
DCECell="/./cics/ppc"
GatewayPrincipal=""
RemoteSysSecurity=local
LinkUserId=""
RemoteSysEncrypt=none
OutboundUserIds=sent
InService=yes
```

B.4 Application Definition Examples

B.4.1 Transaction Definition

Transaction Identifier="TECH"

TECH:
GroupName="tech"
ActivateOnStartup=yes
ResourceDescription="Demo transaction"
AmendCounter=0
Permanent=no
EnableStatus=enabled
RemoteSysId=""
RemoteName=""
RSLKey=public
TSLKey=1
RSLCheck=none
TSLCheck=internal
TransDump=no
ProgName="TECCPROG"
TPNSNAProfile="TDEFAULT"
TemplateDefined=no
IsBackEndDTP=no
EnabledAtShutdown=no
TWASize=0
Priority=0
Purgeability=purgeable
DeadLockTimeout=0
InDoubt=wait_backout
UCTranFlag=yes
SNAModeName=""
LocalQ=no
Timeout=0
InvocationMode=no_facility|output_terminal|in_out_terminal|comms_link|
user_start|ATI_start|internal_ATI_start|triggered_start|at_normal_running

B.4.2 Program Definitions

B.4.2.1 Mapset Definition

Program Identifier="NEW1MS"

```
NEW1MS:
GroupName="tech"
ActivateOnStartup=yes
ResourceDescription="Demo mapset"
AmendCounter=0
Permanent=no
EnableStatus=enabled
RemoteSysId=""
RemoteName=""
RSLKey=public
PathName="/var/cics_regions/CICS6K/maps/En_US/NEW1MS.map"
ProgType=map
TemplateDefined=no
```

B.4.2.2 Front-end Program Definition

Program Identifier="TECCPROG"

```
TECCPROG:
GroupName="tech"
ActivateOnStartup=yes
ResourceDescription="Front end program"
AmendCounter=0
Permanent=no
EnableStatus=enabled
RemoteSysId=""
RemoteName=""
RSLKey=public
PathName="teccprog"
ProgType=program
TemplateDefined=no
```

B.4.2.3 Backend Program Definition

Program Identifier="CICSVSE"

```
CICSVSE:
GroupName="tech"
ActivateOnStartup=yes
ResourceDescription="Back end program"
AmendCounter=0
Permanent=no
EnableStatus=enabled
RemoteSysId="SNA1"
RemoteName=""
RSLKey=private
PathName=""
ProgType=program
TemplateDefined=no
```

Appendix C. Sample Application Job Streams and Definitions

The source code and jobstreams listed below can be obtained as package 'VSECSDEM' from disk IBMVSE managed by VMTOOLS at BOEVM3. To get the package use the VM/CMS 'TOOLS GET' command and specify the package name, disk and tools machine mentioned above.

C.1 Front-end Program Source (File Name TECCPROG.CCP)

```
IDENTIFICATION DIVISION.
PROGRAM-ID. TECCPROG.

ENVIRONMENT DIVISION.

DATA DIVISION.
WORKING-STORAGE SECTION.
* PUT IN FOR WRITEQ VVVV
01 WRITEQ-WORDS.
03 OPERATING-SYSTEM          PIC X(9) VALUE 'OS/2 2.1:'.
03 PROGRAM-HEADER           PIC X(9) VALUE 'PROGRAM: '.
03 PROGRAM-NAME             PIC X(9) VALUE 'TECCPROG '.
03 COMMENT-FIELD            PIC X(40).

01 WS-VALUES.
05 MESSAGE-TO-SEND          PIC X(20).
05 TEMP-LEN-20              PIC X(20).
05 RESP-FIELD               PIC S9(8) BINARY.
05 PROGRAM-STATUS          PIC S9(8) BINARY.
05 CREATE-REQUEST           PIC 9    VALUE 1.
05 READ-REQUEST             PIC 9    VALUE 2.
05 UPDATE-REQUEST          PIC 9    VALUE 3.
05 DELETE-REQUEST          PIC 9    VALUE 4.
05 RM-NONE                  PIC 9    VALUE 0.
05 RM-NOMORE                PIC 9    VALUE 1.
* Definitions of the various Resource Managers (Backends..
* note RM-MAX = max no. of Resource managers related to this.
* note RM-STATES-TABLE to store states related to this.
05 RM-AIX                   PIC 9    VALUE 1.
05 RM-AIX-PROG              PIC X(8) VALUE 'CICSAIX'.
05 RM-OS2                   PIC 9    VALUE 2.
05 RM-OS2-PROG              PIC X(8) VALUE 'VSAMSERV'.
05 RM-OS2DB2                PIC 9    VALUE 3.
05 RM-OS2DB2-PROG          PIC X(8) VALUE 'CICSODB'.
05 RM-ESA                   PIC 9    VALUE 4.
05 RM-ESA-PROG              PIC X(8) VALUE 'CICSESA'.
05 RM-400                   PIC 9    VALUE 5.
05 RM-400-PROG              PIC X(8) VALUE 'CICS400'.
05 RM-HPU                   PIC 9    VALUE 6.
05 RM-HPU-PROG              PIC X(8) VALUE 'CICSHPU'.
05 RM-VSE                   PIC 9    VALUE 7.
05 RM-VSE-PROG              PIC X(8) VALUE 'CICSVSE'.
* Now variable to keep track of enabled / disabled...
05 RM-ENABLED               PIC 9    VALUE 1.
```

```

05 RM-DISABLED          PIC 9    VALUE 2.
05 RM-MAX               PIC 9    VALUE 8.
05 RESOURCE-MANAGER    PIC S9(8) BINARY.
05 NUMBER-OF-RESOURCE-MANAGERS
                        PIC S9(8) BINARY.

05 RM-STATES-TABLE.
   10 RM-STATES OCCURS 8 TIMES PIC 9.

01 ACCOUNT-ADDED-MESSAGE.
   05 FILLER             PIC X(36)
      VALUE 'Account created with account number '.
   05 ACCOUNT-NUMBER-ADDED PIC 9(5).

01 NO-RESOURCE-MANAGER-MESSAGE PIC X(36)
   VALUE 'TECH: No resource Managers present!!'.

01 WS-COMMAREA.
   05 COMMAREA-REQUEST PIC 9.
   05 COMMAREA-RETURN-VALUE PIC 9.
   05 COMMAREA-KEY-SIZE PIC S9(4) BINARY.
   05 COMMAREA-ACCOUNT-RECORD.
       10 COMMAREA-ACCOUNT PIC X(5).
       10 COMMAREA-SURNAME PIC X(15).
       10 COMMAREA-FORENAME PIC X(10).
       10 COMMAREA-ADDR1 PIC X(15).
       10 COMMAREA-ADDR2 PIC X(15).
       10 COMMAREA-ADDR3 PIC X(15).
       10 COMMAREA-BALANCE PIC S9(4) BINARY.
   05 COMMAREA-RESPONSES PIC X(50).

COPY NEWIMS.
COPY DFHAID.

LINKAGE SECTION.
01 DFHCOMMAREA PIC X.

PROCEDURE DIVISION.
* DEBUGGING CODE
  MOVE 'STARTING' TO COMMENT-FIELD
  PERFORM WRITEQ-TS

  MOVE LOW-VALUES TO T000I T001I T002I T003I T004I T005I
  T006I T007I.
*****
* Initialise resource manager, this is a logic error.. *
* Check resource manager has been modified to correct it. *
*****
  MOVE RM-AIX TO RESOURCE-MANAGER.
*****
* Check which backends are actually available..... *
*****
  PERFORM CHECK-RESOURCE-MANAGERS.
  IF NUMBER-OF-RESOURCE-MANAGERS = 0
    EXEC CICS SEND CONTROL ERASE FREEKB END-EXEC
    EXEC CICS SEND FROM(NO-RESOURCE-MANAGER-MESSAGE)
      LENGTH(LENGTH OF NO-RESOURCE-MANAGER-MESSAGE)
    END-EXEC
    EXEC CICS RETURN END-EXEC.

```


EXEC CICS SEND CONTROL ERASE END-EXEC.

* Main Program loop starts here... *

MAIN-PROCESS-LOOP.

PERFORM SHOW-RESOURCE-MANAGER.

MOVE TEMP-LEN-20 TO RMUSED10.

MOVE ' ' TO OPTION0.

EXEC CICS SEND MAP('TO01')

MAPSET('NEWIMS')

FREEKB ERASEAUP END-EXEC.

EXEC CICS RECEIVE MAP('TO01')

MAPSET('NEWIMS')

NOHANDLE

END-EXEC.

MOVE ' ' TO MESS10.

EVALUATE EIBAID

WHEN DFHPF3 PERFORM EXIT-TRANSACTION

WHEN DFHPF12 PERFORM EXIT-TRANSACTION

WHEN DFHCLEAR CONTINUE

WHEN DFHPF6

IF NUMBER-OF-RESOURCE-MANAGERS = 1

MOVE 'No other resource mangers' TO MESS10

ELSE

PERFORM NEXT-RESOURCE-MANAGER

END-IF

WHEN DFHENTER

IF OPTIONL = 0

MOVE 'You must enter an option' TO MESS10

ELSE

EVALUATE OPTIONI

WHEN '1'

PERFORM PROCESS-SHOW THRU

PROCESS-SHOW-EXIT

EXEC CICS SEND CONTROL ERASE

FREEKB END-EXEC

WHEN '2'

PERFORM PROCESS-ADD THRU PROCESS-ADD-EXIT

EXEC CICS SEND CONTROL ERASE

FREEKB END-EXEC

WHEN '3'

PERFORM PROCESS-UPDATE THRU

PROCESS-UPDATE-EXIT

EXEC CICS SEND CONTROL ERASE

FREEKB END-EXEC

WHEN '4'

PERFORM PROCESS-DELETE THRU

PROCESS-DELETE-EXIT

EXEC CICS SEND CONTROL ERASE

FREEKB END-EXEC

WHEN OTHER

MOVE 'Invalid option chosen' to MESS10

END-EVALUATE

END-IF

WHEN OTHER

MOVE 'Invalid function key' TO MESS10

END-EVALUATE.

GO TO MAIN-PROCESS-LOOP.

```

MAIN-PROCESS-LOOP-EXIT.
  EXEC CICS RETURN END-EXEC.
*****
* Main Program loop ENDS here... *
*****

*****
* Process-Update Starts here... *
*****

PROCESS-UPDATE.
  EXEC CICS SEND CONTROL ERASE END-EXEC.
PROCESS-UPDATE-LOOP.
  PERFORM SHOW-RESOURCE-MANAGER.
  MOVE TEMP-LEN-20 TO RMUSED70.
  MOVE LOW-VALUES TO ACC70.
  EXEC CICS SEND MAP('T007') MAPSET('NEW1MS')
    ERASEAUP FREEKB END-EXEC.
  EXEC CICS RECEIVE MAP('T007') MAPSET('NEW1MS')
    NOHANDLE END-EXEC.
  MOVE ' ' TO MESS70.
  EVALUATE EIBAID
    WHEN DFHCLEAR          CONTINUE
    WHEN DFHPF3            GO TO PROCESS-UPDATE-EXIT
    WHEN DFHPF12           PERFORM EXIT-TRANSACTION
    WHEN DFHENTER
      IF ACC7L = 0
        MOVE 'You must enter an account number for update'
          TO MESS70
      ELSE
        MOVE ACC7I TO COMMAREA-ACCOUNT
        MOVE 1 TO COMMAREA-RETURN-VALUE
        MOVE READ-REQUEST TO COMMAREA-REQUEST
        PERFORM LINK-BACKEND
        IF COMMAREA-RETURN-VALUE NOT = 0
          MOVE COMMAREA-RESPONSES TO MESS70
        ELSE
          PERFORM UPDATE-ACCOUNT THRU
            UPDATE-ACCOUNT-EXIT
          EXEC CICS SEND CONTROL ERASE END-EXEC
        END-IF
      END-IF
    END-EVALUATE.
  GO TO PROCESS-UPDATE-LOOP.
PROCESS-UPDATE-EXIT.
  EXIT.
*****
* Process-Update ends here... *
*****

NEXT-RESOURCE-MANAGER.
  ADD 1 TO RESOURCE-MANAGER.
  IF RESOURCE-MANAGER > RM-MAX
    MOVE 1 TO RESOURCE-MANAGER.
  PERFORM UNTIL RM-STATES(RESOURCE-MANAGER) = RM-ENABLED
    ADD 1 TO RESOURCE-MANAGER
    IF RESOURCE-MANAGER > RM-MAX
      MOVE 1 TO RESOURCE-MANAGER
  END-IF

```

END-PERFORM.

SHOW-RESOURCE-MANAGER.

```
*****
* Start of SHOW-RESOURCE-MANAGER *
* Puts appropriate string into TEMP-LEN-20 so the backend *
* in use shows on the maps. *
*****
```

EVALUATE RESOURCE-MANAGER

```
      WHEN RM-AIX      MOVE 'CICS/6000 VSAM      '
                          TO TEMP-LEN-20
      WHEN RM-OS2      MOVE 'CICS/OS2 VSAM      '
                          TO TEMP-LEN-20
      WHEN RM-OS2DB2   MOVE 'CICS/OS2 DB/2      '
                          TO TEMP-LEN-20
      WHEN RM-ESA      MOVE 'CICS/ESA VSAM      '
                          TO TEMP-LEN-20
      WHEN RM-400      MOVE 'CICS/400 VSAM      '
                          TO TEMP-LEN-20
      WHEN RM-HPU      MOVE 'CICS ON HP VSAM    '
                          TO TEMP-LEN-20
      WHEN RM-VSE      MOVE 'CICS/VSE VSAM      '
                          TO TEMP-LEN-20
```

END-EVALUATE.

```
*****
* End of SHOW-RESOURCE-MANAGER *
*****
```

CHECK-RESOURCE-MANAGERS.

```
*****
* Start of check which backends are actually available.. *
* N.B. This check just determines if they're defined! NOT *
* necessarily guaranteed to be available..... *
* Done by issuing INQUIRE's against each possible backend *
*****
```

MOVE 0 TO NUMBER-OF-RESOURCE-MANAGERS.

EXEC CICS INQUIRE

```
      PROGRAM(RM-AIX-PROG)
      STATUS(PROGRAM-STATUS)
      RESP(RESP-FIELD)
      END-EXEC.
```

```
IF ((PROGRAM-STATUS = DFHVALUE(DISABLED)) OR
    (RESP-FIELD NOT = DFHRESP(NORMAL)))
  MOVE RM-DISABLED TO RM-STATES(RM-AIX)
```

ELSE

```
  MOVE RM-ENABLED TO RM-STATES(RM-AIX)
  MOVE RM-AIX TO RESOURCE-MANAGER
  ADD 1 to NUMBER-OF-RESOURCE-MANAGERS.
```

EXEC CICS INQUIRE

```
      PROGRAM(RM-OS2DB2-PROG)
      STATUS(PROGRAM-STATUS)
      RESP(RESP-FIELD)
      END-EXEC.
```

```
IF ((PROGRAM-STATUS = DFHVALUE(DISABLED)) OR
    (RESP-FIELD NOT = DFHRESP(NORMAL)))
  MOVE RM-DISABLED TO RM-STATES(RM-OS2DB2)
```

```

ELSE
    MOVE RM-ENABLED TO RM-STATES(RM-OS2DB2)
    MOVE RM-OS2DB2 TO RESOURCE-MANAGER
    ADD 1 to NUMBER-OF-RESOURCE-MANAGERS.

EXEC CICS INQUIRE
    PROGRAM(RM-ESA-PROG)
    STATUS(PROGRAM-STATUS)
    RESP(RESP-FIELD)
    END-EXEC.
IF ((PROGRAM-STATUS = DFHVALUE(DISABLED)) OR
    (RESP-FIELD NOT = DFHRESP(NORMAL)))
    MOVE RM-DISABLED TO RM-STATES(RM-ESA)
ELSE
    MOVE RM-ENABLED TO RM-STATES(RM-ESA)
    MOVE RM-ESA TO RESOURCE-MANAGER
    ADD 1 to NUMBER-OF-RESOURCE-MANAGERS.

EXEC CICS INQUIRE
    PROGRAM(RM-400-PROG)
    STATUS(PROGRAM-STATUS)
    RESP(RESP-FIELD)
    END-EXEC.
IF ((PROGRAM-STATUS = DFHVALUE(DISABLED)) OR
    (RESP-FIELD NOT = DFHRESP(NORMAL)))
    MOVE RM-DISABLED TO RM-STATES(RM-400)
ELSE
    MOVE RM-ENABLED TO RM-STATES(RM-400)
    MOVE RM-400 TO RESOURCE-MANAGER
    ADD 1 to NUMBER-OF-RESOURCE-MANAGERS.

EXEC CICS INQUIRE
    PROGRAM(RM-HPU-PROG)
    STATUS(PROGRAM-STATUS)
    RESP(RESP-FIELD)
    END-EXEC.
IF ((PROGRAM-STATUS = DFHVALUE(DISABLED)) OR
    (RESP-FIELD NOT = DFHRESP(NORMAL)))
    MOVE RM-DISABLED TO RM-STATES(RM-HPU)
ELSE
    MOVE RM-ENABLED TO RM-STATES(RM-HPU)
    MOVE RM-HPU TO RESOURCE-MANAGER
    ADD 1 to NUMBER-OF-RESOURCE-MANAGERS.

EXEC CICS INQUIRE
    PROGRAM(RM-VSE-PROG)
    STATUS(PROGRAM-STATUS)
    RESP(RESP-FIELD)
    END-EXEC.
IF ((PROGRAM-STATUS = DFHVALUE(DISABLED)) OR
    (RESP-FIELD NOT = DFHRESP(NORMAL)))
    MOVE RM-DISABLED TO RM-STATES(RM-VSE)
ELSE
    MOVE RM-ENABLED TO RM-STATES(RM-VSE)
    MOVE RM-VSE TO RESOURCE-MANAGER
    ADD 1 to NUMBER-OF-RESOURCE-MANAGERS.

EXEC CICS INQUIRE
    PROGRAM(RM-OS2-PROG)

```

```

                STATUS(PROGRAM-STATUS)
                RESP(RESF-FIELD)
                END-EXEC.
IF ((PROGRAM-STATUS = DFHVALUE(DISABLED)) OR
    (RESF-FIELD NOT = DFHRESP(NORMAL)))
    MOVE RM-DISABLED TO RM-STATES(RM-OS2)
ELSE
    MOVE RM-ENABLED TO RM-STATES(RM-OS2)
    MOVE RM-OS2 TO RESOURCE-MANAGER
    ADD 1 to NUMBER-OF-RESOURCE-MANAGERS.

```

```

*****
* End of check which backends are actually available..
*****

```

```
LINK-BACKEND.
```

```

*****
* Start of Link to Backend...
*****

```

```

* DEBUGGING CODE
  MOVE 'ABOUT TO LINK TO BACKEND.' TO COMMENT-FIELD
  PERFORM WRITEQ-TS

```

```

EVALUATE RESOURCE-MANAGER
  WHEN RM-AIX
    EXEC CICS LINK PROGRAM(RM-AIX-PROG)
      COMMAREA(WS-COMMAREA)
      LENGTH(LENGTH OF WS-COMMAREA)
      NOHANDLE
      END-EXEC
  WHEN RM-OS2
    EXEC CICS LINK PROGRAM(RM-OS2-PROG)
      COMMAREA(WS-COMMAREA)
      LENGTH(LENGTH OF WS-COMMAREA)
      NOHANDLE
      END-EXEC
  WHEN RM-OS2DB2
    EXEC CICS LINK PROGRAM(RM-OS2DB2-PROG)
      COMMAREA(WS-COMMAREA)
      LENGTH(LENGTH OF WS-COMMAREA)
      NOHANDLE
      END-EXEC
  WHEN RM-ESA
    EXEC CICS LINK PROGRAM(RM-ESA-PROG)
      COMMAREA(WS-COMMAREA)
      LENGTH(LENGTH OF WS-COMMAREA)
      NOHANDLE
      END-EXEC
  WHEN RM-400
    EXEC CICS LINK PROGRAM(RM-400-PROG)
      COMMAREA(WS-COMMAREA)
      LENGTH(LENGTH OF WS-COMMAREA)
      NOHANDLE
      END-EXEC
  WHEN RM-HPU
    EXEC CICS LINK PROGRAM(RM-HPU-PROG)
      COMMAREA(WS-COMMAREA)
      LENGTH(LENGTH OF WS-COMMAREA)
      NOHANDLE

```

```

                END-EXEC
            WHEN RM-VSE
                EXEC CICS LINK PROGRAM(RM-VSE-PROG)
                    COMMAREA(WS-COMMAREA)
                    LENGTH(LENGTH OF WS-COMMAREA)
                    NOHANDLE
                END-EXEC
        END-EVALUATE.
        IF EIBRESP NOT = 0
            MOVE 'BACKEND NOT AVAILABLE' TO COMMAREA-RESPONSES
        END-IF
    * DEBUGGING CODE
        MOVE 'RETURNED FROM BACKEND' TO COMMENT-FIELD
        PERFORM WRITEQ-TS

        EXEC CICS SYNCPOINT END-EXEC.
*****
* End of Link to Backend... *
*****

        EXIT-TRANSACTION.
*****
* End of this transaction, put out message and die...
*****
    * DEBUGGING CODE
        MOVE 'EXITING.....' TO COMMENT-FIELD
        PERFORM WRITEQ-TS

        MOVE 'TECH: Session Over' to MESSAGE-TO-SEND.
        EXEC CICS SEND CONTROL ERASE FREEKB END-EXEC.
        EXEC CICS SEND FROM(MESSAGE-TO-SEND)
            LENGTH(LENGTH OF MESSAGE-TO-SEND) END-EXEC.
        EXEC CICS RETURN END-EXEC.
*****
* End of End of this transaction, put out message and *
* DIE... *
*****

        SHOW-ACCOUNT.
*****
* SHOW-ACCOUNT start... *
*****

        MOVE COMMAREA-SURNAME TO SUR30.
        MOVE COMMAREA-FORENAME TO FOR30.
        MOVE COMMAREA-ADDR1 TO ADD130.
        MOVE COMMAREA-ADDR2 TO ADD230.
        MOVE COMMAREA-ADDR3 TO ADD330.
        MOVE COMMAREA-ACCOUNT TO ACC30.
        MOVE COMMAREA-BALANCE TO BAL30.
        SHOW-ACCOUNT-LOOP.
        EXEC CICS SEND CONTROL ERASE END-EXEC.
        PERFORM SHOW-RESOURCE-MANAGER.
        MOVE TEMP-LEN-20 TO RMUSED30.
        EXEC CICS SEND MAP('T003') MAPSET('NEW1MS') FREEKB END-EXEC.
        EXEC CICS RECEIVE MAP('T003') MAPSET('NEW1MS')
            NOHANDLE END-EXEC.

        EVALUATE EIBAID
            WHEN DFHPF12 PERFORM EXIT-TRANSACTION
            WHEN DFHENTER GO TO SHOW-ACCOUNT-EXIT

```

```

        WHEN DFHPF3      GO TO SHOW-ACCOUNT-EXIT
        WHEN DFHCLEAR
            MOVE COMMAREA-SURNAME TO SUR30
            MOVE COMMAREA-FORENAME TO FOR30
            MOVE COMMAREA-ADDR1    TO ADD130
            MOVE COMMAREA-ADDR2    TO ADD230
            MOVE COMMAREA-ADDR3    TO ADD330
            MOVE COMMAREA-ACCOUNT  TO ACC30
            MOVE COMMAREA-BALANCE  TO BAL30
        END-EVALUATE.
        GO TO SHOW-ACCOUNT-LOOP.
    SHOW-ACCOUNT-EXIT.
    EXIT.
*****
* SHOW-ACCOUNT end...
*****

PROCESS-SHOW.
*****
* PROCESS-SHOW start...
*****
        EXEC CICS SEND CONTROL ERASE END-EXEC.
    PROCESS-SHOW-LOOP.
        PERFORM SHOW-RESOURCE-MANAGER.
        MOVE TEMP-LEN-20 TO RMUSED20.
        MOVE LOW-VALUES TO SUR20.
        MOVE LOW-VALUES TO ACC20.
        EXEC CICS SEND MAP('T002') MAPSET('NEW1MS')
            FREEKB ERASEAUP END-EXEC.
        EXEC CICS RECEIVE MAP('T002') MAPSET('NEW1MS')
            NOHANDLE END-EXEC.
        MOVE ' ' TO MESS20.
        EVALUATE EIBAID
            WHEN DFHENTER
                IF ((SUR2L NOT = 0) AND (ACC2L NOT = 0))
                    MOVE 'Not valid to enter data in BOTH surname and account'
                        TO MESS20
                ELSE
                    IF ((SUR2L = 0) AND (ACC2L = 0))
                        MOVE 'You must enter either a Surname OR an Account number'
                            TO MESS20
                ELSE
                    MOVE SUR2I TO COMMAREA-SURNAME
                    MOVE ACC2I TO COMMAREA-ACCOUNT
                    MOVE SUR2L TO COMMAREA-KEY-SIZE
                    MOVE 1 TO COMMAREA-RETURN-VALUE
                    MOVE READ-REQUEST TO COMMAREA-REQUEST
                    PERFORM LINK-BACKEND
                    IF COMMAREA-RETURN-VALUE NOT = 0
                        MOVE COMMAREA-RESPONSES TO MESS20
                    ELSE
                        PERFORM SHOW-ACCOUNT THRU
                            SHOW-ACCOUNT-EXIT
                        EXEC CICS SEND CONTROL ERASE END-EXEC
                    END-IF
                END-IF
            END-IF
        WHEN DFHPF3      GO TO PROCESS-SHOW-EXIT

```

```

        WHEN DFHPF12 PERFORM EXIT-TRANSACTION
        WHEN DFHCLEAR CONTINUE
        WHEN OTHER
            MOVE 'Invalid function key' TO MESS20
        END-EVALUATE.
        GO TO PROCESS-SHOW-LOOP.
    PROCESS-SHOW-EXIT.
    EXIT.
*****
* PROCESS-SHOW end... *
*****

    PROCESS-ADD.
*****
* PROCESS-ADD start... *
*****
        EXEC CICS SEND CONTROL ERASE END-EXEC.
        MOVE ' ' TO SUR40.
        MOVE ' ' TO FOR40.
        MOVE ' ' TO ADD140.
        MOVE ' ' TO ADD240.
        MOVE ' ' TO ADD340.
        MOVE ' ' TO BAL40.
    PROCESS-ADD-LOOP.
        PERFORM SHOW-RESOURCE-MANAGER.
        MOVE TEMP-LEN-20 TO RMUSED40.
        EXEC CICS SEND MAP('T004') MAPSET('NEW1MS') FREEKB END-EXEC.
        EXEC CICS RECEIVE MAP('T004') MAPSET('NEW1MS')
            NOHANDLE END-EXEC.
        MOVE ' ' TO MESS40.
        EVALUATE EIBAID
        WHEN DFHENTER
            IF ((SUR4L = 0) OR (SUR4I = LOW-VALUE) OR (SUR4I = ' '))
                MOVE 'You must enter a Surname' TO MESS40
            ELSE
                MOVE SUR4I TO COMMAREA-SURNAME
                MOVE FOR4I TO COMMAREA-FORENAME
                MOVE ADD14I TO COMMAREA-ADDR1
                MOVE ADD24I TO COMMAREA-ADDR2
                MOVE ADD34I TO COMMAREA-ADDR3
                MOVE BAL4I TO COMMAREA-BALANCE
                MOVE CREATE-REQUEST TO COMMAREA-REQUEST
                MOVE 1 TO COMMAREA-RETURN-VALUE
                PERFORM LINK-BACKEND
                IF COMMAREA-RETURN-VALUE NOT = 0
                    MOVE COMMAREA-RESPONSES TO MESS40
                ELSE
                    MOVE COMMAREA-ACCOUNT TO
                        ACCOUNT-NUMBER-ADDED
                    MOVE ACCOUNT-ADDED-MESSAGE
                        TO MESS40
            END-IF
        END-IF
        WHEN DFHPF3 GO TO PROCESS-ADD-EXIT
        WHEN DFHPF12 PERFORM EXIT-TRANSACTION
        WHEN DFHCLEAR
            MOVE ' ' to SUR40

```



```

        MOVE ' ' to FOR40
        MOVE ' ' to ADD140
        MOVE ' ' TO ADD240
        MOVE ' ' TO ADD340
        MOVE ' ' TO BAL40
    WHEN OTHER
        MOVE 'Invalid function key' TO MESS40
    END-EVALUATE.
    GO TO PROCESS-ADD-LOOP.
PROCESS-ADD-EXIT.
EXIT.
*****
* PROCESS-ADD end...
*****

PROCESS-DELETE.
*****
* PROCESS-DELETE start...
*****
    EXEC CICS SEND CONTROL ERASE END-EXEC.
PROCESS-DELETE-LOOP.
    PERFORM SHOW-RESOURCE-MANAGER.
    MOVE TEMP-LEN-20 to RMUSED50.
    MOVE LOW-VALUES to ACC50.
    EXEC CICS SEND MAP('T005') MAPSET('NEW1MS')
        ERASEAUP FREEKB END-EXEC.
    EXEC CICS RECEIVE MAP('T005') MAPSET('NEW1MS')
        NOHANDLE END-EXEC.
    MOVE ' ' TO MESS50.
    EVALUATE EIBRID
        WHEN DFHCLEAR CONTINUE
        WHEN DFHPF3 GO TO PROCESS-DELETE-EXIT
        WHEN DFHPF12 PERFORM EXIT-TRANSACTION
        WHEN DFHENTER
            IF ACC5L = 0
                MOVE 'You must enter an account number for deletion'
                    TO MESS50
            ELSE
                MOVE ACC5I TO COMMAREA-ACCOUNT
                MOVE DELETE-REQUEST TO COMMAREA-REQUEST
                MOVE 1 TO COMMAREA-RETURN-VALUE
                PERFORM LINK-BACKEND
                IF COMMAREA-RETURN-VALUE NOT = '0'
                    MOVE COMMAREA-RESPONSES TO MESS50
                ELSE
                    MOVE 'Account deleted' to MESS50
                END-IF
            END-IF
        END-EVALUATE.
    GO TO PROCESS-DELETE-LOOP.
PROCESS-DELETE-EXIT.
EXIT.
*****
* PROCESS-DELETE end...
*****

UPDATE-ACCOUNT.
*****

```

```

* UPDATE ACCOUNT start...
*****
EXEC CICS SEND CONTROL ERASE END-EXEC.
MOVE COMMAREA-SURNAME TO SUR60.
MOVE COMMAREA-FORENAME TO FOR60.
MOVE COMMAREA-ADDR1 TO ADD160.
MOVE COMMAREA-ADDR2 TO ADD260.
MOVE COMMAREA-ADDR3 TO ADD360.
MOVE COMMAREA-ACCOUNT TO ACC60.
MOVE COMMAREA-BALANCE TO BAL60.
UPDATE-ACCOUNT-LOOP.
PERFORM SHOW-RESOURCE-MANAGER.
MOVE TEMP-LEN-20 TO RMUSED60.
EXEC CICS SEND MAP('T006') MAPSET('NEW1MS')
FREEKB END-EXEC.
EXEC CICS RECEIVE MAP('T006') MAPSET('NEW1MS')
NOHANDLE END-EXEC.
MOVE ' ' TO MESS60.
EVALUATE EIBRID
WHEN DFHCLEAR
MOVE 'Modifications undone' TO MESS60
MOVE COMMAREA-SURNAME TO SUR60
MOVE COMMAREA-FORENAME TO FOR60
MOVE COMMAREA-ADDR1 TO ADD160
MOVE COMMAREA-ADDR2 TO ADD260
MOVE COMMAREA-ADDR3 TO ADD360
MOVE COMMAREA-ACCOUNT TO ACC60
MOVE COMMAREA-BALANCE TO BAL60
WHEN DFHPF3
MOVE 'Updated Canceled' TO MESS70
GO TO UPDATE-ACCOUNT-EXIT
WHEN DFHPF12 PERFORM EXIT-TRANSACTION
WHEN DFHENTER
MOVE SUR6I TO COMMAREA-SURNAME
MOVE FOR6I TO COMMAREA-FORENAME
MOVE ADD16I TO COMMAREA-ADDR1
MOVE ADD26I TO COMMAREA-ADDR2
MOVE ADD36I TO COMMAREA-ADDR3
MOVE BAL6I TO COMMAREA-BALANCE
MOVE UPDATE-REQUEST TO COMMAREA-REQUEST
MOVE 1 TO COMMAREA-RETURN-VALUE
PERFORM LINK-BACKEND
IF COMMAREA-RETURN-VALUE NOT = 0
MOVE COMMAREA-RESPONSES TO MESS60
ELSE
MOVE 'Account Updated' TO MESS70
GO TO UPDATE-ACCOUNT-EXIT
END-IF
END-EVALUATE.
GO TO UPDATE-ACCOUNT-LOOP.
UPDATE-ACCOUNT-EXIT.
EXIT.
*****
* UPDATE ACCOUNT start...
*****

WRITEQ-TS SECTION.
*****
* This section added to write to the CEBR0000 TS Queue *

```

```

* at various points in the program so that you can      *
* prove the program is being used.                      *
*****
EXEC CICS WRITEQ TS
      QUEUE('CEBRO000')
      FROM(WRITEQ-WORDS)
      LENGTH(LENGTH OF WRITEQ-WORDS)
      END-EXEC.

WRITEQ-TS-EXIT.
EXIT.

```

C.2 Backend Program Source (Program CICSVSE)

```

*****
*
* AUTHOR\DATE\AREA JGSM, 11-04-1992, CICS OS/2 DEVELOPMENT.
* REVISED          JGSM 29-04-1993, CICS OS/2 DEVELOPMENT.
*
*
* VSAM File Layout
*
* KSDS base file = TECHBAS   Key - 00 pos=0 len=5
* Alt Index      = TECHALT   Key - 01 pos=5 len=15
*
* TECHBAS - record layout
*
* Field-name / Field-length
*
* Account-NO     5 - CHAR
* Surname        15 - CHAR
* First-Name     10 - CHAR
* Address        45 - CHAR
* Balance        4 - COMP
*
*
*****
IDENTIFICATION DIVISION.
PROGRAM-ID. CICSVSE.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-PC.
OBJECT-COMPUTER. IBM-PC.
*****
DATA DIVISION.

WORKING-STORAGE SECTION.
* PUT IN FOR WRITEQ VVVV
01 WRITEQ-WORDS.
   03 OPERATING-SYSTEM          PIC X(9) VALUE 'VSE/ESA: '.
   03 PROGRAM-HEADER            PIC X(9) VALUE 'PROGRAM: '.
   03 PROGRAM-NAME              PIC X(9) VALUE 'CICSVSE  '.
   03 COMMENT-FIELD             PIC X(40).

01 SERVER-IO-BUFFER.

   03 SERVER-STATE              PIC 9.

```

```

03 RETURN-VALUE          PIC 9.
03 SERVER-KEY-SIZE       PIC S9(4) BINARY.

03 SERVER-CUSTOMER-RECORD.
05 ACCOUNTO             PIC X(5).
05 SURNAMEO             PIC X(15).
05 FIRST-NAMEO          PIC X(10).
05 ADDRESSO             PIC X(45).
05 BALANCEO             PIC S9(4) BINARY.

03 RESPONSES            PIC X(50).

*****

01 MISCEL-VARS.
* LENGTH-VAR made a COMP so its binary, needed for READ
03 CUST-REC-LEN         PIC S9(4) VALUE 77.
03 LENGTH-VAR          PIC S9(4) BINARY VALUE 0.

03 KEY-SIZE             PIC S9(4) BINARY.
03 ACCOUNT-NUM         PIC 9(5) VALUE 0.

03 CHAR-BUFFER          PIC X(15).
03 SPLIT-BUFFER REDEFINES CHAR-BUFFER.
05 BUFFER-ELEMENT OCCURS 15 TIMES PIC X.
03 ELEMENT-PTR         PIC 99.
03 BUFFER-LEN          PIC 99.
03 CHAR-FOUND          PIC 9 VALUE 0.
*

03 BACKUP-CUSTOMER-RECORD.
05 ACCOUNT-BAK         PIC X(5).
05 SURNAME-BAK         PIC X(15).
05 FIRST-NAME-BAK     PIC X(10).
05 ADDRESS-BAK        PIC X(45).
05 BALANCE-BAK        PIC S9(4) BINARY.

03 RESPONSES-BAK      PIC X(50).

03 LEN-PARM            PIC S9(4) BINARY.

03 VALID-STATES.

05 SERVER-STATE-CREATE PIC 9 VALUE 1.
05 SERVER-STATE-READ  PIC 9 VALUE 2.
05 SERVER-STATE-UPDATE PIC 9 VALUE 3.
05 SERVER-STATE-DELETE PIC 9 VALUE 4.

COPY DFHAID.

LINKAGE SECTION.
01 DFHCOMMAREA          PIC X(131).
*****
PROCEDURE DIVISION.

MAINLINE SECTION.

MOVE DFHCOMMAREA TO SERVER-IO-BUFFER

```

```

* Convert trailing LOW-VALUES to SPACES in the customer record
  INSPECT SURNAMEO REPLACING ALL LOW-VALUES BY SPACES
  INSPECT FIRST-NAMEO REPLACING ALL LOW-VALUES BY SPACES
  INSPECT ADDRESSO REPLACING ALL LOW-VALUES BY SPACES

* DEBUGGING CODE
  MOVE 'STARTING' TO COMMENT-FIELD
  PERFORM WRITEQ-TS

  MOVE SERVER-KEY-SIZE TO KEY-SIZE

  EVALUATE SERVER-STATE
    WHEN SERVER-STATE-CREATE
      PERFORM CREATE-RECORD

    WHEN SERVER-STATE-READ
      PERFORM READ-RECORD

    WHEN SERVER-STATE-UPDATE
      PERFORM UPDATE-RECORD

    WHEN SERVER-STATE-DELETE
      PERFORM DELETE-RECORD

  END-EVALUATE
  MOVE SERVER-IO-BUFFER TO DFHCOMMAREA
* DEBUGGING CODE
  MOVE 'EXITING' TO COMMENT-FIELD
  PERFORM WRITEQ-TS
  EXEC CICS RETURN
    END-EXEC.

MAINLINE-EXIT.
  EXIT.
*****
CREATE-RECORD SECTION.

* check for duplicate record in file
* calculate key size (temp support for detecting dup records

  MOVE LOW-VALUES TO CHAR-BUFFER
  MOVE SURNAMEO TO CHAR-BUFFER
  MOVE 15 TO ELEMENT-PTR
  PERFORM STRING-LENGTH
  MOVE BUFFER-LEN TO KEY-SIZE

  MOVE CUST-REC-LEN TO LENGTH-VAR
  MOVE LOW-VALUES TO BACKUP-CUSTOMER-RECORD

  EXEC CICS READ
    FILE('TECHALT')
    INTO(BACKUP-CUSTOMER-RECORD)
    LENGTH(LENGTH-VAR)
    RIDFLD(SURNAMEO)
    KEYLENGTH(KEY-SIZE)
    GENERIC
    NOHANDLE
    END-EXEC

```

```

        IF SURNAMEO = SURNAME-BAK
            MOVE 14 TO EIBRESP
            PERFORM EVALUATE-RESP
            GO TO CREATE-RECORD-EXIT
        END-IF

* get max record number
    MOVE SERVER-CUSTOMER-RECORD TO BACKUP-CUSTOMER-RECORD

    MOVE HIGH-VALUES TO ACCOUNTO

    EXEC CICS STARTBR
        FILE('TECHBAS')
        RIDFLD(ACCOUNTO)
        GTEQ
        NOHANDLE
        END-EXEC

    EXEC CICS READPREV
        FILE('TECHBAS')
        INTO(SERVER-CUSTOMER-RECORD)
        LENGTH(LENGTH-VAR)
        RIDFLD(ACCOUNTO)
        NOHANDLE
        END-EXEC

    EXEC CICS ENDBR
        FILE('TECHBAS')
        NOHANDLE
        END-EXEC

* add 1 and enter new record
    MOVE ACCOUNTO TO ACCOUNT-NUM
    ADD 1 TO ACCOUNT-NUM
    MOVE BACKUP-CUSTOMER-RECORD TO SERVER-CUSTOMER-RECORD
    MOVE ACCOUNT-NUM TO ACCOUNTO

    MOVE 5 TO KEY-SIZE
    MOVE CUST-REC-LEN TO LENGTH-VAR

    EXEC CICS WRITE
        FILE('TECHBAS')
        FROM(SERVER-CUSTOMER-RECORD)
        LENGTH(LENGTH-VAR)
        RIDFLD(ACCOUNTO)
        KEYLENGTH(KEY-SIZE)
        NOHANDLE
        END-EXEC

    PERFORM EVALUATE-RESP.

CREATE-RECORD-EXIT.
    EXIT.
*****
READ-RECORD SECTION.

    MOVE CUST-REC-LEN TO LENGTH-VAR
    IF ACCOUNTO = LOW-VALUES

```

```

EXEC CICS READ
      FILE('TECHALT')
      INTO(SERVER-CUSTOMER-RECORD)
      LENGTH(LENGTH-VAR)
      RIDFLD(SURNAME0)
      KEYLENGTH(KEY-SIZE)
      GENERIC
      NOHANDLE
      END-EXEC

ELSE

      EXEC CICS READ
            FILE('TECHBAS')
            INTO(SERVER-CUSTOMER-RECORD)
            LENGTH(LENGTH-VAR)
            RIDFLD(ACCOUNT0)
            NOHANDLE
            END-EXEC

END-IF

PERFORM EVALUATE-RESP.

READ-RECORD-EXIT.
EXIT.
*****
UPDATE-RECORD SECTION.

MOVE SERVER-CUSTOMER-RECORD TO BACKUP-CUSTOMER-RECORD
MOVE CUST-REC-LEN TO LENGTH-VAR

EXEC CICS READ
      FILE('TECHBAS')
      INTO(SERVER-CUSTOMER-RECORD)
      LENGTH(LENGTH-VAR)
      RIDFLD(ACCOUNT0)
      UPDATE
      NOHANDLE
      END-EXEC

IF EIBRESP = 0

      MOVE BACKUP-CUSTOMER-RECORD TO SERVER-CUSTOMER-RECORD
      MOVE CUST-REC-LEN TO LENGTH-VAR

      EXEC CICS REWRITE
            FILE('TECHBAS')
            FROM(SERVER-CUSTOMER-RECORD)
            LENGTH(LENGTH-VAR)
            NOHANDLE
            END-EXEC

ELSE
PERFORM EVALUATE-RESP
END-IF
PERFORM EVALUATE-RESP.

UPDATE-RECORD-EXIT.
EXIT.

```

DELETE-RECORD SECTION.

```
MOVE CUST-REC-LEN TO LENGTH-VAR

EXEC CICS READ
    FILE('TECHBAS')
    INTO(SERVER-CUSTOMER-RECORD)
    LENGTH(LENGTH-VAR)
    RIDFLD(ACCOUNTO)
    UPDATE
    NOHANDLE
    END-EXEC

IF EIBRESP = 0
    EXEC CICS DELETE
        FILE('TECHBAS')
        NOHANDLE
        END-EXEC
ELSE
    PERFORM EVALUATE-RESP
END-IF

PERFORM EVALUATE-RESP.
```

DELETE-RECORD-EXIT.

EXIT.

EVALUATE-RESP SECTION.

```
EVALUATE EIBRESP

    WHEN DFHRESP(NORMAL)
        MOVE 0 TO RETURN-VALUE

    WHEN DFHRESP(DISABLED)
        MOVE 'File is out of action'
        TO RESPONSES

    WHEN 12
        MOVE 'File cannot be found'
        TO RESPONSES

    WHEN DFHRESP(DUPREC)
        MOVE 'Duplicate found, record NOT created'
        TO RESPONSES

    WHEN DFHRESP(NOTFND)
        MOVE 'Record cannot be found'
        TO RESPONSES

    WHEN DFHRESP(ILLOGIC)
        MOVE 'Illogic operation Hmm!'
        TO RESPONSES

    WHEN DFHRESP(INVREQ)
        MOVE 'Invalid request'
        TO RESPONSES
```



```

        WHEN DFHRESP(IOERR)
            MOVE 'IO error disk full or setup wrong !'
            TO RESPONSES

        WHEN DFHRESP(LENGERR)
            MOVE 'Invalid length Hmm!'
            TO RESPONSES

        WHEN DFHRESP(NOSPACE)
            MOVE 'Not enough disk space ! Hmm!'
            TO RESPONSES
    END-EVALUATE.

EVALUATE-RESP-EXIT.
    EXIT.
*****
STRING-LENGTH SECTION.

* used to calculate keysize
    MOVE ELEMENT-PTR TO BUFFER-LEN
    PERFORM COUNT-LOW-VALS UNTIL ELEMENT-PTR = 1
    GO TO STRING-LENGTH-EXIT.

COUNT-LOW-VALS.
    IF BUFFER-ELEMENT(ELEMENT-PTR) = LOW-VALUE OR SPACE
        SUBTRACT 1 FROM ELEMENT-PTR
    ELSE
        MOVE ELEMENT-PTR TO BUFFER-LEN
        MOVE 1 TO ELEMENT-PTR
    END-IF.

STRING-LENGTH-EXIT.
    EXIT.
*****
WRITEQ-TS SECTION.
* This section added to write to the CEBR0000 TS Queue at
* various points in the program so that you can prove the
* program is being used.
    EXEC CICS WRITEQ TS
        QUEUE('CEBR0000')
        FROM(WRITEQ-WORDS)
    END-EXEC.

WRITEQ-TS-EXIT.
    EXIT.

```

C.3 Compile Backend Program Job

```

* $$ JOB JNM=COMWSR1,DISP=D,CLASS=5,NTFY=YES
* $$ LST DISP=D,CLASS=Q,PRI=3
* $$ PUN DISP=I,PRI=9,CLASS=A
// JOB COMWSR1 TRANSLATE PROGRAM CICSVSE
// ASSGN SYSIPT,SYSRDR
// EXEC IESINSRT
$$$ LST DISP=D,CLASS=Q,PRI=3
// JOB COMWSR1 COMPILE PROGRAM CICSVSE
// SETPARM CATALOG=1

```

```

// IF CATALOG = 1 THEN
// GOTO CAT
// OPTION ERRS,SXREF,SYM,LIST,NODECK
// GOTO ENDCAT
/. CAT
// LIBDEF OBJ,SEARCH=PRD2.CICSR           COBOL II EXTERNAL REFERENCES
// LIBDEF PHASE,CATALOG=MQMUSR1.PHASE
// OPTION ERRS,SXREF,SYM,CATAL,NODECK
// PHASE CICSVSE,*
// INCLUDE DFHECI
/. ENDCAT
// EXEC IGYCRCTL,SIZE=IGYCRCTL
// CBL LIB,APOST,NOADV,NODYNAM,RENT,RESIDENT,BUF(4096)
* $$ END
// ON $CANCEL OR $ABEND GOTO ENDJ2
// OPTION NOLIST,NODUMP,DECK
// EXEC DFHECP1$,SIZE=512K
// CBL XOPTS(COBOL2 CICS DEBUG)
* $$ SLI ICCF=(CICSVSE),LIB=(0091)
/*
/. ENDJ2
// EXEC IESINSRT
/*
// IF CATALOG NE 1 OR $MRC GT 4 THEN
// GOTO NOLNK
// EXEC LNKEDT,SIZE=256K
/. NOLNK
#&
$ $$ EOJ
* $$ END
/&
* $$ EOJ

```

C.4 BMS Maps (File Name NEW1MS.MAP)

```

NEWIMS DFHMSD TYPE=&SYSPARM,MODE=INOUT,LANG=COBOL,STORAGE=AUTO,      X
        DSATTS=(COLOR,HILIGHT),MAPATTS=(COLOR,HILIGHT),              X
        TIOAPFX=YES,COLOR=GREEN,HILIGHT=OFF
TO00   DFHMDI SIZE=(24,80),LINE=1,COLUMN=1
        DFHMDF POS=(1,1),LENGTH=18,ATTRB=(UNPROT,NORM,IC),          X
        INITIAL=' TECC: Session over'
TO01   DFHMDI SIZE=(24,80),LINE=1,COLUMN=1,COLOR=NEUTRAL
        DFHMDF POS=(1,25),LENGTH=30,ATTRB=(PROT,NORM),HILIGHT=REVERSE, X
        COLOR=YELLOW,INITIAL=' Credit Account Demonstration '
        DFHMDF POS=(1,56),LENGTH=1,JUSTIFY=(LEFT,ZERO),              X
        ATTRB=(PROT,NORM),COLOR=GREEN
        DFHMDF POS=(1,68),LENGTH=11,ATTRB=(PROT,NORM),COLOR=BLUE,    X
        INITIAL=' DDWIMS.TO01'
        DFHMDF POS=(3,27),LENGTH=26,JUSTIFY=(LEFT,ZERO),              X
        ATTRB=(PROT,NORM),COLOR=YELLOW,                               X
        INITIAL=' Account Manager: Main Menu'
        DFHMDF POS=(6,20),LENGTH=2,ATTRB=(PROT,NORM),INITIAL='1.'
        DFHMDF POS=(6,23),LENGTH=20,ATTRB=(PROT,NORM),COLOR=GREEN,    X
        INITIAL=' Show account details'
        DFHMDF POS=(8,20),LENGTH=2,ATTRB=(PROT,NORM),INITIAL='2.'
        DFHMDF POS=(8,23),LENGTH=14,ATTRB=(PROT,NORM),COLOR=GREEN,    X
        INITIAL=' Add an account'
        DFHMDF POS=(10,20),LENGTH=2,ATTRB=(PROT,NORM),INITIAL='3.'

```

```

DFHMDF POS=(10,23),LENGTH=17,ATTRB=(PROT,NORM),COLOR=GREEN, X
INITIAL='Update an account'
DFHMDF POS=(12,20),LENGTH=2,ATTRB=(PROT,NORM),INITIAL='4.'
DFHMDF POS=(12,23),LENGTH=17,ATTRB=(PROT,NORM),COLOR=GREEN, X
INITIAL='Delete an account'
DFHMDF POS=(17,9),LENGTH=38,ATTRB=(PROT,BRT),COLOR=GREEN, X
INITIAL='Please select an option (1,2,3 or 4) : '
OPTION DFHMDF POS=(17,48),LENGTH=1,JUSTIFY=(LEFT,ZERO), X
ATTRB=(UNPROT,NUM,NORM,IC),HILIGHT=UNDERLINE
DFHMDF POS=(17,50),LENGTH=1,JUSTIFY=(LEFT,ZERO), X
ATTRB=(ASKIP,NORM),COLOR=GREEN
DFHMDF POS=(20,1),LENGTH=17,ATTRB=(PROT,NORM), X
INITIAL='Resource manager:'
RMUSED1 DFHMDF POS=(20,19),LENGTH=20,ATTRB=(PROT,NORM)
MESS1 DFHMDF POS=(22,1),LENGTH=79,ATTRB=(PROT,NORM),COLOR=RED
DFHMDF POS=(23,1),LENGTH=79,ATTRB=(PROT,NORM),COLOR=YELLOW, X
INITIAL='-----X
-----'
DFHMDF POS=(24,1),LENGTH=8,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM),COLOR=YELLOW,INITIAL='PF3=Exit'
DFHMDF POS=(24,26),LENGTH=29,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM),COLOR=YELLOW,
INITIAL='PF6=Use Next Resource Manager'
DFHMDF POS=(24,71),LENGTH=9,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM),COLOR=YELLOW,INITIAL='PF12=Exit'
T002 DFHMDF SIZE=(24,80),LINE=1,COLUMN=1
DFHMDF POS=(1,25),LENGTH=30,ATTRB=(PROT,NORM),HILIGHT=REVERSE, X
COLOR=YELLOW,INITIAL='Credit Account Demonstration '
DFHMDF POS=(1,56),LENGTH=1,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM)
DFHMDF POS=(1,69),LENGTH=11,ATTRB=(PROT,NORM),COLOR=BLUE, X
INITIAL='DDWIMS.T002'
DFHMDF POS=(3,22),LENGTH=37,ATTRB=(PROT,NORM),COLOR=YELLOW, X
INITIAL='Account Manager: Show account details'
DFHMDF POS=(9,16),LENGTH=31,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM),
INITIAL='Enter Surname of account owner:'
SUR2 DFHMDF POS=(9,48),LENGTH=15,ATTRB=(UNPROT,NORM,IC), X
HILIGHT=UNDERLINE,COLOR=NEUTRAL
DFHMDF POS=(9,64),LENGTH=1,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM)
DFHMDF POS=(11,39),LENGTH=2,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,BRT),INITIAL='OR'
DFHMDF POS=(13,26),LENGTH=21,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM),INITIAL='Enter account number:'
ACC2 DFHMDF POS=(13,48),LENGTH=5,JUSTIFY=(RIGHT,ZERO), X
ATTRB=(UNPROT,NUM,NORM),HILIGHT=UNDERLINE,COLOR=NEUTRAL
DFHMDF POS=(13,54),LENGTH=1,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM)
DFHMDF POS=(20,1),LENGTH=17,ATTRB=(PROT,NORM),COLOR=NEUTRAL, X
INITIAL='Resource manager:'
RMUSED2 DFHMDF POS=(20,19),LENGTH=20,ATTRB=(PROT,NORM),COLOR=NEUTRAL
MESS2 DFHMDF POS=(22,1),LENGTH=79,ATTRB=(PROT,NORM),COLOR=RED
DFHMDF POS=(23,1),LENGTH=79,ATTRB=(PROT,NORM),COLOR=YELLOW, X
INITIAL='-----X
-----'
DFHMDF POS=(24,1),LENGTH=10,ATTRB=(PROT,NORM),COLOR=YELLOW, X
INITIAL='PF3=Return'
DFHMDF POS=(24,71),LENGTH=9,ATTRB=(PROT,NORM),COLOR=YELLOW, X

```

```

INITIAL=' PF12=Exit'
T003 DFHMDI SIZE=(24,80),LINE=1,COLUMN=1
DFHMDF POS=(1,25),LENGTH=30,ATTRB=(PROT,NORM),HILIGHT=REVERSE, X
COLOR=YELLOW,INITIAL=' Credit Account Demonstration '
DFHMDF POS=(1,56),LENGTH=1,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM)
DFHMDF POS=(1,69),LENGTH=11,ATTRB=(PROT,NORM),COLOR=BLUE, X
INITIAL=' DDW1MS.T003'
DFHMDF POS=(3,24),LENGTH=32,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM),COLOR=YELLOW, X
INITIAL=' Account Manager: Account details'
DFHMDF POS=(8,20),LENGTH=8,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM),INITIAL=' Surname:'
SUR3 DFHMDF POS=(8,44),LENGTH=15,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM)
DFHMDF POS=(8,60),LENGTH=1,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM)
DFHMDF POS=(10,20),LENGTH=11,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM),INITIAL=' First name:'
FOR3 DFHMDF POS=(10,44),LENGTH=10,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM)
DFHMDF POS=(10,55),LENGTH=1,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM)
DFHMDF POS=(12,20),LENGTH=8,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM),INITIAL=' Account:'
ACC3 DFHMDF POS=(12,44),LENGTH=5,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM)
DFHMDF POS=(12,50),LENGTH=1,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM)
DFHMDF POS=(14,20),LENGTH=8,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM),INITIAL=' Address:'
ADD13 DFHMDF POS=(14,44),LENGTH=15,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM)
DFHMDF POS=(14,60),LENGTH=1,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM)
ADD23 DFHMDF POS=(15,44),LENGTH=15,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM)
DFHMDF POS=(15,60),LENGTH=1,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM)
ADD33 DFHMDF POS=(16,44),LENGTH=15,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM)
DFHMDF POS=(16,60),LENGTH=1,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM)
DFHMDF POS=(18,20),LENGTH=8,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM),INITIAL=' Balance:'
BAL3 DFHMDF POS=(18,44),LENGTH=4,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM)
DFHMDF POS=(18,49),LENGTH=1,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM)
DFHMDF POS=(20,1),LENGTH=17,ATTRB=(PROT,NORM),COLOR=NEUTRAL, X
INITIAL=' Resource manager:'
RMUSED3 DFHMDF POS=(20,19),LENGTH=20,ATTRB=(PROT,NORM),COLOR=NEUTRAL
MESS3 DFHMDF POS=(22,1),LENGTH=79,ATTRB=(PROT,NORM),COLOR=RED
DFHMDF POS=(23,1),LENGTH=79,ATTRB=(PROT,NORM),COLOR=YELLOW, X
INITIAL=' -----X
-----'
DFHMDF POS=(24,1),LENGTH=16,ATTRB=(PROT,NORM),COLOR=YELLOW, X
INITIAL=' PF3/Enter=Return'
DFHMDF POS=(24,71),LENGTH=9,ATTRB=(PROT,NORM),COLOR=YELLOW, X

```

```

INITIAL=' PF12=Exit'
T004 DFHMDI SIZE=(24,80),LINE=1,COLUMN=1,COLOR=NEUTRAL
DFHMDF POS=(1,25),LENGTH=30,ATTRB=(PROT,NORM),HILIGHT=REVERSE, X
COLOR=YELLOW,INITIAL=' Credit Account Demonstration '
DFHMDF POS=(1,56),LENGTH=1,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM),COLOR=GREEN
DFHMDF POS=(1,69),LENGTH=11,ATTRB=(PROT,NORM),COLOR=BLUE, X
INITIAL=' DDWIMS.T004'
DFHMDF POS=(3,25),LENGTH=31,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM),COLOR=YELLOW, X
INITIAL=' Account Manager: Add an account'
DFHMDF POS=(8,23),LENGTH=8,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM),COLOR=GREEN,INITIAL=' Surname:'
SUR4 DFHMDF POS=(8,40),LENGTH=15,ATTRB=(UNPROT,NORM,IC,FSET), X
HILIGHT=UNDERLINE
DFHMDF POS=(8,56),ATTRB=(ASKIP,NORM),LENGTH=1
DFHMDF POS=(10,23),LENGTH=11,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM),COLOR=GREEN,INITIAL=' First name:'
FOR4 DFHMDF POS=(10,40),LENGTH=10,ATTRB=(UNPROT,NORM,FSET), X
HILIGHT=UNDERLINE
DFHMDF POS=(10,51),ATTRB=(ASKIP,NORM),LENGTH=1
DFHMDF POS=(12,23),LENGTH=8,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM),COLOR=GREEN,INITIAL=' Address:'
ADD14 DFHMDF POS=(12,40),LENGTH=15,ATTRB=(UNPROT,NORM,FSET), X
HILIGHT=UNDERLINE
DFHMDF POS=(12,56),ATTRB=(ASKIP,NORM),LENGTH=1
ADD24 DFHMDF POS=(13,40),LENGTH=15,ATTRB=(UNPROT,NORM,FSET), X
HILIGHT=UNDERLINE
DFHMDF POS=(13,56),ATTRB=(ASKIP,NORM),LENGTH=1
ADD34 DFHMDF POS=(14,40),LENGTH=15,ATTRB=(UNPROT,NORM,FSET), X
HILIGHT=UNDERLINE
DFHMDF POS=(14,56),ATTRB=(ASKIP,NORM),LENGTH=1
DFHMDF POS=(16,23),LENGTH=8,ATTRB=(PROT,NORM,FSET),COLOR=GREEN,X
INITIAL=' Balance:'
BAL4 DFHMDF POS=(16,40),LENGTH=4,JUSTIFY=(RIGHT,ZERO), X
ATTRB=(UNPROT,NUM,NORM,FSET),HILIGHT=UNDERLINE
DFHMDF POS=(16,45),LENGTH=1,ATTRB=(PROT,NORM),COLOR=GREEN
DFHMDF POS=(20,1),LENGTH=17,ATTRB=(PROT,NORM), X
INITIAL=' Resource manager:'
RMUSED4 DFHMDF POS=(20,19),LENGTH=20,ATTRB=(PROT,NORM)
MESS4 DFHMDF POS=(22,1),LENGTH=79,ATTRB=(PROT,NORM),COLOR=RED
DFHMDF POS=(23,1),LENGTH=79,ATTRB=(PROT,NORM),COLOR=YELLOW, X
INITIAL=' -----X
-----'
DFHMDF POS=(24,1),LENGTH=10,ATTRB=(PROT,NORM),COLOR=YELLOW, X
INITIAL=' PF3=Return'
DFHMDF POS=(24,71),LENGTH=9,ATTRB=(PROT,NORM),COLOR=YELLOW, X
INITIAL=' PF12=Exit'
T005 DFHMDI SIZE=(24,80),LINE=1,COLUMN=1,COLOR=YELLOW
DFHMDF POS=(1,25),LENGTH=30,ATTRB=(PROT,NORM),HILIGHT=REVERSE, X
INITIAL=' Credit Account Demonstration '
DFHMDF POS=(1,56),LENGTH=1,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM),COLOR=GREEN
DFHMDF POS=(1,69),LENGTH=11,ATTRB=(PROT,NORM),COLOR=BLUE, X
INITIAL=' DDWIMS.T005'
DFHMDF POS=(3,25),LENGTH=31,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM), X
INITIAL=' Account Manager: Delete account'
DFHMDF POS=(8,19),LENGTH=34,JUSTIFY=(RIGHT,ZERO), X

```

		ATTRB=(PROT,NORM),COLOR=GREEN,	X
		INITIAL='Enter account number for deletion:'	
ACC5	DFHMDF	POS=(8,54),LENGTH=5,JUSTIFY=(RIGHT,ZERO),	X
		ATTRB=(UNPROT,NUM,NORM,IC),HILIGHT=UNDERLINE,	X
		COLOR=NEUTRAL	
	DFHMDF	POS=(8,60),LENGTH=1,JUSTIFY=(RIGHT,ZERO),	X
		ATTRB=(PROT,NORM),COLOR=GREEN	
	DFHMDF	POS=(20,1),LENGTH=17,ATTRB=(PROT,NORM),COLOR=NEUTRAL,	X
		INITIAL='Resource manager:'	
RMUSED5	DFHMDF	POS=(20,19),LENGTH=20,ATTRB=(PROT,NORM),COLOR=NEUTRAL	
MESS5	DFHMDF	POS=(22,1),LENGTH=79,ATTRB=(PROT,NORM),COLOR=RED	
	DFHMDF	POS=(23,1),LENGTH=79,ATTRB=(PROT,NORM),	X
		INITIAL='-----X	
		-----'	
	DFHMDF	POS=(24,1),LENGTH=10,ATTRB=(PROT,NORM),	X
		INITIAL='PF3=Return'	
	DFHMDF	POS=(24,71),LENGTH=9,ATTRB=(PROT,NORM),	X
		INITIAL='PF12=Exit'	
T006	DFHMDI	SIZE=(24,80),LINE=1,COLUMN=1	
	DFHMDF	POS=(1,25),LENGTH=30,ATTRB=(PROT,NORM),HILIGHT=REVERSE,	X
		COLOR=YELLOW,INITIAL='Credit Account Demonstration'	
	DFHMDF	POS=(1,56),LENGTH=1,JUSTIFY=(LEFT,ZERO),	X
		ATTRB=(PROT,NORM)	
	DFHMDF	POS=(1,69),LENGTH=11,ATTRB=(PROT,NORM),COLOR=BLUE,	X
		INITIAL='DDWIMS.T006'	
	DFHMDF	POS=(3,25),LENGTH=31,JUSTIFY=(LEFT,ZERO),	X
		ATTRB=(PROT,NORM),COLOR=YELLOW,	X
		INITIAL='Account Manager: Account update'	
	DFHMDF	POS=(6,2),LENGTH=23,ATTRB=(PROT,NORM),	X
		INITIAL='Update account entries:'	
	DFHMDF	POS=(8,20),LENGTH=8,JUSTIFY=(LEFT,ZERO),	X
		ATTRB=(PROT,NORM),INITIAL='Surname:'	
SUR6	DFHMDF	POS=(8,44),LENGTH=15,ATTRB=(UNPROT,NORM,IC,FSET),	X
		HILIGHT=UNDERLINE,COLOR=NEUTRAL	
	DFHMDF	POS=(8,60),LENGTH=1,JUSTIFY=(LEFT,ZERO),	X
		ATTRB=(PROT,NORM)	
	DFHMDF	POS=(10,20),LENGTH=11,JUSTIFY=(LEFT,ZERO),	X
		ATTRB=(PROT,NORM),INITIAL='First name:'	
FOR6	DFHMDF	POS=(10,44),LENGTH=10,ATTRB=(UNPROT,NORM,FSET),	X
		HILIGHT=UNDERLINE,COLOR=NEUTRAL	
	DFHMDF	POS=(10,55),LENGTH=1,JUSTIFY=(LEFT,ZERO),	X
		ATTRB=(PROT,NORM)	
	DFHMDF	POS=(12,20),LENGTH=8,JUSTIFY=(LEFT,ZERO),	X
		ATTRB=(PROT,NORM),INITIAL='Account:'	
ACC6	DFHMDF	POS=(12,44),LENGTH=5,JUSTIFY=(LEFT,ZERO),	X
		ATTRB=(PROT,NORM)	
	DFHMDF	POS=(14,20),LENGTH=8,JUSTIFY=(LEFT,ZERO),	X
		ATTRB=(PROT,NORM),INITIAL='Address:'	
ADD16	DFHMDF	POS=(14,44),LENGTH=15,ATTRB=(UNPROT,NORM,FSET),	X
		HILIGHT=UNDERLINE,COLOR=NEUTRAL	
	DFHMDF	POS=(14,60),LENGTH=1,JUSTIFY=(LEFT,ZERO),	X
		ATTRB=(PROT,NORM)	
ADD26	DFHMDF	POS=(15,44),LENGTH=15,ATTRB=(UNPROT,NORM,FSET),	X
		HILIGHT=UNDERLINE,COLOR=NEUTRAL	
	DFHMDF	POS=(15,60),LENGTH=1,JUSTIFY=(LEFT,ZERO),	X
		ATTRB=(PROT,NORM)	
ADD36	DFHMDF	POS=(16,44),LENGTH=15,ATTRB=(UNPROT,NORM,FSET),	X
		HILIGHT=UNDERLINE,COLOR=NEUTRAL	
	DFHMDF	POS=(16,60),LENGTH=1,JUSTIFY=(LEFT,ZERO),	X

```

ATTRB=(PROT,NORM)
DFHMDF POS=(18,20),LENGTH=8,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM),INITIAL='Balance:'
BAL6 DFHMDF POS=(18,44),LENGTH=4,ATTRB=(UNPROT,NORM,FSET), X
HIGHLIGHT=UNDERLINE,COLOR=NEUTRAL
DFHMDF POS=(18,49),LENGTH=1,ATTRB=(UNPROT,NORM,FSET), X
COLOR=NEUTRAL
DFHMDF POS=(20,1),LENGTH=17,ATTRB=(PROT,NORM),COLOR=NEUTRAL, X
INITIAL='Resource manager:'
RMUSED6 DFHMDF POS=(20,19),LENGTH=20,ATTRB=(PROT,NORM),COLOR=NEUTRAL
MESS6 DFHMDF POS=(22,1),LENGTH=79,ATTRB=(PROT,NORM),COLOR=RED
DFHMDF POS=(23,1),LENGTH=79,ATTRB=(PROT,NORM),COLOR=YELLOW, X
INITIAL='-----X
-----'
DFHMDF POS=(24,1),LENGTH=16,ATTRB=(PROT,NORM),COLOR=YELLOW, X
INITIAL='PF3/Enter=Return'
DFHMDF POS=(24,71),LENGTH=9,ATTRB=(PROT,NORM),COLOR=YELLOW, X
INITIAL='PF12=Exit'
T007 DFHMDF SIZE=(24,80),LINE=1,COLUMN=1,COLOR=YELLOW
DFHMDF POS=(1,25),LENGTH=30,ATTRB=(PROT,NORM),HIGHLIGHT=REVERSE, X
INITIAL='Credit Account Demonstration'
DFHMDF POS=(1,56),LENGTH=1,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM),COLOR=GREEN
DFHMDF POS=(1,69),LENGTH=11,ATTRB=(PROT,NORM),COLOR=BLUE, X
INITIAL='DDWIMS.T007'
DFHMDF POS=(3,25),LENGTH=31,JUSTIFY=(LEFT,ZERO), X
ATTRB=(PROT,NORM), X
INITIAL='Account Manager: Update account'
DFHMDF POS=(8,19),LENGTH=32,JUSTIFY=(RIGHT,ZERO), X
ATTRB=(PROT,NORM),COLOR=GREEN, X
INITIAL='Enter account number for update:'
ACC7 DFHMDF POS=(8,52),LENGTH=5,JUSTIFY=(RIGHT,ZERO), X
ATTRB=(UNPROT,NUM,NORM,IC),HIGHLIGHT=UNDERLINE, X
COLOR=NEUTRAL
DFHMDF POS=(8,58),LENGTH=1,JUSTIFY=(RIGHT,ZERO), X
ATTRB=(PROT,NORM),COLOR=GREEN
DFHMDF POS=(20,1),LENGTH=17,ATTRB=(PROT,NORM),COLOR=NEUTRAL, X
INITIAL='Resource manager:'
RMUSED7 DFHMDF POS=(20,19),LENGTH=20,ATTRB=(PROT,NORM),COLOR=NEUTRAL
MESS7 DFHMDF POS=(22,1),LENGTH=79,ATTRB=(PROT,NORM),COLOR=RED
DFHMDF POS=(23,1),LENGTH=79,ATTRB=(PROT,NORM), X
INITIAL='-----X
-----'
DFHMDF POS=(24,1),LENGTH=10,ATTRB=(PROT,NORM), X
INITIAL='PF3=Return'
DFHMDF POS=(24,71),LENGTH=9,ATTRB=(PROT,NORM), X
INITIAL='PF12=Exit'
NEWIMS DFHMDF TYPE=FINAL
END

```

C.5 COBOL Directives (File COBOL.DIR)

```

ALIGN"4"
ANIM
NOALTER
ANS85
COBOL370
NODATALIT

```

```

DEFAULTBYTE"0"
IBMCOMP
OPTSPEED
OSVS
NOREALOVL
NOSEG
TARGET"386-16"
VSC2

```

C.6 CICS OS/2Tailor Environment Variables (File CICSENV.COMD)

```

/* CICSENV.COMD */
.
.
.
' SET CICS DRV=D:'
' SET CICS HLD=CICS200'
SQL_Inc    ='
SQL_Lib    ='
SQL_Cpy    ='
UserWrk    ='
CicsRgrp   ='
UserCobdir =' D:\CICS200\UTIL'
UserCobcpy =' D:\CICS200\COPYBOOK'
UserCobwrk ='
UserCobxil ='
UserCobidy =' D:\CICS200\BIN'
UserCwrk   ='
UserInclude='
UserIcc    ='
UserTRL    ='
UserEdit   ='
UserText   ='
UserTrace  ='
.
.
.
Return pname
/*-----*/

```

C.7 Create VSAM Base File

```

* $$ JOB JNM=CREATBAS,CLASS=0,DISP=D,NTFY=YES
// JOB WSR1 DEFINE FILE
// EXEC IDCAMS,SIZE=AUTO
DEFINE CLUSTER ( -
    NAME (TECHDEM.BASEFILE ) -
    TRACKS (2 2 ) -
    SHAREOPTIONS (4) -
    RECORDSIZE (80 80 ) -
    VOLUMES (SYSWK3 ) -
    NOREUSE -
    INDEXED -
    FREESPACE (15 7) -
    KEYS (5 0 ) -
    TO (99366) -
    DATA (NAME (TECHDEM.BASEFILE.@@ ) -

```



```

CONTROLINTERVALSIZE (4096 ) -
INDEX (NAME (TECHDEM.BASEFILE.@I@
CATALOG (UCATEZB ) ) -
IF LASTCC NE 0 THEN CANCEL JOB
/*
// OPTION STDLABEL=ADD
// DLBL TECHBAS,' TECHDEM.BASEFILE',, VSAM, X
CAT=UCATEZB
/*
// EXEC IESVCLUP,SIZE=AUTO
A TECHDEM.BASEFILE TECHBAS UCATEZB STDLABUP
/*
/&
* $$ EOJ

```

C.8 Load VSAM Base File

```

* $$ JOB JNM=LOADDUMM,CLASS=A,DISP=D,NTFY=YES
// JOB WSR1 LOAD FILE FROM ICCF MEMBER
// DLBL LOADFL,' TECHDEM.BASEFILE',, VSAM, X
CAT=UCATEZB
// EXEC IESVSMLD,SIZE=AUTO
80,K,LOADFL
/INCLUDE DUMMY
/*
// EXEC IDCAMS,SIZE=AUTO
PRINT INFILE (LOADFL) CHARACTER
/*
/&
* $$ EOJ

```

C.9 Sample VSAM Load Program

```

*****
* Initialize VSAM FILE
*****
IDENTIFICATION DIVISION.
PROGRAM-ID. INITVSAM.
ENVIRONMENT DIVISION.
*****
DATA DIVISION.

WORKING-STORAGE SECTION.

01 SERVER-CUSTOMER-RECORD.
02 ACCOUNTO PIC X(5) VALUE '00001'.
02 SURNAMEO PIC X(15) VALUE 'A'.
02 FIRST-NAMEO PIC X(10) VALUE 'A'.
02 ADDRESSO PIC X(45) VALUE 'A'.
02 BALANCEO PIC S9(4) COMP VALUE 1.
01 MSG1 PIC X(40).

COPY DFHAID.

*****
PROCEDURE DIVISION.

```

MAINLINE SECTION.

```
* Convert trailing LOW-VALUES to SPACES in the customer record
  INSPECT SURNAMEO REPLACING ALL LOW-VALUES BY SPACES
  INSPECT FIRST-NAMEO REPLACING ALL LOW-VALUES BY SPACES
  INSPECT ADDRESSO REPLACING ALL LOW-VALUES BY SPACES
```

```
EXEC CICS WRITE
      FILE(' TECHBAS')
      FROM(SERVER-CUSTOMER-RECORD)
      RIDFLD(ACCOUNTO)
      NOHANDLE
      END-EXEC.
```

EVALUATE EIBRESP

```
  WHEN DFHRESP(NORMAL)
    MOVE 'INIT Succesful.' TO MSG1
    EXEC CICS SEND TEXT FROM(MSG1) LENGTH(60) FREEKB END-EXEC
```

```
  WHEN DFHRESP(DUPREC)
    MOVE 'File is already initialized.' TO MSG1
    EXEC CICS SEND TEXT FROM(MSG1) LENGTH(60) FREEKB END-EXEC
```

```
  WHEN OTHER
    MOVE 'Something is wrong.....' TO MSG1
    EXEC CICS SEND TEXT FROM(MSG1) LENGTH(60) FREEKB END-EXEC
```

```
END-EVALUATE.
EXEC CICS RETURN
      END-EXEC.
```

EXIT.

C.10 Define VSAM Alternate Index

```
* $$ JOB JNM=BLDINDEX,CLASS=0,DISP=D,NTFY=YES
// JOB WSR1 DEFINE ALTERNATE INDEX
// EXEC IDCAMS,SIZE=AUTO
DEFINE ALTERNATEINDEX ( -
      NAME (TECHDEM.ALTXFILE.@A@                               ) -
      TRACKS (00000001 00000001   ) -
      TO (99366) -
      SHAREOPTIONS (4) -
      RECORDSIZE (000045 000045) -
      VOLUMES (SYSWK3 ) -
      RELATE (TECHDEM.BASEFILE                               ) -
      KEYS (15 5   ) -
      FREESPACE (15 7)) -
      DATA (NAME (TECHDEM.ALTXFILE.@A@D@                     )) -
      INDEX (NAME (TECHDEM.ALTXFILE.@A@I@                     )) -
      CATALOG (UCATEZB                                       )
  IF LASTCC NE 0 THEN CANCEL JOB
  DEFINE PATH ( -
      NAME (TECHDEM.ALTXFILE                               ) -
      PATHENTRY (TECHDEM.ALTXFILE.@A@                       ) -
      TO (99366) -
      UPDATE) -
```

```

        CATALOG (UCATEZB                                )
    IF LASTCC NE 0 THEN -
DELETE (TECHDEM.ALTXFILE.@A@                          ) -
    ALTERNATEINDEX -
    PURGE -
    CATALOG (UCATEZB                                )
    IF LASTCC NE 0 THEN CANCEL JOB
/*
// OPTION STDLABEL=ADD
// DLBL TECHALT,'TECHDEM.ALTXFILE',,VSAM,              X
    CAT=UCATEZB
/*
// EXEC IESVCLUP,SIZE=AUTO
A TECHDEM.ALTXFILE                                     TECHALT UCATEZB      STDLABUP
/*
/&
* $$ EOJ

```

C.11 Build VSAM Alternate Index

```

* $$ JOB JNM=LOADALT,CLASS=A,DISP=D,NTFY=YES
// JOB WSR1 LOAD ALTERNATE INDEX
// DLBL IJSYSUC,'UCATEZB',,VSAM
// EXEC IDCAMS,SIZE=AUTO
BLDINDEX INDATASET (TECHDEM.BASEFILE                   ) -
    OUTDATASET (TECHDEM.ALTXFILE                       ) -
    CATALOG (VSESP.USER.CATALOG) -
    WORKVOLUMES (SYSWK1) -
    EXTERNALSORT
/*
/&
* $$ EOJ

```

List of Abbreviations

APA	all points addressable	LAPS	LAN Adapter and Protocol Support
APPC	Advanced Program to Program Communications	LPP	Licensed Program Product
ASCII	American National Standard for Information Interchange	LU	Logical Unit
BMS	Basic Mapping Support	NetBIOS	Network Basic Input Output System
CD	Communications Definition	PTF	Program Temporary Fix
CDS	Cell Directory Service	PD	Program Definition
CICS	Customer Information Control System	PROFS	Professional Office System
DCE	Distributed Computing Environment	RTPN	Remote Transaction Program Name
DPL	Distributed Program Link	RD	Region Definition
DTP	Distributed Transaction Processing	RPC	Remote Procedure Call
EBCDIC	Extended Binary-Coded Decimal Interchange Code	SFS	Structured File Server
FS	Function Shipping	SMIT	System Management Interface Tool
GUI	Graphical User Interface	SSCP	System Services Control Point
IBM	International Business Machines Corporation	TCP/IP	Transmission Control Protocol/Internet Protocol
ISC	Inter System Communication	TD	Transaction Definition
ITSO	International Technical Support Organization	TPN	Transaction Program Name
		TR	Transaction Routing
		VSAM	Virtual Storage Access Method

Index

Numerics

- 3172 11
 - customization 13
 - definition in IOCDS 12
 - VM definition 12

A

- abbreviations 163
- ACBNAME parameter 19
- ACCESSMETHOD parameter 23
- ACCMETH parameter 20
- acronyms 163
- ACSGTAB 30
- AIX SNA Services/6000 75
- APPL major node 18
- APPLID parameter 19
- asynchronous processing 68
- attach security 43
- AUTOCONNECT parameter 25

B

- BUILDCHAIN parameter 25

C

- call mode 77
- CECI 98
- CEDA 19, 22, 40
- CEMT command 113
- CHNASSY parameter 20
- CICS/6000 55
 - AIX prerequisites 62
 - CICS/6000 client 55
 - CICS/6000 server 55
 - configure DCE 63
 - configure Encina 63
 - create and start CICS/6000 region 64
 - hardware used 57
 - install and configure client 65
 - install LPPs 62
 - installation summary 64
 - LPPs required for ISC 61
 - LPPs required for server 61
 - sample network configuration 58
 - server installation 61
 - software required/used 57
- CICS/6000 and SNA 68
- CICS/6000 and TCP/IP 68
- CICS/6000 ISC 67
 - asynchronous processing 68
 - Distributed Program Link 67
 - Distributed Transaction Processing 68

- CICS/6000 ISC (*continued*)
 - Function Shipping 68
- CICS/6000 sample application 103
 - application description 103
 - application development 104
 - CICS/6000 table entries 105
- CICS/6000 to CICS/VSE connection 67
 - ACF/VTAM customization 69
 - AIX customization 73
 - AIX SNA Services/6000 75
 - CD/Encina PPC Gateway attributes 92
 - CD/SNA attributes 90
 - CICS/6000 customization 73
 - CICS/6000 to CICS/VSE considerations 94
 - CICS/VSE customization 70
 - customization summary 100
 - Encina PPC Gateway monitoring 97
 - operations 95
 - RD attributes 89
 - token ring adapter customization 74, 75
 - VSE/ESA host customization 69
- CICS/OS2
 - CEDA 40
 - customization 40
 - LAPS installation 27
 - PCT 44
 - PPT 45
 - SIT 41
 - TCS 42
 - terminal control table 42
- CICS/VSE
 - CEDA 19, 22
 - CEMT command 113
 - connection definition 23
 - CRTE transaction 40
 - data conversion program 21
 - DFHFCT 21
 - EAS= parameter 18
 - macro definition 19
 - OS/2 support 20
 - PARSESS 18
 - session definition 24
 - SIT 19
 - terminal control program 20
 - VTAM customization 15
- CICSENV command 50
- CICSMAP command 49
- CICSTCL command 50
- CLSDST macro 25
- COBDIR variable 50
- COBOL directives 50
- code page 43
- Communications Definitions (CD) 90

Communications Manager

- ACSGTAB 30
- configuration 30
- CRSR transaction 39
- CRTE transaction 40
- DLC token-ring 30
- link name 32
- local node characteristics 31
- logon mode 36
- MACADDR 32
- partner LU 35
- SNA connection 32
- SNA features 32
- transaction programs 36
- translation table file name 30

connection definition 71

CONNECTION parameter 23, 24

contention winners 25, 36, 72

control point profile for SNA 88

CONVERSATION type 85

CRSR transaction 39

CRTE transaction 40

customization 29

D

data conversion 94

DCE cell 62, 63, 65

DFHCNV program 21

DFHFCT 21

DIALNO 16

DIALNO parameter 29

DISCREQ parameter 25

Distributed Computing Environment (DCE) 75

Distributed Program Link (DPL) 67

Distributed Transaction Processing (DTP) 68

DLC token-ring 30

DLOGMODE 16

E

EAS= parameter 18

Encina PPC Gateway 92, 94

Encina Structured File Server 63

EXEC= parameter 20

External Call Interface (ECI) 7

External Presentation Interface (EPI) 7, 55

External Transaction Initiation (ETI) 7

F

FAACLPI.EXE 37

FAARQ.INI 42

function shipping 67

G

GROUP parameter 23, 24

I

IDBLK 16, 31, 32

IDNUM 16, 31, 32

IOCDS 12

ISC connection display 113

ISC= parameter 20

L

LAPS installation 27

LIBPATH statement 45

link name 32

listen mode 77

LOCADDR 16

local LU definition 34

local node characteristics 31

LOGMODE 17

logon mode 36

LU name 34

LU6.2 local LU profile 81

LU6.2 logical connection profile 79

LU6.2 mode list profile 81

LU6.2 mode profile 82

LU6.2 RTPN list profile 85

LU6.2 RTPN profile 86

LU6.2 TPN list profile 83

LU6.2 TPN profile 84

M

MAC address 13

MACADDR 16, 29, 32

MAXIMUM parameter 25

mirror transaction 37, 39

mode name 36, 44, 83

MODENAME parameter 24

N

NCP 74

NETNAME 71

NETNAME parameter 23

network adapter address parameter 29

node address 13

node profile for SNA 87

nonfacility tasks 41

P

PARSESS 18

partner LU 35

partner LU alias 44

PATH 16

PATH macro 29

PCT entries for OS/2 44

PPT entries for OS/2 45

Presentation Manager 8

profile customization 75
PROTOCOL parameter 23, 25
PSERVIC values 17

R

RECEIVESIZE parameter 25
Region Definition (RD) 89
RELREQ parameter 25
RUSIZE parameter 25, 36

S

SENDSIZE parameter 25
session buffer size 43
session count 43
session definition 72
SESSION parameter 24
SINGLESESS parameter 23, 71
SIT 19, 41
skinny client 7
skinny clients 11, 25, 28
SNA
 control point profile 88
 LU6.2 unit profiles 79
 node profiles 87
 physical unit profiles 76
SNA connection 32
subchannel 13
switched major node 29
SWNET major node 16
SYNC level 85
SYSIDNT parameter 19
SYSTEM CONFIG file 12

T

TCS 42
token-ring attachment profile 76
token-ring customization 74
token-ring data link control logical link 78
token-ring data link control physical link 78
token-ring hardware address 74
TRACE 114
translation table file name 30

V

VM/ESA*
 3172 definition 12
 SYSTEM CONFIG file 12
 VM customization 12
 VSE user directory 12
VSE/ESA
 user directory 12
VTAM
 APPL major node 18
 CLSDST macro 25
 configuration list 11

VTAM (continued)

DIALNO 16
DIALNO parameter 29
display sessions command 113
DLOGMODE 16
IDBLK 16, 31, 32
IDNUM 16, 31, 32
LOCADDR 16
LOGMODE 17
PATH 16
PATH macro 29
PU Type 2.1 11
switched major node 29
SWNET major node 16
TRACE 114
VSE customization 15
XCA 15
VTAM trace 97
VTAMDEV parameter 20

X

XCA 15
XID node ID 89

**CICS/VSE to CICS OS/2 and CICS/6000
A Guide to Client/Server Solutions
Publication No. GG24-4262-00**

Your feedback is very important to help us maintain the quality of ITSO Bulletins. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

**Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)**

Overall Satisfaction	_____		
Organization of the book	_____	Grammar/punctuation/spelling	_____
Accuracy of the information	_____	Ease of reading and understanding	_____
Relevance of the information	_____	Ease of finding information	_____
Completeness of the information	_____	Level of technical detail	_____
Value of illustrations	_____	Print quality	_____

Please answer the following questions:

- a) If you are an employee of IBM or its subsidiaries:
Do you provide billable services for 20% or more of your time? Yes____ No____
Are you in a Services Organization? Yes____ No____
- b) Are you working in the USA? Yes____ No____
- c) Was the Bulletin published in time for your needs? Yes____ No____
- d) Did this Bulletin meet your needs? Yes____ No____
- If no, please explain:

What other topics would you like to see in this Bulletin?

What other Technical Bulletins would you like to see published?

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM International Technical Support Organization
Department 3222, Building 71032-02
POSTFACH 1380
71032 BOEBLINGEN
GERMANY

Fold and Tape

Please do not staple

Fold and Tape



Printed in U.S.A.

GG24-4262-00

