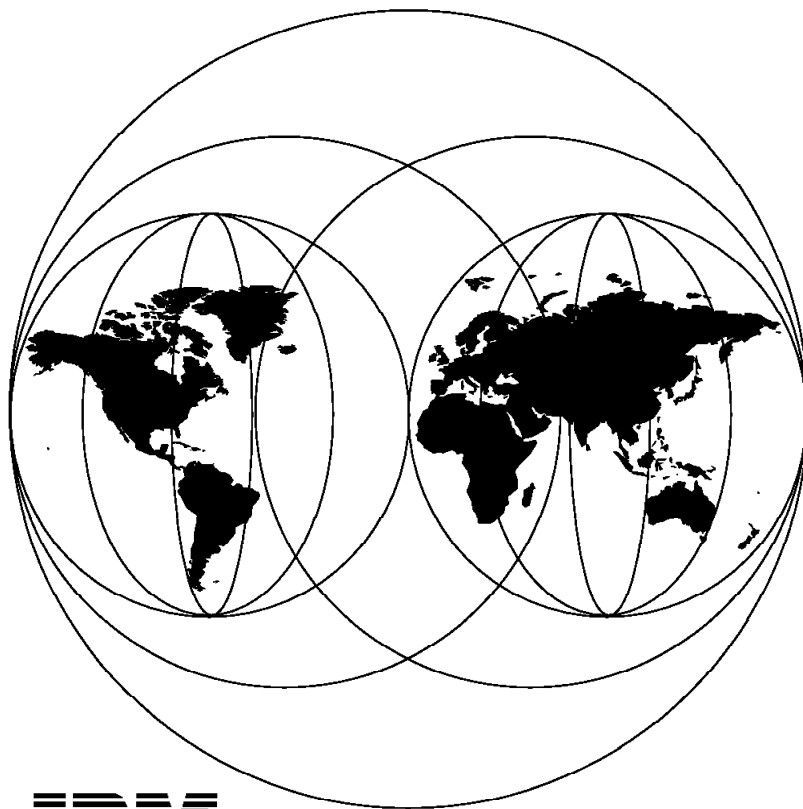


International Technical Support Organization

GG24-4427-00

## **Easy Access to Host Data with Distributed FileManager**

November 1994



**International Technical Support Organization  
San Jose Center**





International Technical Support Organization

GG24-4427-00

## **Easy Access to Host Data with Distributed FileManager**

November 1994

**Take Note!**

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

**First Edition (November 1994)**

This edition applies to Version 1, Release 2 of DFSMS/MVS, Program Number 5695-DF1 for use with the Data Access Services (DAS) component of ADSTAR Distributed Storage Manager (ADSM) on OS/2 2.x systems.

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. 471 Building 070B  
5600 Cottle Road  
San Jose, California 95193-0001

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 the functions of Distributed FileManager (DFM) in an OS/2-MVS/ESA environment and explains how to use Distributed FileManager to access record-oriented and stream-oriented data sets. It describes the setup for the OS/2, MVS/ESA, and OS/400 systems and provides sample applications written in high level languages such as C and PL/I.

This document is written for OS/2 workstation users and application designers who want to know how to use Distributed FileManager to access data sets that reside on a host system (MVS/ESA or OS/400) from an OS/2 workstation. Some knowledge of Distributed FileManager OS/2, APPC, and programming languages is assumed.

(121 pages)



---

# Contents

<b>Abstract</b>	iii
<b>Special Notices</b>	xi
<b>Preface</b>	xiii
How This Document Is Organized	xiii
Related Publications	xiii
International Technical Support Organization Publications	xiv
Acknowledgments	xv
<b>Chapter 1. Introduction to Distributed FileManager</b>	1
1.1 What Is Distributed FileManager?	1
1.2 Distributed FileManager Components and Data Flow	2
1.3 Distributed FileManager Working Environment	4
<b>Chapter 2. DFM/2 and DFM/MVS</b>	5
2.1 MVS/ESA Environment	5
2.1.1 ACF/VTAM Definitions	5
2.1.2 APPC/MVS Definitions	7
2.1.3 DFM/MVS Definitions	9
2.1.4 DFSMS/MVS ACS Routines Definition	10
2.2 OS/2 Environment	10
2.2.1 Communications Manager/2 Definitions	11
2.2.2 DFM/2 Definitions	25
2.3 Record Level I/O Samples	30
2.3.1 C Application	30
2.3.2 PL/I Application	39
2.4 Stream I/O Samples	46
2.4.1 Stream I/O Support	46
2.4.2 Stream I/O Setup	46
2.4.3 Stream I/O Usage Scenarios	47
<b>Chapter 3. DFM/2 and AS/400 Distributed Data Management</b>	51
3.1 AS/400 Environment	51
3.2 OS/2 Environment	55
3.2.1 Communications Manager/2 Definitions	55
3.2.2 DFM/2 Definitions	60
3.3 Record Level I/O Sample	61
3.4 Stream I/O Sample	67
<b>Appendix A. Sample Diskette Description</b>	71
<b>Appendix B. Sample Network Definitions File for a Dialup Line</b>	73
<b>Appendix C. C Sample Application Listings</b>	75
C.1 C Sample Application Source Code	76
C.2 ADL for Host View of Customer Master Record	102
C.3 ADL for Application View of Customer Master Record	102
C.4 ADL for Host View of Account Details Record	103
C.5 ADL for Application View of Account Details Record	103
C.6 LISTCAT Output for Customer Master File	104

C.7 LISTCAT Output for Account Details File . . . . .	105
C.8 Customer Master File with Data Conversion . . . . .	106
C.9 Account Details File with Data Conversion . . . . .	106
C.10 Customer Master File without Data Conversion . . . . .	107
C.11 Account Details File without Data Conversion . . . . .	107
 <b>Appendix D. PL/I Sample Application Listings</b> . . . . .	 109
D.1 PLILOAD Source Code . . . . .	109
D.2 PLIREAD Source Code . . . . .	110
D.3 PLIMAIN Source Code . . . . .	111
D.4 ADL for Host View of Books Catalog Record . . . . .	113
D.5 ADL for Application View of Books Catalog Record . . . . .	113
D.6 LISTCAT Output of PL/I Data Set after Running Sample . . . . .	114
D.7 PL/I Sample Data Set after Running Sample . . . . .	115
 <b>List of Abbreviations</b> . . . . .	 117
 <b>Index</b> . . . . .	 119



## Figures

1.	Distributed FileManager Client/Server Data Flow	2
2.	DFM/MVS and Other MVS Components	3
3.	Distributed FileManager Working Environment	4
4.	APPC/MVS LU Definition from MVS3	6
5.	PWS LU Definition	6
6.	LOGMODE Entry Source Code	7
7.	Startup Parameters for APPC on MVS3 (APPCPM00)	7
8.	Startup Parameters for ASCH (ASCHPM00)	8
9.	Transaction Program Profile Creation	9
10.	APPC Startup Procedure	9
11.	ASCH Startup Procedure	9
12.	DFM/MVS: Parameters Member in SYS1.PARMLIB	10
13.	DFM/MVS: Startup Procedure	10
14.	Communications Manager Setup Window	11
15.	Open Configuration Window	12
16.	Communications Manager Configuration Definition Window	13
17.	Communications Manager Profile List Sheet Window	14
18.	Local Node Characteristics Window	14
19.	Local Node Options Window	15
20.	Connections List Window	15
21.	Adapter List Window	16
22.	Change a Connection to a Host Window	16
23.	Change Partner LUs Window: No LU Added	17
24.	Change Partner LUs Window: Both LUs Added	18
25.	SNA Features List Window	19
26.	Change a Partner LU Window	19
27.	OS/2 Communications Manager Warning Window	20
28.	SNA Features List Window: Adding the QPCSUPP Logmode Name	20
29.	Create a Mode Definition Window	21
30.	Communications Manager - Checking Values Window	22
31.	CM/2 NDF for Using DFM/2	22
32.	DFM/2 Configuration File (CONFIG.DFM)	25
33.	STRTDFMC and STRTDFMR Commands	27
34.	DFMDRIVE - Drive Control Window: No Drive Assigned	28
35.	Assign a Drive to a System	28
36.	DFMDRIVE - Drive Control Window: Two Drives Assigned	29
37.	DFMSAMP - Data Access Services Sample Program Main Window	32
38.	DFMSAMP - Data Access Services Sample Program File Settings Pull-down	32
39.	FILE_DESCRIPTOR_MAPs Added to CONFIG.DFM File	34
40.	DFMSAMP - Keyed File Settings	35
41.	ISMF Panel with DFMSAMP.ACCT Data Set	36
42.	DFMSAMP - Load Keyed Files	36
43.	DFMSAMP - Retrieve by Customer Number	37
44.	DFMSAMP - Customer Numbers Retrieved	37
45.	DFMSAMP - Main Window with Data	38
46.	FILE_DESCRIPTOR_MAPs Added for the PL/I Sample	40
47.	JCL to Create the VSAM Data Set	41
48.	Execution of the SETDIREC Command	41
49.	Sample Data for the PL/I Application	42
50.	PL/I Sample: Loading Data	42

51.	PL/I Sample: Reading Data . . . . .	43
52.	Sample Update Data for the PL/I Application . . . . .	43
53.	PL/I Sample: Updating Data . . . . .	44
54.	ISMF Panel with PLISAMP.DATA Data Set . . . . .	45
55.	Assign a Drive to an MVS Data Set . . . . .	46
56.	Drives Assigned to MVS Data Sets . . . . .	47
57.	Copy Files to SAM Data Sets and PDSE Members . . . . .	48
58.	DIR Command for the Assigned Drives . . . . .	49
59.	Reading a Stream-File from an MVS Data Set . . . . .	49
60.	ISPF Data Set List of the MVS Data Sets . . . . .	50
61.	Member List of the PDSE . . . . .	50
62.	Contents of the PDSE Member and the SAM Data Set . . . . .	50
63.	AS/400 Host: Display Network Attributes Command Output . . . . .	52
64.	AS/400 Host: Display Line Description Command Output . . . . .	53
65.	AS/400 Host: Display Controller and Device Description Command Output . . . . .	54
66.	AS/400 Host: 5250 Emulation through the Token-Ring Window . . . . .	56
67.	AS/400 Host: 5250 Emulation Window . . . . .	57
68.	AS/400 Host: 5250 Emulation Window with Partner LU Drop-down List . . . . .	57
69.	AS/400 Host: CM/2 NDF with AS/400 Definitions . . . . .	58
70.	AS/400 Host: DFM_TARGET Added for AS/400 . . . . .	60
71.	AS/400 Host: DFMDRIVE - Drive Control Window . . . . .	61
72.	AS/400 Host: DFMSAMP - Keyed File Settings . . . . .	62
73.	AS/400 Host: Work with Files Panel . . . . .	62
74.	AS/400 Host: Display Physical File Member (Character and Hex Format) . . . . .	63
75.	AS/400 Host: DFMSAMP - Retrieve by Customer Number Input Window . . . . .	64
76.	AS/400 Host: Record Retrieved by Customer Number . . . . .	64
77.	AS/400 Host: Get Record by an Alternate Key . . . . .	64
78.	AS/400 Host: Record Retrieved by Customer's Last Name . . . . .	65
79.	AS/400 Host: FILE_DESCRIPTOR_MAP Keywords Added for AS/400 . . . . .	65
80.	AS/400 Host: DFMSAMP - Keyed File Settings (with Data Conversion) . . . . .	66
81.	AS/400 Host: Data Files on the AS/400 System (with Data Conversion ) . . . . .	67
82.	AS/400 Host: DFMSAMP - Local File Settings (Report File) . . . . .	68
83.	AS/400 Host: Stream File in an AS/400 Folder (OS/2 View) . . . . .	68
84.	AS/400 Host: Stream File in an AS/400 Folder (AS/400 View) . . . . .	69
85.	AS/400 Host: Freelance Graphics Using AS/400 Storage (OS/2 View) . . . . .	70
86.	AS/400 Host: Freelance Graphics Using AS/400 Storage (AS/400 View) . . . . .	70

---

## Tables

1. DFMSAMP Action Bar Choices and Pull-down Selections . . . . .	31
2. PL/I Sample Application: REXX Procedures Overview . . . . .	39



---

## Special Notices

This document is written for OS/2 workstation users and application designers who want to know how to use Distributed FileManager to access data sets that reside on a host system (MVS/ESA or OS/400) from an OS/2 workstation. The information in this publication is not intended as the specification of any programming interfaces that are provided by the Distributed FileManager component of DFSMS/MVS Version 1, Release 2, and the DAS component of ADSM. See the PUBLICATIONS section of the IBM Programming Announcement for DFSMS/MVS and ADSM for more information about what publications are considered to be product documentation.

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

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

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

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The 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.

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

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

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

ACF/VTAM  
Application System/400  
CICS/MVS

ADSTAR  
AS/400  
CICS/VSE

DFSMS/MVS	IBM
MVS/ESA	Operating System/2
Operating System/400	OS/2
OS/400	Personal System/2
Presentation Manager	PS/2
RACF	SAA
Systems Application Architecture	VTAM

The following terms are trademarks of other companies:

Freelance Graphics	Lotus Development Corporation.
--------------------	--------------------------------

Other trademarks are trademarks of their respective companies.

---

## Preface

This document describes the functions of Distributed FileManager (DFM) in an OS/2-MVS/ESA environment and explains how to use Distributed FileManager to access record-oriented and stream-oriented data sets. It describes the setup for the OS/2, MVS/ESA, and OS/400 systems and provides sample applications written in high level languages such as C and PL/I.

This document is written for OS/2 workstation users and application designers who want to know how to use Distributed FileManager to access data sets that reside on a host system (MVS/ESA or OS/400) from an OS/2 workstation.

---

## How This Document Is Organized

The document is organized as follows:

- Chapter 1, "Introduction to Distributed FileManager"
- Chapter 2, "DFM/2 and DFM/MVS"

This chapter describes the required setup of Distributed FileManager on OS/2 and MVS/ESA and provides samples for record level I/O and byte-stream I/O applications.

- Chapter 3, "DFM/2 and AS/400 Distributed Data Management"

This chapter describes the required steps for adding an AS/400 target system to the Distributed FileManager environment and provides samples for record level I/O and byte-stream I/O applications.

- Appendix A, "Sample Diskette Description"

This appendix describes the contents of the diskette that accompanies this book.

- Appendix C, "C Sample Application Listings"
- Appendix D, "PL/I Sample Application Listings"

---

## Related Publications

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

- *Distributed Storage Manager Data Access Services CD-ROM*, SK2T-8706
- *ADSTAR Distributed Storage Manager Data Access Services Distributed FileManager Administration and Customization Guide*, SH35-0121 (this publication is part of *Distributed Storage Manager Data Access Services CD-ROM*)
- *ADSTAR Distributed Storage Manager: General Information*, GH35-0114
- *MVS/ESA Planning: APPC Management*, GC28-1503
- *Distributed Data Management Architecture: General Information*, GC21-9527
- *Distributed Data Management Architecture: Specifications for a Data Language*, SC21-8286

- *Distributed Data Management Architecture: Implementation Programmer's Guide*, SC21-9529
- *Distributed Data Management Architecture: Reference*, SC21-9526
- *DFSMS/MVS Version 1 Release 2 DFSMSdpf Storage Administration Reference*, SC26-4920
- *DFSMS/MVS Version 1 Release 2 Distributed FileManager/MVS Guide and Reference*, SC26-4915
- *VTAM Resource Definition Reference*, SC31-6438

---

## International Technical Support Organization Publications

- *DFSMS/MVS Version 1 Release 2.0 Distributed FileManager/MVS*, GG24-4403

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 ITSO technical bulletins (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. IBM employees and customers may also order redbooks in online format on CD-ROM collections, which contain the redbooks for multiple products.



---

## Acknowledgments

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

The project was designed and managed by:

- Joshua Peleg - ITSO, San Jose Center.

We want to acknowledge the excellent work of the participating resident:

- Torbjörn Kaving, IBM Sweden.

We also wish to acknowledge the following contributors and reviewers whose valuable assistance is highly appreciated:

- Andi Bitterer, ITSO - San Jose Center
- Maggie Cutler, Technical Editor
- Mansoor Dawoodbhoy, IBM Storage Systems Division - San Jose
- Randy Larson, US Marketing and Services - Carolina's Trading Area
- Valerie Poggi, IBM Storage Systems Division - San Jose
- Seppo Rinne, ITSO - San Jose Center
- Mike Roney, IBM Storage Systems Division - San Jose
- Srinu Sista, IBM Storage Systems Division - San Jose.



---

## Chapter 1. Introduction to Distributed FileManager

When developing applications in a distributed environment, have you ever wanted to run your Operating System/2 (OS/2) programs using the existing host-based data sets, without having to write additional code on the host side, translate data and record formats, or transfer the data sets to your workstation? Now, by using the Distributed FileManager (DFM) set of products, you can access and manipulate existing host data sets. If the data sets are under the control of Multiple Virtual Storage/Enterprise Systems Architecture (MVS/ESA), Operating System/400 (OS/400), or any other Systems Application Architecture (SAA) operating system that supports Distributed FileManager, all you need to do is write your application, define the logical connection to your host system, and define the data translation that you need. Then let Distributed FileManager do the rest.

---

### 1.1 What Is Distributed FileManager?

Distributed FileManager as a client server concept is implemented as a set of products that act as the interface between your applications and the file data, irrespective of where the data resides. Hence this implies that the data may be local to the application or remote to the application on either a local area network (LAN) server or a mainframe server on a wide area network. This product set comprises client products and mainframe-based products.

The Distributed FileManager product set is based on the Distributed Data Management (DDM) Architecture that has introduced the terms *source* and *target* for data requests. These terms are equivalent to the more modern terms, *client* and *server*, respectively. One of the primary ingredients of the Distributed FileManager suite of products is Data Access Services (DAS) for both record-oriented and stream-oriented data on local systems and remote servers.

Distributed FileManager uses Advanced Program-to-Program Communication (APPC) to establish a connection between the source system and the target system.

At the current level of implementation source systems can be:

- OS/2
- OS/400.

Target systems can be:

- OS/400
- MVS/ESA with Data Facility Storage Management Subsystem/MVS (DFSMS/MVS) Version 1 Release 2
- 4680 Store Systems
- CICS/MVS or CICS/VSE.

With Distributed FileManager you can also invoke commands to create, delete, rename, and copy data sets on a remote system.

## 1.2 Distributed FileManager Components and Data Flow

To better understand the Distributed FileManager concept, let's take a closer look at DAS and Distributed FileManager/MVS (DFM/MVS) and the data and control flow between the two products. DAS consists of two components—VSAM/2 (previously called RLIO) and Distributed FileManager/2 (DFM/2)—and is an OS/2-based 32-bit product (see Figure 1).

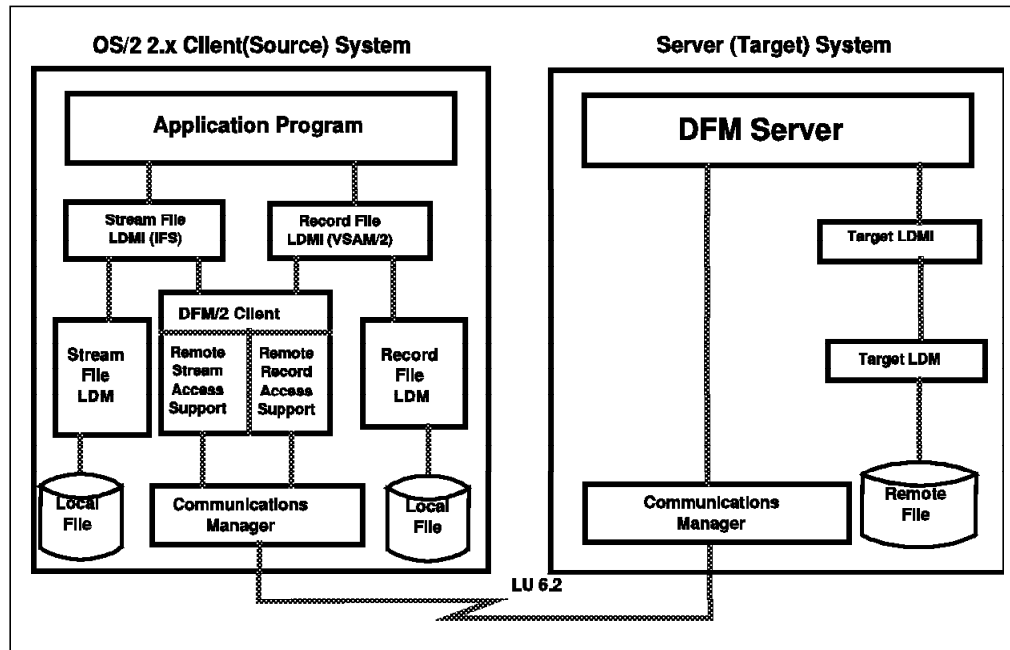


Figure 1. Distributed FileManager Client/Server Data Flow

DFM/MVS is the server in the client/server relationship. The DFM/MVS software support is required to receive a DDM request from a source system and convert the DDM request into a local data management request.

Figure 2 on page 3 shows in greater detail the interaction between DFM/MVS and other MVS components.

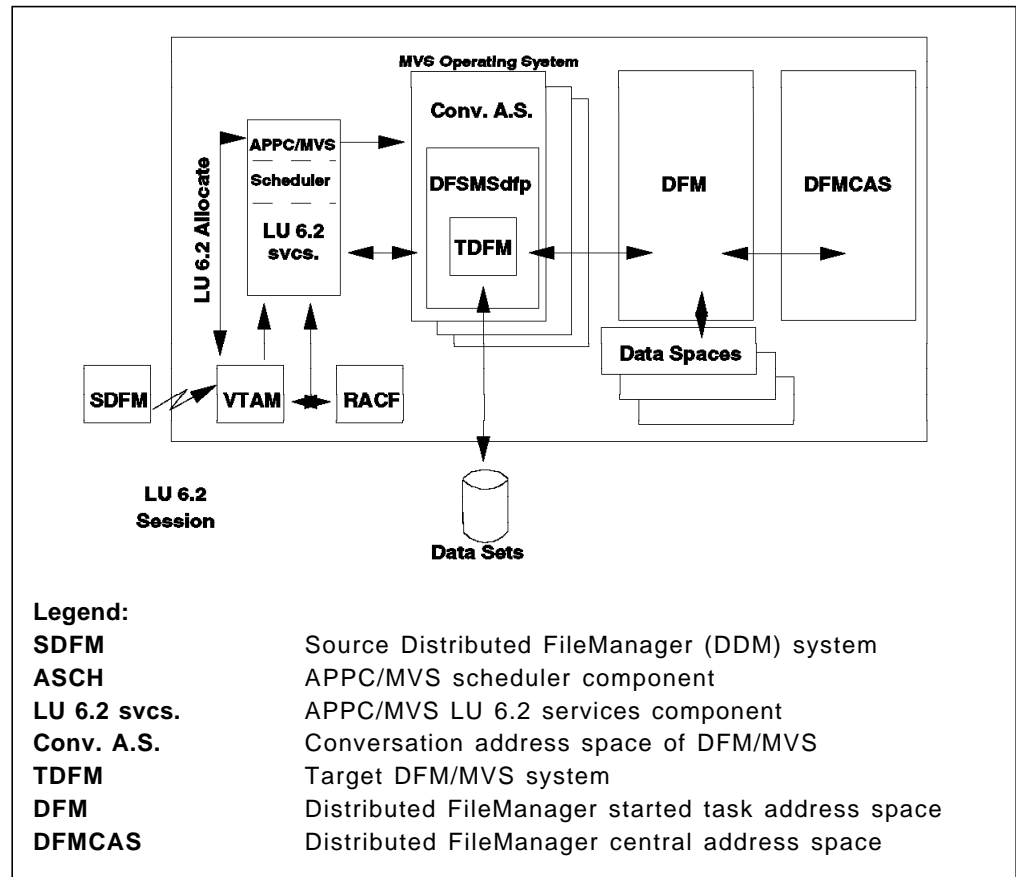


Figure 2. DFM/MVS and Other MVS Components

You start DFM/MVS by using the START DFM command. The DFM started task procedure automatically generates the DFMCAS address space.

If DFM/MVS has been properly set to interact with APPC and Resource Access Control Facility (RACF), conversation address spaces are started when a source DDM or Distributed FileManager system initiates a request to use the target.

Whenever a conversation is established, APPC uses the ASCH component to schedule a new conversation address space. RACF checking is performed to authenticate the user.

Advanced Communications Function/Virtual Telecommunications Access Method (ACF/VTAM) handles the physical communications to the source DDM or Distributed FileManager system.

DFM/2 is the client in the client/server relationship. DFM/2 is the interface between application programs running under OS/2 and the various file systems.

Application programs can use both stream and record level access to data. The OS/2 environment has two local data management interfaces:

- Stream-file-installable file system

This component is automatically active when DFM/2 is started.

- Record file

This local data management interface (LDMI) is referred to as the VSAM/2 component of DFM/2.

The VSAM/2 application program interface (API) support is provided in an OS/2 dynamic link library (DLL) and is available to all VSAM/2 applications when the path to the DLL is in the LIBPATH statement of the OS/2 CONFIG.SYS file. The DLL provides record access to local data. If you want to access remote record data, you must first start the DFM/2 Remote Record Access Support.

Once each component is active, file requests are routed to either the locally attached devices or the target Distributed FileManager system through the OS/2 Communications Manager (see Figure 1 on page 2).

### 1.3 Distributed FileManager Working Environment

In this publication we discuss the implementation of DAS on an OS/2 2.1 source system with DFSMS/MVS 1.2 and OS/400 2.3 as the target systems. Figure 3 shows the environment.

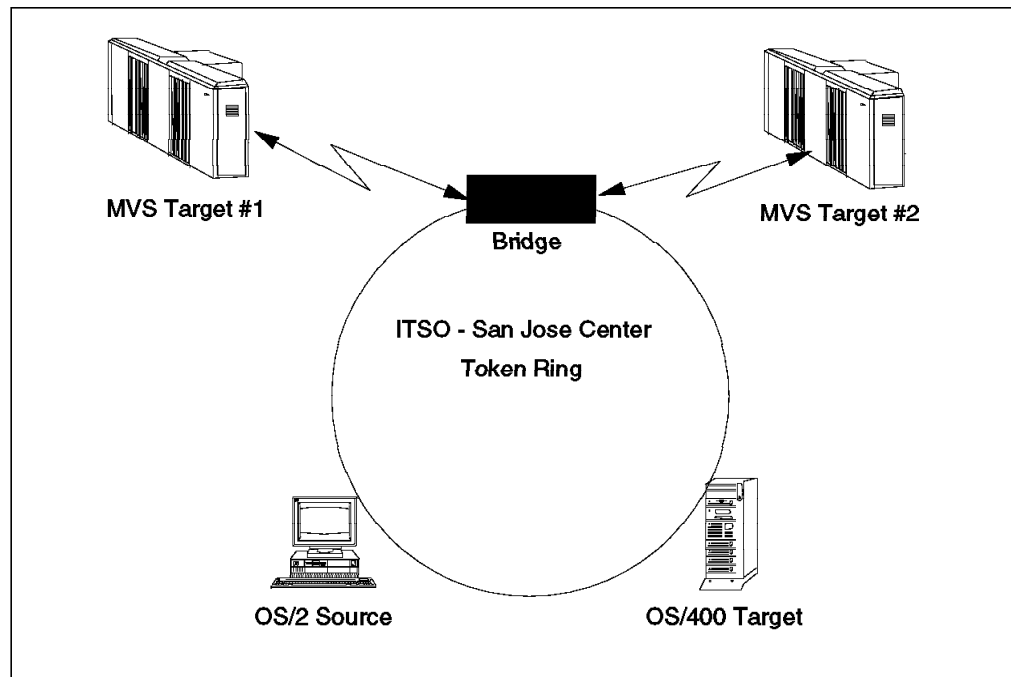


Figure 3. Distributed FileManager Working Environment

**Note !**

Before you start using DFM/MVS and DAS, we suggest you check with your support center to see whether there are any new enhancements that you can apply to your system.

---

## Chapter 2. DFM/2 and DFM/MVS

In this chapter we describe the environment where we implemented the sample applications. We also explain how to set up Distributed FileManager communication between an OS/2 workstation and one or more MVS systems and describe the record level I/O and stream I/O sample applications.

We used a Personal System/2 (PS/2) model 80 with OS/2 as our programmable workstation (PWS). The PWS was connected to a token ring local area network (TRLAN).

We implemented and ran the samples in the PWS. The samples accessed data in two host systems using the MVS/ESA operating system.

DFM/MVS is part of DFSMS/MVS V1.2.0, and DFM/2 and VSAM/2 are part of DAS. We assume that you have installed DFSMS/MVS V1.2.0 in your MVS system and DAS in your PWS. DAS is shipped as part of ADSTAR Distributed Storage Manager (ADSM) when you order an OS/2 client, or you will receive it when you order *ADSTAR Distributed Storage Manager Data Access Services: RLIO and Distributed FileManager Installation Guide*, SK2T-8706 (CD-ROM).

---

### 2.1 MVS/ESA Environment

The two MVS/ESA systems that we used were MVS1 and MVS3. The environment and definitions in the two systems were identical with a few exceptions. We point out all differences in the descriptions.

In the MVS/ESA systems we used the following program product versions:

- MVS/ESA SP JES2 Version 5.1.0
- DFSMS/MVS Version 1.2.0 + APAR OW07011
- ACF/VTAM Version 3.4.1.

In the host system you need definitions for ACF/VTAM, Advanced Program-to-Program Communication/Multiple Virtual Storage (APPC/MVS), and DFM/MVS.

#### 2.1.1 ACF/VTAM Definitions

Communication between DFM/2 and DFM/MVS is set up as a logical unit (LU)-to-LU session using the LU 6.2 protocol. The host LU is used by APPC/MVS, and you define it to ACF/VTAM in SYS1.VTAMLST as an application LU. Figure 4 on page 6 shows the application LU definition from MVS3.

```

*** APPLAPPC - VTAM LU DEFINITIONS FOR APPC/MVS          ****
APPLAPPC VBUILD TYPE=APPL
MVS3APPC APPL  APPC=YES,                                  *
                AUTOSSES=0,                              *
                DDRAINL=NALLOW,                          *
                DRESPL=NALLOW,                          *
                DSESLIM=32,                              *
                EAS=32,                                    *
                MODETAB=WRLMODE,                          *
                SECACPT=CONV,                             *
                SRBEXIT=YES,                              *
                VERIFY=NONE

```

Figure 4. APPC/MVS LU Definition from MVS3

The MVS1 system differs from the MVS3 system in its name (MVS1APPC). The rest of the definitions are identical in the two systems.

#### DSESLIM Parameter

The default value for the number of sessions is DSESLIM=2. This can be a serious limitation because each assigned drive and each application require a unique session. Make sure that you specify a higher value for the DSESLIM parameter.

You define the PWS as an independent LU in order to use APPC. We defined the PWS as a switched major node with a physical unit (PU) and LUs because it was connected through a TRLAN. DFM/2 used the independent LU SNC4879I. See Figure 5.

You must activate the application LU before you start APPC/MVS, and you must activate the PWS LU before you can start any Distributed FileManager communication.

```

SNC4879  PU      ADDR=01,                                *
                IDBLK=05D,IDNUM=C4879,                  *
                ANS=CONT,DISCNT=NO,                     *
                IRETRY=NO,ISTATUS=ACTIVE,                *
                MAXDATA=265,MAXOUT=1,                    *
                MAXPATH=1,                               *
                PUTYPE=2,SECNET=NO,                      *
                MODETAB=WRLMODE,DLOGMOD=D1ADYNA,        *
                USSTAB=USSTABEN,                        *
                PACING=1,VPACING=2
*
SNC4879A LU      LOCADDR=002,LOGAPPL=SAMONEN
SNC4879B LU      LOCADDR=003,LOGAPPL=SAMONEN
SNC4879C LU      LOCADDR=004,LOGAPPL=SAMONEN
SNC4879D LU      LOCADDR=005,LOGAPPL=SAMONEN
*
SNC4879I LU      LOCADDR=0,DLOGMOD=LU62APPB
SNC4879J LU      LOCADDR=0,DLOGMOD=LU62APPB

```

Figure 5. PWS LU Definition



DFM/2 requires a VTAM logmode entry with a logmode name of QPCSUPP. Figure 6 on page 7 shows the source code for the logmode entry. You must add this to a logmode table, and you must assemble the logmode table and link-edit it to VTAMLIB. Enter the name of the logmode table in the MODETAB parameter of the APPL statement for APPC/MVS. We used MODETAB=WRLMODE. See Figure 4 on page 6.

```

                TITLE 'QPCSUPP'
*****
*          LOGMODE TABLE ENTRY FOR RESOURCES CAPABLE OF ACTING          *
*              AS LU 6.2 DEVICES                                          *
*              REQUIRED FOR LU MANAGEMENT                                  *
*****
QPCSUPP  MODEENT LOGMODE=QPCSUPP,FMPROF=X'13',TSPROF=X'07',
                PRIPROT=X'B0',SECPROT=X'B0',COMPROT=X'D0B1',
                RUSIZES=X'8585',ENCR=B'0000',
                PSERVIC=X'060200000000000000000000300'

```

Figure 6. LOGMODE Entry Source Code

## 2.1.2 APPC/MVS Definitions

APPC/MVS is a standard component of MVS/ESA. DFM/MVS is just one of the possible users of APPC/MVS. Thus it is possible that you already have APPC/MVS defined and running. If this is the case, you just have to find the LU name that APPC/MVS uses. See Figure 4 on page 6 and Figure 7. You will need this LU name for your Communications Manager/2 (CM/2) definitions. See Figure 24 on page 18.

APPC/MVS consists of two address spaces: the APPC address space and the APPC transaction scheduler (ASCH) address space.

APPC/MVS requires startup parameters from PARMLIB members APPCPMxx for APPC, and ASCHPMxx for ASCH. Figure 7 shows the PARMLIB member for APPC from the MVS3 system.

```

LUADD
  ACBNAME(MVS3APPC)
  BASE
  TPDATA(SYS1.APPCTP)
  TPLEVEL(USER)

SIDEINFO
  DATASET(SYS1.APPCSI)

```

Figure 7. Startup Parameters for APPC on MVS3 (APPCPM00)

We omitted the SCHED parameter and let it take the default value, SCHED(ASCH).

The APPCPM00 member from the MVS1 system specifies ACBNAME(MVS1APPC). The rest of the members are identical in the two systems.

Note that in each system the LUADD ACBNAME parameter and the APPL statement defined to ACF/VTAM must specify the same name. In our systems the names were:

```
MVS1:      MVS1APPC
MVS3:      MVS3APPC
```

If you are already using APPC/MVS, find the LUADD statement with BASE and SCHED(ASCH) (or without the SCHED parameter) and note the name in the ACBNAME parameter. DFM/MVS uses the base LU for the APPC/MVS transaction scheduler.

Figure 8 shows the PARMLIB member for ASCH.

```
CLASSADD
  CLASSNAME(DEFAULT)
  MAX(20)
  MIN(1)
  MSGLIMIT(5000)

OPTIONS
  DEFAULT(DEFAULT)

TPDEFAULT
  REGION(2M)
  TIME(5)
  MSGLEVEL(1,1)
  OUTCLASS(9)
```

*Figure 8. Startup Parameters for ASCH (ASCHPM00)*

APPC/MVS requires two Virtual Storage Access Method (VSAM) data sets: a transaction program (TP) profile data set and a side information data set. For information on how to create these data sets, see *MVS/ESA Planning: APPC Management*.

You must create a TP profile for DFM/MVS in the APPC/MVS TP profile data set. Figure 9 on page 9 shows the job step used to create the TP profile for DFM/MVS.

```

//STEP      EXEC PGM=ATBSDFMU
//SYSPRINT DD  SYSOUT=*
//SYSSDOUT DD  SYSOUT=*
//SYSSDLIB DD  DSN=SYS1.APPCTP,DISP=SHR
//SYSIN      DD  DATA,DLM=XX
      TPADD
        TPNAME(-X'07'001)
        ACTIVE(YES)
        TPSCHED_DELIMITER(##)
        CLASS(DEFAULT)
        JCL_DELIMITER(ENDJCL)
//GDEDFM JOB MSGCLASS=H,MSGLEVEL=(1,1),CLASS=A
//GDEDFM EXEC PGM=GDEISASB
ENDJCL
##
XX

```

Figure 9. Transaction Program Profile Creation

APPC/MVS requires two startup procedures: APPC and ASCH. These procedures should be available to the MVS START command. Figure 10 shows the APPC procedure, and Figure 11, the ASCH procedure.

```

//APPC      PROC APPC=00
//APPC      EXEC PGM=ATBINITM,PARM=' APPC=&APPC' , REGION=64M

```

Figure 10. APPC Startup Procedure

```

//ASCH      PROC ASCH=00
//ASCH      EXEC PGM=ASBSCHIN,PARM=' ASCH=&ASCH' , REGION=64M

```

Figure 11. ASCH Startup Procedure

Before you start APPC/MVS make sure that you have activated the APPC/MVS application LU in ACF/VTAM. Use the following commands to start APPC/MVS:

```

START APPC,SUB=MSTR
START ASCH,SUB=MSTR

```

These commands start the APPC and ASCH address spaces.

Additional information on customizing APPC/MVS can be found in *MVS/ESA Planning: APPC Management*.

### 2.1.3 DFM/MVS Definitions

Before you can start DFM/MVS you must update the SYS1.PARMLIB member DFM00, and verify that the PROCLIB member DFM exists. Figure 12 on page 10 shows the DFM00 parameters member, and Figure 13 on page 10, the DFM procedure.

```

DFM LOCK_WAIT_INTV(20)
  MAX_CONV_LOCK(5)
  LOCK_RETRY(3)
  CCSID(0)
  CLOSE_CHECK_INTV(0)
  DEFER_CLOSE_TIME(0)
  MAX_AGENT_TSKS(5)
  STREAM_LRECL(8196)
  SEND_BUFFER_THRESHOLD(100)

```

Figure 12. DFM/MVS: Parameters Member in SYS1.PARMLIB

```

//DFM      PROC PARMS=NORMAL
//*****
//*        ADSTAR DISTRIBUTED FILE MANAGER ADDRESS SPACE      *
//*****
//          EXEC PGM=GDEISBOT,PARM='&PARMS',REGION=32M,TIME=1440
//IEFPARM DD DSN=SYS1.PARMLIB,DISP=SHR
//SYSPRINT DD SYSOUT=H

```

Figure 13. DFM/MVS: Startup Procedure

Before you start DFM/MVS make sure that the APPC/MVS and PWS LUs are activated in ACF/VTAM and that APPC/MVS is started.

The following command starts DFM/MVS:

```
START DFM,SUB=MSTR
```

Two address spaces are created, DFM and DFMCAS.

#### 2.1.4 DFSMS/MVS ACS Routines Definition

The two MVS systems that we used had Storage Management Subsystem (SMS) active, and every new allocation was for an SMS-managed data set as default.

You must activate SMS in your MVS system and make sure that your storage class automatic class selection (ACS) routine forces every data set created by DFM/MVS to be SMS-managed.

---

## 2.2 OS/2 Environment

We used the following program product versions in the PWS:

- OS/2 Version 2.1
- Communications Manager/2 Version 1.01
- Data Access Services.

In the PWS you need definitions for CM/2 and DFM/2.

## 2.2.1 Communications Manager/2 Definitions

In our environment we used CM/2 1.01 and our workstation was connected to the network through a token-ring LAN. If you have a different level of CM/2 some of the windows will not match the ones we are showing in this section. You can also use any other type of connections that is supported for APPC communication (see Appendix B, "Sample Network Definitions File for a Dialup Line" on page 73).

We set the definitions for CM/2 by using Communications Manager Setup. The description that follows guides you through the Communications Manager Setup windows to set or verify the APPC parameters needed for Distributed FileManager. We assume that your PWS uses a connection that supports APPC.

On your OS/2 desktop select Communications Manager/2, and from the next window select Communications Manager Setup, or you can enter the CMSETUP command from an OS/2 window. The CMSETUP window with an IBM logo appears. Select **OK** to get to the Communications Manager Setup window shown in Figure 14.



Figure 14. Communications Manager Setup Window

Select **Setup...** to get to the Open Configuration window shown in Figure 15 on page 12.

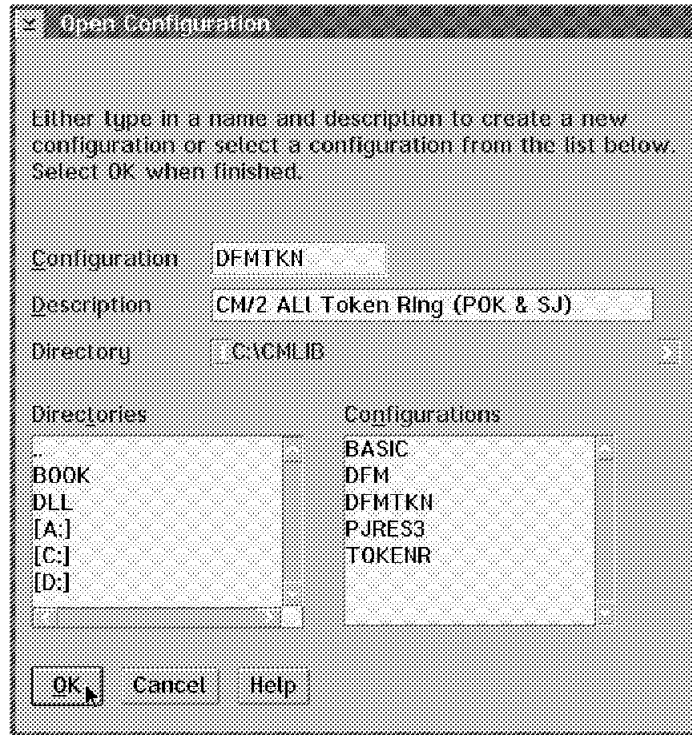


Figure 15. Open Configuration Window

This window shows your current configuration and the directory in which it is stored, in our example, DFMTKN and C:\CMLIB, respectively. Now go to an OS/2 window and make a backup copy of your configuration files by using the following commands:

```
C: 1
CD CMLIB 1
DIR config.* 2
COPY config.* configB.* 3
```

**Notes:**

**1** Specify the drive ID and directory that the Open Configuration window displays, usually C: and CMLIB, respectively.

**2** Specify the name of your configuration for config. You should get a list of four files:

- config.CFG
- config.CF2
- config.NDF
- config.SEC.

**3** Make a backup copy of each file. If you experience any problem, you can always come back here and restore the files.

After you have made the backup copies, continue by selecting **OK** on the Open Configuration window. This selection takes you to the Communications Manager Configuration Definition window shown in Figure 16 on page 13.

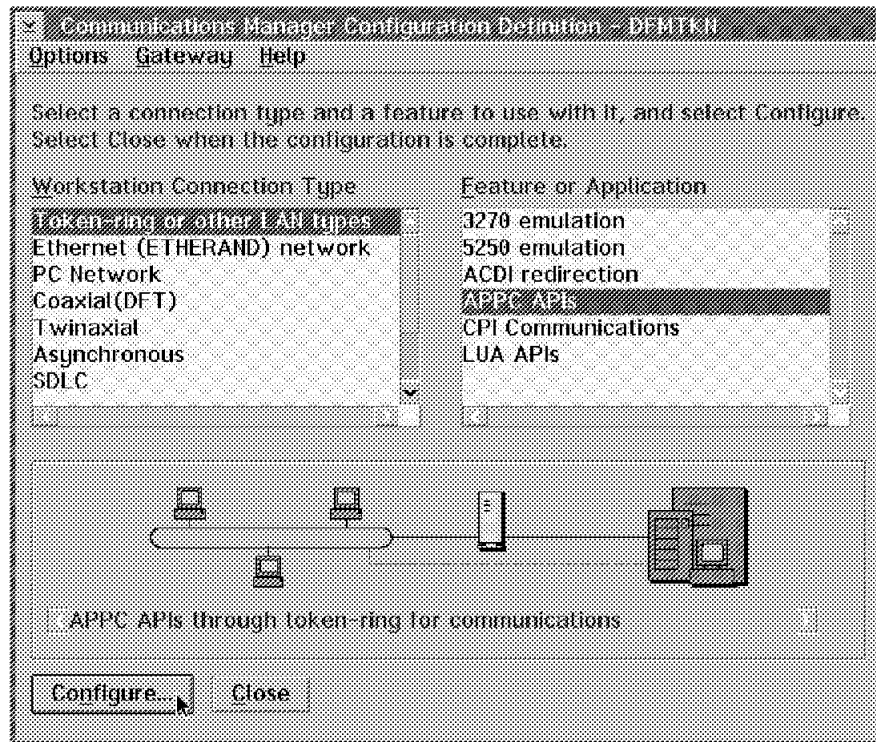


Figure 16. Communications Manager Configuration Definition Window

Select the Workstation Connection Type you are using, select APPC APIs, and then select **Configure....** The Communications Manager Profile List Sheet as shown in Figure 17 on page 14 appears.

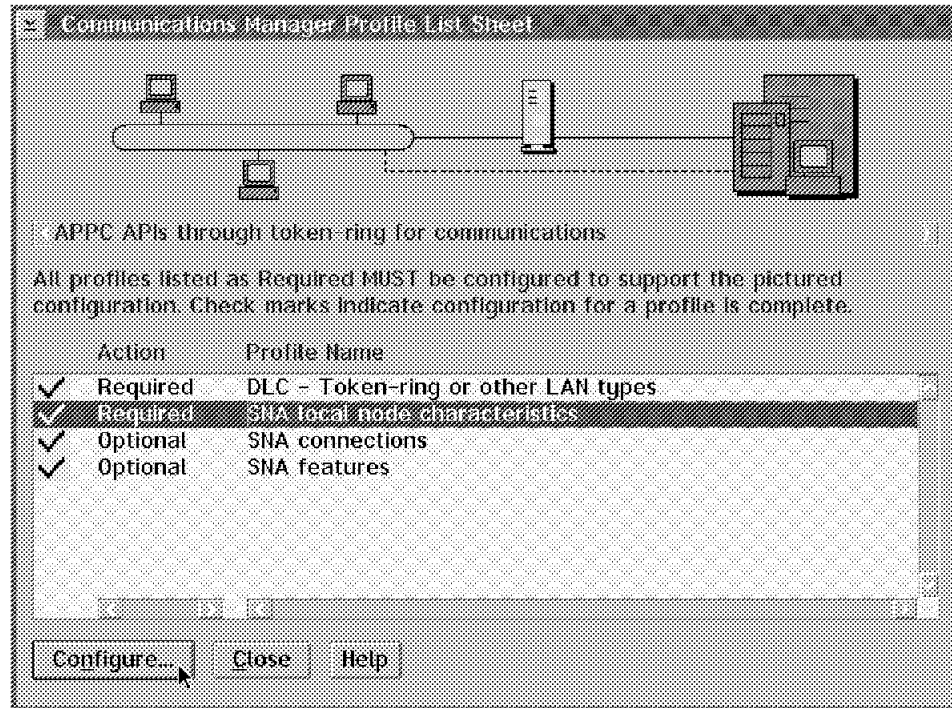


Figure 17. Communications Manager Profile List Sheet Window

You have to update or verify:

- Systems Network Architecture (SNA) local node characteristics
- SNA connections
- SNA features.

Select SNA local node characteristics and **Configure...**, and you get the Local Node Characteristics Window shown in Figure 18.

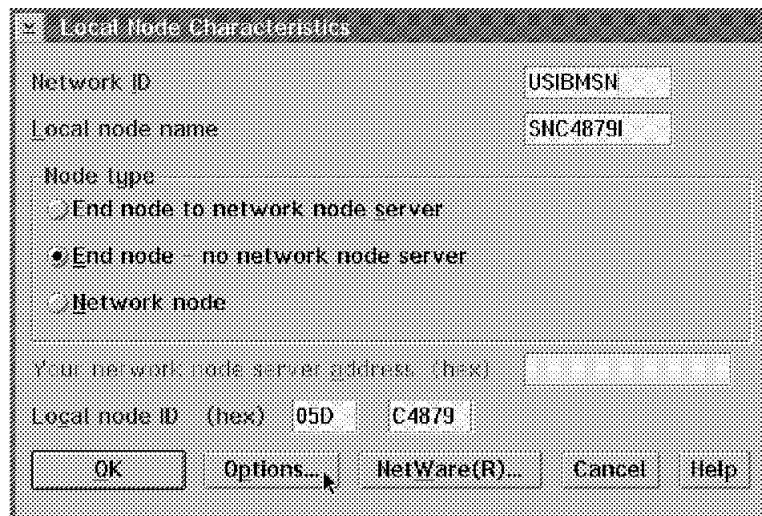


Figure 18. Local Node Characteristics Window



Select **Options**, and you get the Local Node Options window shown in Figure 19.

You have to decide on the alias name you want to use for your workstation LU. You specify this alias name to DFM/2. Write your alias in the Local node alias name field, replacing the previous contents of the field. Figure 19 shows the alias we used, DFMRES. (If you use DFMRES, you have less customization to do to run the samples.)



Figure 19. Local Node Options Window

After you have entered your alias, select **OK**, and you will return to the Local Node Characteristics window. Selecting **OK** again takes you back to the Communications Manager Profile List Sheet window. Now select SNA connections and **Configure...**, and you get the Connections List window. See Figure 20.

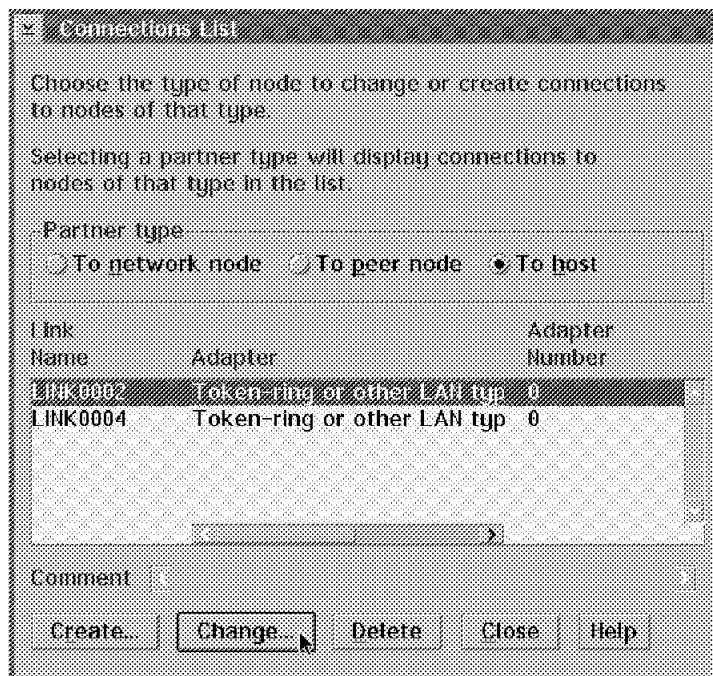


Figure 20. Connections List Window

Because we were defining a connection from an OS/2 workstation to an MVS/ESA host, we selected the **To host** radio button. In our PWS we had two connections to host systems, but we used only the first connection, LINK0002, for

APPC. If you have more than one connection, you probably have to consult a network specialist at your installation to find out which connection to use.

Once you have decided on the connection to use, select that connection and then select **Change....** On the Adapter List window (see Figure 21) select the adapter type you are using and **Continue...**, and you get the Change a Connection to a Host window shown in Figure 22.

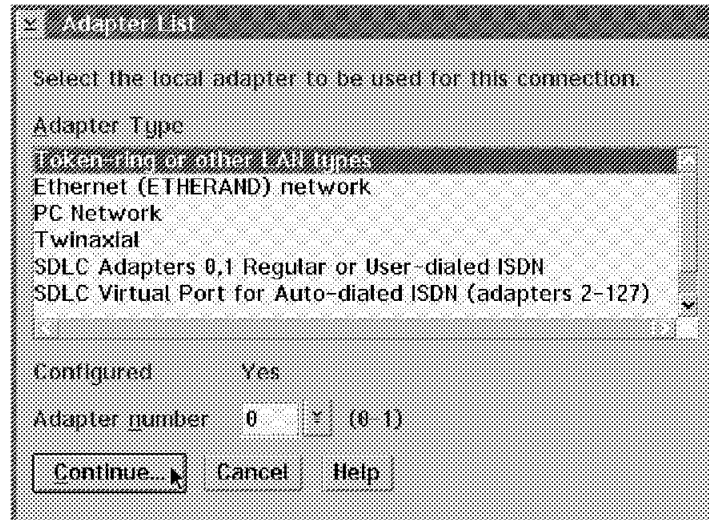


Figure 21. Adapter List Window

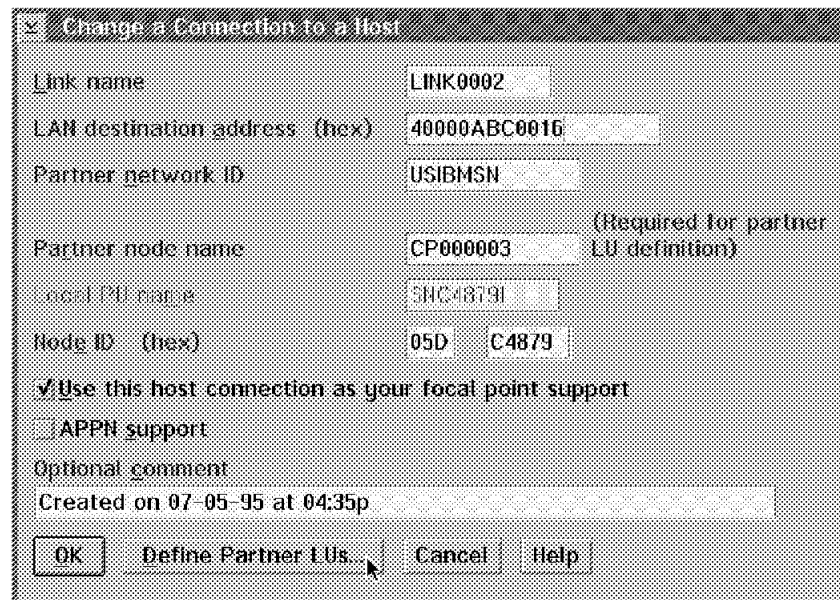


Figure 22. Change a Connection to a Host Window

If the Partner network ID and Partner node name fields are not filled in, you need to get the ID and name from your network specialist. You must know the

network ID to which your workstation is connected, but for a partner node name you can use a dummy name like CP000003 as we used in our example.

When you have the partner network ID and partner node name, enter them in the fields, and then select **Define Partner LUs....** This takes you to the Change Partner LUs window shown in Figure 23. Fill in your APPC/MVS LU name defined in 2.1.2, “APPC/MVS Definitions” on page 7 in the LU name field. Decide on the alias name you want, enter it in the Alias field, and then select **Add**. (If you use one of our aliases (MVS1DFM or MVS3DFM), you have less customization to do to run the samples.)

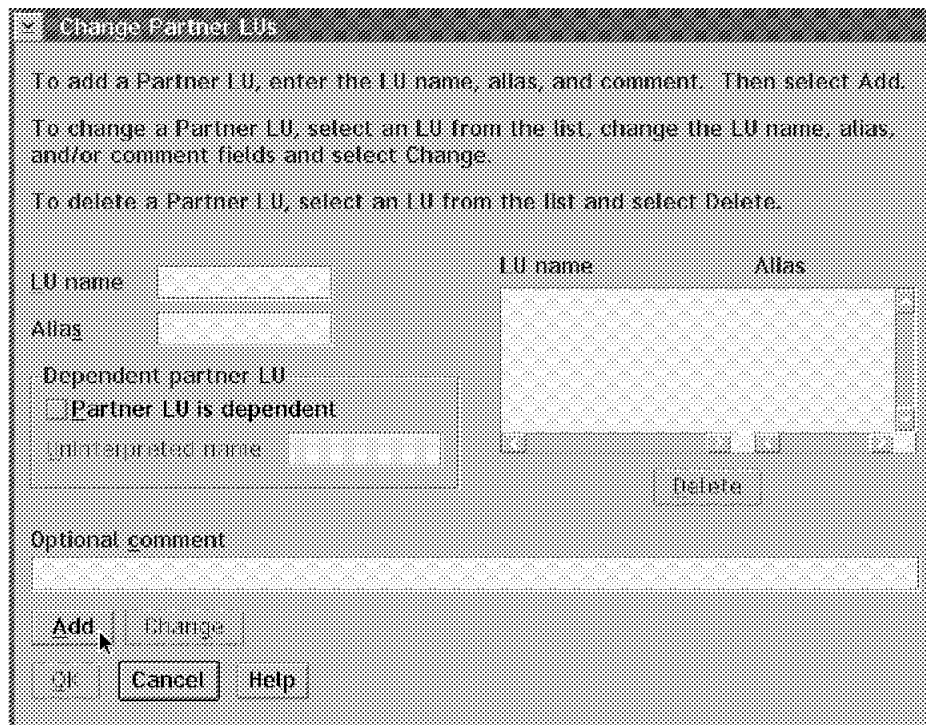


Figure 23. Change Partner LUs Window: No LU Added

Because we used two MVS systems, we had to define two LU names. These names are the same as those we defined to VTAM and APPC/MVS in 2.1.1, “ACF/VTAM Definitions” on page 5 and 2.1.2, “APPC/MVS Definitions” on page 7.

Figure 24 on page 18 shows the Change Partner LUs window after we added both LU names.

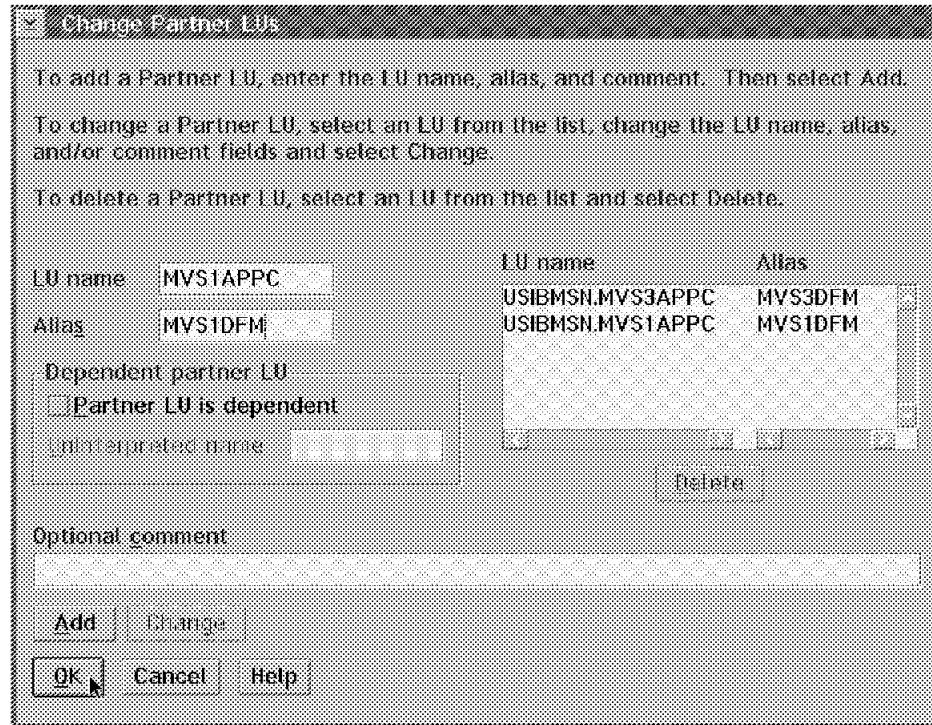


Figure 24. Change Partner LUs Window: Both LUs Added

Selecting **OK** takes you back to the Change a Connection to a Host window. Select **OK** again, and you return to the Connections List window. Now select **Close**, and you come back to the Communications Manager Profile List Sheet window.

In our case the PWS and the MVS systems were in different VTAM networks. Therefore we had to change the network IDs. We did this through the SNA features selection on the Communications Manager Profile List Sheet window.

Selecting SNA features and **Configure...** takes you to the SNA Features List window. After we selected Partner LUs, the two LU aliases appeared in the window as shown in Figure 25 on page 19.

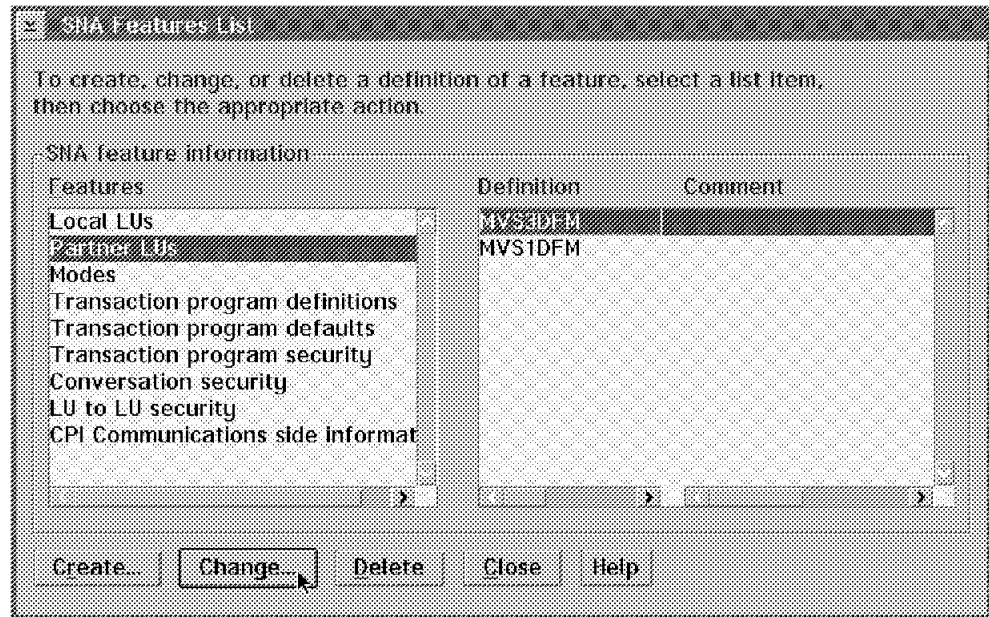


Figure 25. SNA Features List Window

Here we selected MVS3DFM and then **Change...** and got the Change a Partner LU window shown in Figure 26.



Figure 26. Change a Partner LU Window

The Fully qualified LU name field shows the network ID (USIBMSN) for the PWS network. We had to change this to the network ID (USIBMSJ) for the MVS3 network. When we selected **OK** we got the warning shown in Figure 27 on page 20.

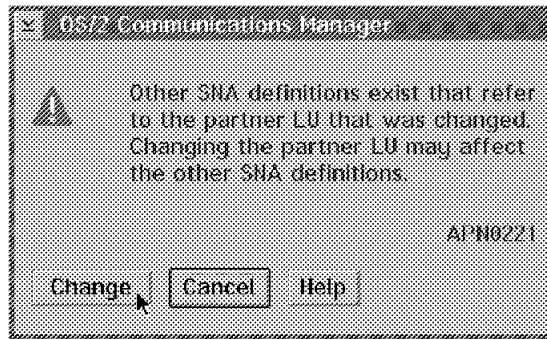


Figure 27. OS/2 Communications Manager Warning Window

We selected **Change**, and the required changes were made.

The network ID for MVS1 is USIBMST. We had to repeat the operation for the other LU and change the network ID for MVS1DFM in the same way.

You must define the logmode name, QPCSUPP, to CM/2. (We defined this name to VTAM in 2.1.1, "ACF/VTAM Definitions" on page 5.) On the SNA Features List window (Figure 25 on page 19) select Modes, and the window changes to look like that in Figure 28.

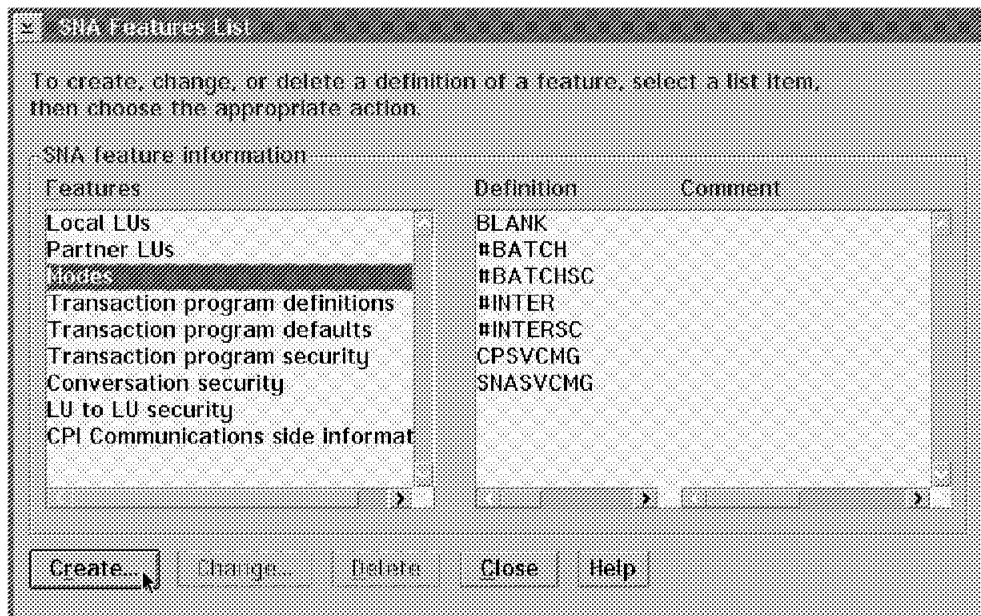


Figure 28. SNA Features List Window: Adding the QPCSUPP Logmode Name

Select **Create...**, and the Change a Mode Definition window appears. Fill in the fields as shown in Figure 29 on page 21.

**Notes:**

The maximum request unit (RU) size can have a significant impact on performance. The optimum RU size is the maximum value that the hardware buffer of the communications adapter can accommodate. Some examples are:

- 1500 bytes for an Ethernet adapter
- 1920 bytes for an IBM Token-Ring Adapter or Token-Ring Adapter/A
- 15360 bytes for a Token-Ring 16/4 Adapter.

The RU size must match the RUSIZES parameter in the logmode entry defined to VTAM as described in 2.1.1, “ACF/VTAM Definitions” on page 5. Check with your network specialist on how to interpret the RUSIZES parameter.

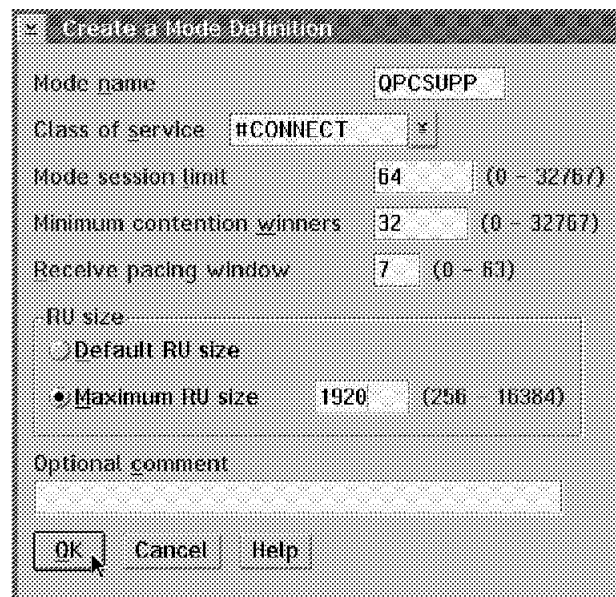


Figure 29. Create a Mode Definition Window

After you select **OK**, you return to the SNA Features List window. Note that QPCSUPP has been added to the list of names under Definition.

You are now done with Communications Manager Setup, and you just go back through the windows selecting **Close** on each window until you see the Communications Manager - Checking Values window shown in Figure 30 on page 22.

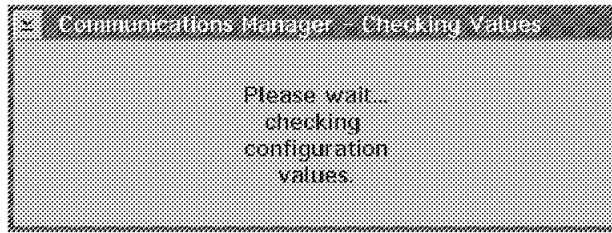


Figure 30. Communications Manager - Checking Values Window

After a moment a window appears asking if you want to dynamically update your SNA resources. If you select **No**, you must stop and restart communications to test your new configuration. If you select **Yes**, CM/2 tries to activate the new configuration immediately.

Now go to an OS/2 window and enter the

DIR config.\*

command again, and you will see that the four files have been changed. They have new time stamps.

The config.NDF file is an American National Standard Code for Information Interchange (ASCII) text file containing all of the definitions made in Communications Manager Setup. You can display the file with an editor or with the TYPE command. Figure 31 shows the network definition file (NDF) file from our workstation.

```

DEFINE_LOCAL_CP  FQ_CP_NAME(USIBMSN.SNC4879I ) 1
                  CP_ALIAS(DFMRES )
                  NAU_ADDRESS(INDEPENDENT_LU)
                  NODE_TYPE(EN)
                  NODE_ID(X'05DC4879')
                  HOST_FP_SUPPORT(YES)
                  HOST_FP_LINK_NAME(LINK0002);

DEFINE_LOGICAL_LINK LINK_NAME(LINK0002) 2
                    FQ_ADJACENT_CP_NAME(USIBMSN.CP000003 )
                    ADJACENT_NODE_TYPE(LEN)
                    DLC_NAME(IBMTRNET)
                    ADAPTER_NUMBER(0)
                    DESTINATION_ADDRESS(X'40000ABC0016')
                    CP_CP_SESSION_SUPPORT(NO)
                    ACTIVATE_AT_STARTUP(YES)
                    LIMITED_RESOURCE(USE_ADAPTER_DEFINITION)
                    LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
                    SOLICIT_SSCP_SESSION(YES)
                    NODE_ID(X'05DC4879')

```

Figure 31 (Part 1 of 3). CM/2 NDF for Using DFM/2



```

EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
COST_PER_BYTE(USE_ADAPTER_DEFINITION)
SECURITY(USE_ADAPTER_DEFINITION)
PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
USER_DEFINED_1(USE_ADAPTER_DEFINITION)
USER_DEFINED_2(USE_ADAPTER_DEFINITION)
USER_DEFINED_3(USE_ADAPTER_DEFINITION);

DEFINE_LOGICAL_LINK LINK_NAME(LINK0004)
                    ADJACENT_NODE_TYPE(LEN)
                    DLC_NAME(IBMTRNET)
                    ADAPTER_NUMBER(0)
                    DESTINATION_ADDRESS('40000ABC0200')
                    CP_SESSION_SUPPORT(NO)
                    ACTIVATE_AT_STARTUP(YES)
                    LIMITED_RESOURCE(USE_ADAPTER_DEFINITION)
                    LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
                    SOLICIT_SSCP_SESSION(YES)
                    PU_NAME(SJA2951 )
                    NODE_ID('05DA2051')
                    EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
                    COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
                    COST_PER_BYTE(USE_ADAPTER_DEFINITION)
                    SECURITY(USE_ADAPTER_DEFINITION)
                    PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
                    USER_DEFINED_1(USE_ADAPTER_DEFINITION)
                    USER_DEFINED_2(USE_ADAPTER_DEFINITION)
                    USER_DEFINED_3(USE_ADAPTER_DEFINITION);

DEFINE_PARTNER_LU FQ_PARTNER_LU_NAME(USIBMSJ.MVS3APPC ) 3
PARTNER_LU_ALIAS(MVS3DFM)
PARTNER_LU_UNINTERPRETED_NAME(MVS3APPC)
MAX_MC_LL_SEND_SIZE(32767)
CONV_SECURITY_VERIFICATION(NO)
PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU FQ_PARTNER_LU_NAME(USIBMSN.MVS1APPC ) 3
PARTNER_LU_ALIAS(MVS1DFM)
PARTNER_LU_UNINTERPRETED_NAME(MVS1APPC)
MAX_MC_LL_SEND_SIZE(32767)
CONV_SECURITY_VERIFICATION(NO)
PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU_LOCATION FQ_PARTNER_LU_NAME(USIBMSJ.MVS3APPC ) 4
WILDCARD_ENTRY(NO)
FQ_OWNING_CP_NAME(USIBMSN.CP000003 )
LOCAL_NODE_NN_SERVER(NO);

DEFINE_PARTNER_LU_LOCATION FQ_PARTNER_LU_NAME(USIBMSN.MVS1APPC ) 4
WILDCARD_ENTRY(NO)
FQ_OWNING_CP_NAME(USIBMSN.CP000003 )
LOCAL_NODE_NN_SERVER(NO);

```

Figure 31 (Part 2 of 3). CM/2 NDF for Using DFM/2

```

DEFINE_MODE  MODE_NAME(QPCSUPP ) 5
              COS_NAME(#CONNECT)
              DEFAULT_RU_SIZE(NO)
              MAX_RU_SIZE_UPPER_BOUND(1920)
              RECEIVE_PACING_WINDOW(7)
              MAX_NEGOTIABLE_SESSION_LIMIT(32767)
              PLU_MODE_SESSION_LIMIT(64)
              MIN_CONWINNERS_SOURCE(32);

DEFINE_DEFAULTS  IMPLICIT_INBOUND_PLU_SUPPORT(YES)
                  DEFAULT_MODE_NAME(BLANK)
                  MAX_MC_LL_SEND_SIZE(32767)
                  DIRECTORY_FOR_INBOUND_ATTACHES(*)
                  DEFAULT_TP_OPERATION(NONQUEUED_AM_STARTED)
                  DEFAULT_TP_PROGRAM_TYPE(BACKGROUND)
                  DEFAULT_TP_CONV_SECURITY_RQD(NO)
                  MAX_HELD_ALERTS(10);

START_ATTACH_MANAGER;

```

Figure 31 (Part 3 of 3). CM/2 NDF for Using DFM/2

**Notes:**

- 1** Definition of our local workstation.
- 2** Definition of the link from our TRLAN to the network.
- 3** Definition of the MVS/APPC LU names on the MVS systems.
- 4** Definition of the location of the MVS systems.
- 5** Definition of the communication mode that DAS uses.

## 2.2.2 DFM/2 Definitions

You must create a CONFIG.DFM file containing your definitions for DFM/2.

For a complete description of the CONFIG.DFM file see the *ADSTAR Distributed Storage Manager Data Access Services Distributed FileManager Administration and Customization Guide*.

Figure 32 shows the CONFIG.DFM file that we used initially.

```
; Definition of an MVS/ESA system as target:
DFM_TARGET (
    remote_lu(MVS3DFM)
    description(MVS3 in San Jose)
    conversation(HOLD)
    max_send_limit(4096)
    userid (PJRES1)
)

; Definition of an MVS/ESA system as target:
DFM_TARGET (
    remote_lu(MVS1DFM)
    description(MVS1 in Santa Teresa)
    conversation(HOLD)
    max_send_limit(4096)
    userid (PJRES1)
)

; Define the local LU alias:
LOCAL_LU (DFMRES)

; Define the Mode Name:
MODE_NAME (QPCSUPP)

; Define the Default DFM Target System:
DEFAULT_DFM_TARGET(MVS3DFM)

; Define the default CCSID (Coded-Character-Set-Id):
DEFAULT_CCSID(00037)

; ***** end of file *****
```

Figure 32. DFM/2 Configuration File (CONFIG.DFM)

You must use the following keywords:

- DFM\_TARGET** identifies a target system. Specify the partner LU alias name that you set in Communications Manager Setup. See Figure 24 on page 18. You must have at least one DFM\_TARGET keyword, and you must identify each target with a separate DFM\_TARGET keyword.
- LOCAL\_LU** identifies the PWS source system. Specify the local node alias name that you set in Communications Manager Setup. See Figure 19 on page 15.

The following keywords are optional in the CONFIG.DFM file:

<b>DEFAULT_DFM_TARGET</b>	identifies the default target system. This is the default name for drive letter assignment if you do not specify a target system. You can have only one DEFAULT_DFM_TARGET keyword.
<b>MODE_NAME</b>	specifies the mode name to be used by DFM/2. The only mode name that is supported is QPCSUPP, this value is also the default. You can only have one MODE_NAME keyword.
<b>DEFAULT_CCSID</b>	is the default coded character set for remote record file data. The default value is 00500. You can have only one DEFAULT_CCSID keyword.
<b>FILE_DESCRIPTOR_MAP</b>	controls data conversion for remote record file data.
<b>DEFAULT_CONVERSION_TABLE</b>	controls conversion between ASCII and Extended Binary Coded Decimal Interchange Code (EBCDIC) for all file names exchanged with a Distributed FileManager target system. The DEFAULT_CONVERSION_TABLE keyword has no influence on the conversion of remote file data contents.
<b>DFM_CACHE</b>	controls caching for remote record access.
<b>TRACE_BUFFER</b>	defines the size of the buffer for trace entries.

The sample diskette (see Appendix A, "Sample Diskette Description" on page 71) contains the CONFIG.BAS file, which is a copy of our initial CONFIG.DFM file. This CONFIG.BAS file contains all of the definitions shown in Figure 32 on page 25. The sample diskette also contains the following CONFIG files:

<b>CONFIG.CPP</b>	CONFIG.DFM file including additions for the C sample application
<b>CONFIG.PLI</b>	CONFIG.DFM file including additions for the PL/I sample application
<b>CONFIG.DFM</b>	CONFIG.DFM file including additions for the AS/400 system

Choose the file that is closest to your needs, copy the file to your PWS hard disk, rename it to CONFIG.DFM, and use it when you start DFM/2. If you use the same LU names as we did (MVS1DFM or MVS3DFM, and DFMRES), you only have to change the userid parameter. If you want, you can of course create your own CONFIG.DFM file.

### 2.2.2.1 Starting DFM/2

After you have activated your CM/2 configuration and created, or copied and modified, the CONFIG.DFM file, you can start DFM/2:

1. Make the directory containing your CONFIG.DFM file the current directory.
2. Enter the command:

STRTDFMC

If you have specified a userid parameter in DFM\_TARGET, you will be prompted for a password. If you do not have a userid parameter in DFM\_TARGET, you will be prompted for a userid and password. The prompts are repeated for each DFM\_TARGET that you have defined.

The STRTDFMC command starts the DFM/2 communication environment and the Remote Stream Access Support functions.

3. To start the Remote Record Access Support function enter the command:

STRTDFMR

Figure 33 shows the OS/2 window after we entered STRTDFMC and STRTDFMR.

```
[D:\DFMSAMP]strtdfmc

Distributed FileManager (DFM/2) Version 1.00

PC Support/400
Operating System/2
Starting Router
(C) Copyright IBM Corp. 1984, 1991. All rights reserved.
Version 2.0 Release 1.1 Level 00

Processing: RTYP CMGR

Processing: A2ET D:\IBMDDM\CONVTABL\04370500.CVT
Processing: E2AT D:\IBMDDM\CONVTABL\05000437.CVT
Processing: LCLN DFMRES

Processing: RMTN MVS3DFM,PJRES1,MVS3 in San Jose
Enter password for system MVS3DFM user ID PJRES1:
Processing: RMTN MVS1DFM,PJRES1,MVS1 in Santa Teresa
Enter password for system MVS1DFM user ID PJRES1:

Processing: MODN QFSPC,QPCSUPP

Processing: RTDN MVS3DFM

Default remote system name: MVS3DFM
stream Function successfully started
STRTDFMC processing complete

[D:\DFMSAMP]strtdfmr

Distributed FileManager (DFM/2) Version 1.00

STRTDFMR processing complete
```

Figure 33. STRTDFMC and STRTDFMR Commands

You should now be able to assign drive letters to your MVS/ESA target system(s). To do this enter the command:

DFMDRIVE

You get the DFMDRIVE - Drive Control window shown in Figure 34 on page 28.

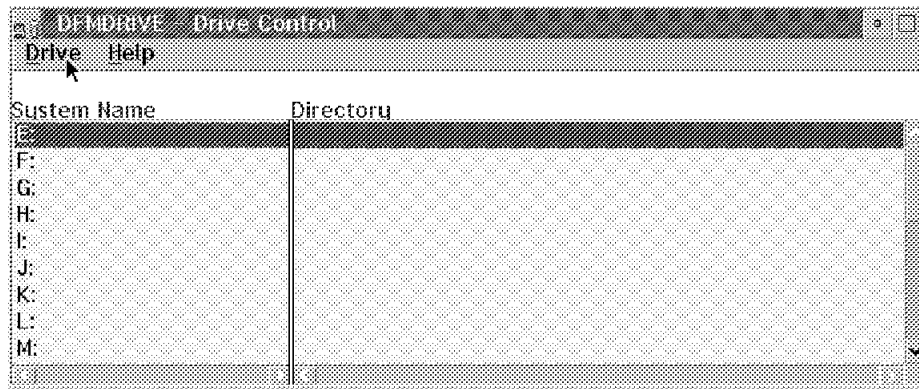


Figure 34. DFMDRIVE - Drive Control Window: No Drive Assigned

As you can see, the E: drive is already selected. If you want, you can select another drive. After you have selected a drive, select **Drive** from the action bar and **Assign...** from the pull-down menu. This takes you to the Assign a Drive to a System window. See Figure 35.

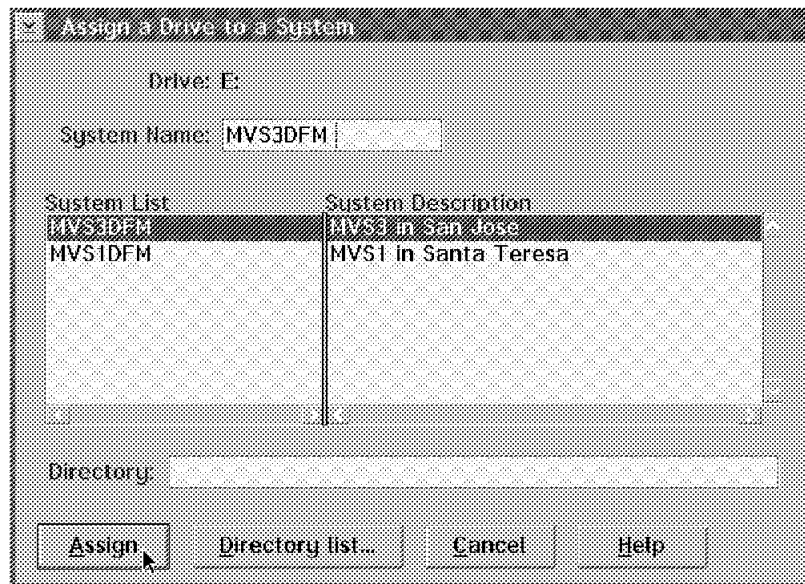


Figure 35. Assign a Drive to a System

The values that are shown in the fields on this window come from the CONFIG.DFM file (Figure 32 on page 25). The System Name field contains the DEFAULT\_DFM\_TARGET. System List and System Description contain the DFM\_TARGET names and descriptions. In the System List, the default system is already selected. If you change the selection, the System Name field also changes. When you select **Assign**, the assignment will be made and you return to the DFMDRIVE - Drive Control window.

We assigned the E: drive to MVS3DFM and the F: drive to MVS1DFM. Figure 36 shows the DFMDRIVE - Drive Control window after we made both assignments.

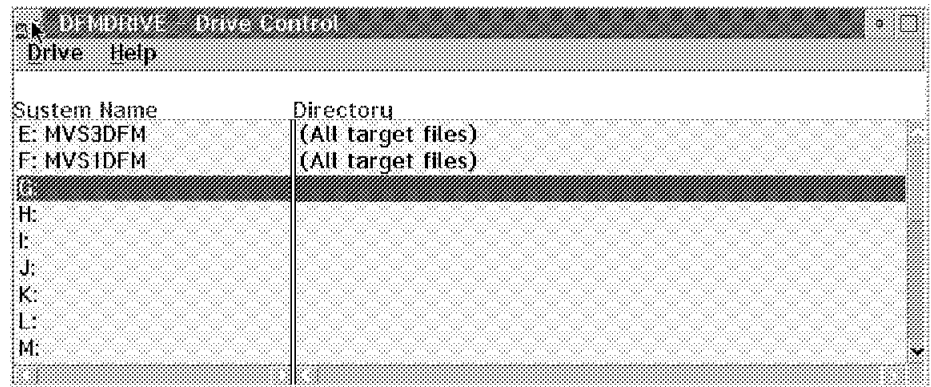


Figure 36. DFMDRIVE - Drive Control Window: Two Drives Assigned

In this section we have explained how to start the two components of DAS and assign logical drives to remote systems by issuing three different DAS commands:

- STRTDFMC
- STRTDFMR
- DFMDRIVE.

In your \IBMDDM\SAMPLE\ directory you have the STARTDFM.CMD command procedure. You can pass to this command procedure as a parameter a Distributed FileManager configuration file name (the default is CONFIG.DFM in the current directory), and the procedure will start all of the components of DAS. In addition to starting the software support, you can modify the DFMDRIVE commands in the procedure to assign logical drives to the remote systems you need to access.

You have now completed the setup for Distributed FileManager and can access data in MVS from applications in your PWS. We suggest that you try some of the samples that we provide.

---

## 2.3 Record Level I/O Samples

In this section we describe two sample applications and explain how to install, customize, and run them. The sample applications are written in C and PL/I.

### 2.3.1 C Application

DFMSAMP is a sample Presentation Manager (PM) application that provides a PM front end to the DAS file access functions from OS/2.

The application is written in the C programming language, and it uses calls to DDM routines to do file operations. The application was developed using the C-Set/2 product. The program module doing the DDM calls is DFMTHRD. You can find a listing of the DFMTHRD.C source code in C.1, "C Sample Application Source Code" on page 76. The DDM calls are marked with <<<<< in the right-hand margin.

This sample application provides several simple functions to display, update, add, and delete records from two keyed files: a customer master file and an account details file. These files can exist locally on OS/2, be remote as DFSMS/MVS V1.2.0 VSAM data sets, or be remote as Application System/400 (AS/400) keyed files. The files can even be placed on different systems; for example, one file can be local on OS/2 and the other remote on MVS or AS/400. Or, as we show in 2.3.1.3, "Execution Scenarios" on page 35, they can be placed on two different MVS systems. The sample also allows you to create and delete the files, either locally or remotely, and you can request information about the files through a query function.

The customer master file uses the customer number as a key. The application also allows use of an alternate index file using the customer last name as a key. We did not use alternate indexes, but you can see references to them in the windows that the application displays.

The main window (Figure 37 on page 32) of the application displays the last customer master record retrieved and enables you to perform the following functions on the record:

<b>Get</b>	get a record
<b>GetNext</b>	get the next record
<b>GetPrev</b>	get the previous record
<b>Account</b>	list the account details
<b>Update</b>	update the record
<b>Add</b>	add a record
<b>Delete</b>	delete the record

The action bar provides three choices with pull-down menus, which present several selections for managing various settings used by the application. Table 1 on page 31 is a quick reference to the action bar choices, the pull-down menu selections, and the functions they perform.



<i>Table 1. DFMSAMP Action Bar Choices and Pull-down Selections</i>		
<b>Action Bar Choice</b>	<b>Pull-down Menu Selection</b>	<b>Function</b>
File Settings	Keyed File Names	Presents the DFMSAMP - Keyed File Settings window where you specify drive IDs and file names
	Open Keyed Files	Opens the files
	Close Keyed Files	Closes the files
	Query File Info	Presents the DFMSAMP - Query Response window with information about the files
	Access By Customer Number	Sets access by customer number
	Access By Customer Name	Sets access by customer name
	Load Keyed Files	Presents the DFMSAMP - Load Keyed Files window where you specify the files from which to load
	Unload Keyed Files	Presents the DFMSAMP - Unload Keyed Files window where you specify the files to which to unload
	View Logical Drives	Calls DFMDRIVE, which presents the DFMDRIVE - Drive Control window where you can view drive assignments and assign and release drives
	Exit Immediately	Exits without prompting
Report Settings	Set Report File Name	Presents the DFMSAMP - Local File Settings window where you specify the name of the report file
	Open Report File	Opens the report file
	Close Report File	Closes the report file
	Write Report Record	Writes current record to the report file
Trace Settings	Set Trace File Name	Presents the DFMSAMP - Trace Settings window where you specify the name of the trace file
	Open Trace File	Opens the trace file
	Close Trace File	Closes the trace file

Figure 37 on page 32 shows the DFMSAMP - Data Access Services Sample Program main window as it is presented when you start the program.

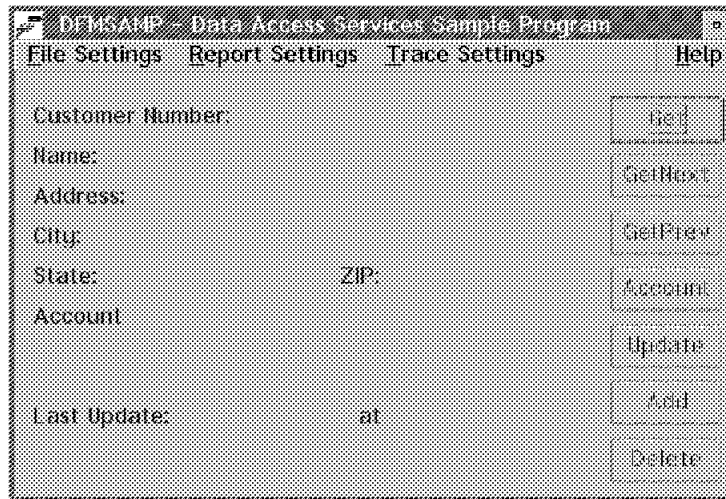


Figure 37. DFMSAMP - Data Access Services Sample Program Main Window

Figure 38 shows the DFMSAMP - Data Access Services Sample Program main window with the File Settings pull-down displayed.

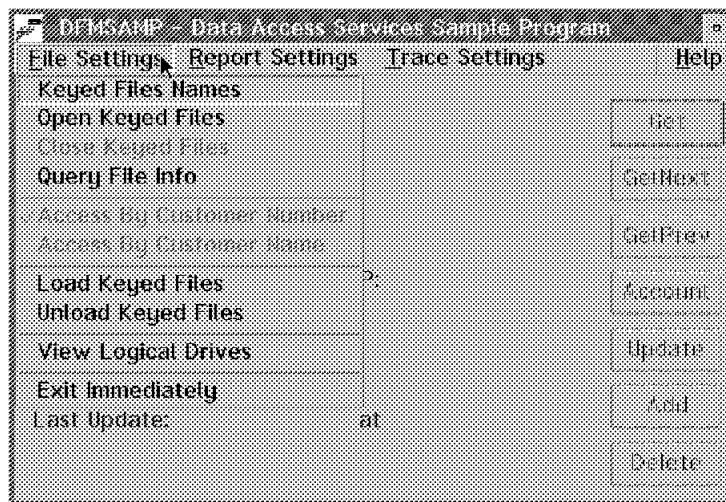


Figure 38. DFMSAMP - Data Access Services Sample Program File Settings Pull-down

### 2.3.1.1 Installation

It is very easy to install the sample. Just follow these steps:

1. Insert the diskette that accompanies this book (see Appendix A, "Sample Diskette Description" on page 71).
2. Create a directory on your hard disk for the sample. If possible use the D drive and the name DFMSAMP.
3. Make the new directory the current directory.
4. Copy all of the files from the DFMSAMP directory on the diskette to your new directory:

```
COPY A:\DFMSAMP\*. * *
```

The following files will be copied to the current directory:

DFMRCLAB ADL	DFMTRACE C	DFMSAMP ICO
DFMRCLAV ADL	DFMUPDTE C	DFMSAMP INI
DFMRCLCB ADL	DFMWNDW C	DFMSAMP IPF
DFMRCLCV ADL	CONFIG DFM	PHONE ICO
DFMADD C	DFMRCLAB DDF	DFMSAMP MAK
DFMDTAIL C	DFMRCLAV DDF	290112LO BMP
DFMGET C	DFMRCLCB DDF	DFMSAMP RC
DFMHHELP C	DFMRCLCV DDF	DFMSAMP RES
DFMKEYED C	DFMSAMP DEF	ACCOUNT SRC
DFMLOAD C	DFMSAMP DEP	CUSTMAST SRC
DFMLOCAL C	DFMSAMP DLG	
DFMQUERY C	DFMSAMP EXE	
DFMRMSG C	DFMHHELP H	
DFMSAMP C	DFMRECRD H	
DFMSCLAS C	DFMSAMP H	
DFMTHRD C	DFMSAMP HLP	

You have now installed the sample application.

### 2.3.1.2 Customization

When you exchange data between a PWS and an MVS/ESA system, you may have to do data conversion. The PWS uses ASCII code, and MVS/ESA uses EBCDIC. If you use the MVS system just for data storage and only access the data from programs in your PWS, you do not need to convert data. If, however, you want to access data from programs in your PWS and from MVS programs, you must convert the data.

Distributed FileManager can convert data for you. DFM/2 supports the IBM A Data Language (ADL). To define data conversion for a record file, you need to create two ADL files:

<b>Base sequence</b>	Contains the description of the record file as it is stored on the remote (MVS) system.
<b>View sequence</b>	Contains the description of the PWS application view of the remote record file.

You translate the ADL files into a DFM/2 internal format called Data Description File (DDF). The C sample provides the following ADL and DDF files for conversion of its application files:

<b>DFMRCLAB</b>	Base (or host) sequence for account details record
<b>DFMRCLAV</b>	View (or application) sequence for account details record
<b>DFMRCLCB</b>	Base (or host) sequence for customer master record
<b>DFMRCLCV</b>	View (or application) sequence for customer master record

You can find listings of the ADL files in C.2, "ADL for Host View of Customer Master Record" on page 102 through C.5, "ADL for Application View of Account Details Record" on page 103.

In addition to data conversion, DFM/2 can do data field remapping. Compare the ADL files in C.2, "ADL for Host View of Customer Master Record" on page 102 and C.3, "ADL for Application View of Customer Master Record" on page 102. The data fields are in different order. The data field CustUpDYr specifies PRECISION(9) in the host ADL file and PRECISION(2) in the application ADL file. The length of this field changes when you use data conversion. The record

length of the data set also changes. Corresponding differences exist for the account details file. The data fields are in different order, and the data field AcctUpDYr has different precision in the host and application views.

You use FILE\_DESCRIPTOR\_MAP keywords in the CONFIG.DFM file to request data conversion. The Distributed FileManager specifies the target system and data sets to be converted. It also specifies the ADL or DDF files that describe how data is converted. Figure 39 shows the FILE\_DESCRIPTOR\_MAP keywords that we added to the CONFIG.DFM file. All data sets with the names \*.DFMSAMP.CUST and \*.DFMSAMP.ACCT on the MVS1DFM and MVS3DFM systems are subject to conversion. The wildcard character, \*, represents any high level qualifier. These FILE\_DESCRIPTOR\_MAP keywords exist in the CONFIG.CPP file on the sample diskette. You must modify the TARGET\_FILENAME parameters, if your data set names do not match. If you created your own CONFIG.DFM file, you must add FILE\_DESCRIPTOR\_MAP keywords to your CONFIG.DFM file, if you want to convert the data.

```
; ADL for customer master file base cluster for MVS1DFM
FILE_DESCRIPTOR_MAP(
    REMOTE_LU(MVS1DFM)
    TARGET_FILENAME(*.DFMSAMP.CUST)
    BASE_DDF(D:\DFMSAMP\DFMRCLCB.DDF)
    VIEW_DDF(D:\DFMSAMP\DFMRCLCV.DDF)
)

; ADL for account details file base cluster for MVS1DFM
FILE_DESCRIPTOR_MAP(
    REMOTE_LU(MVS1DFM)
    TARGET_FILENAME(*.DFMSAMP.ACCT)
    BASE_DDF(D:\DFMSAMP\DFMRCLAB.DDF)
    VIEW_DDF(D:\DFMSAMP\DFMRCLAV.DDF)
)

; ADL for customer master file base cluster for MVS3DFM
FILE_DESCRIPTOR_MAP(
    REMOTE_LU(MVS3DFM)
    TARGET_FILENAME(*.DFMSAMP.CUST)
    BASE_DDF(D:\DFMSAMP\DFMRCLCB.DDF)
    VIEW_DDF(D:\DFMSAMP\DFMRCLCV.DDF)
)

; ADL for account details file base cluster for MVS3DFM
FILE_DESCRIPTOR_MAP(
    REMOTE_LU(MVS3DFM)
    TARGET_FILENAME(*.DFMSAMP.ACCT)
    BASE_DDF(D:\DFMSAMP\DFMRCLAB.DDF)
    VIEW_DDF(D:\DFMSAMP\DFMRCLAV.DDF)
)
```

Figure 39. FILE\_DESCRIPTOR\_MAPs Added to CONFIG.DFM File

To activate the changes you make to the CONFIG.DFM file, you must issue the following commands to start DFM/2 again:

```
STRTDFMC
STRTDFMR
```

You can issue these commands when DFM/2 is active, and changes will take effect immediately.

### 2.3.1.3 Execution Scenarios

To start the C sample application, enter:

```
START DFMSAMP
```

You will get a window with an IBM logo and a copyright notice. When you select **OK**, you get to the DFMSAMP - Data Access Services Sample Program main window shown in Figure 37 on page 32. Click on File Settings and select View Logical Drives on the pull-down menu. DFMDRIVE is started and displays your current drive assignments. If you do not have a drive assigned to your MVS system, assign one now. Click on File Settings again and now select Keyed File Names on the pull-down menu, and you get to the DFMSAMP - Keyed File Settings window shown in Figure 40. On this window you specify the drive IDs and data set names that you want to use.

DFMSAMP - Keyed File Settings

Customer File

Drive ID: E (DFM at MVS3DFM)

Base File Name: PJRES1.DFMSAMP.CUST

Alternate Index Name:

☒ ADL Associated with Customer file

Account File

Drive ID: F (DFM at MVS1DFM)

Base File Name: PJRES1.DFMSAMP.ACCT

☒ ADL Associated with Account file

Save Create Delete Cancel Help

Figure 40. DFMSAMP - Keyed File Settings

The Drive ID fields have drop-down lists where you can see all of the drives from which you can choose. The lists contain both local drives on your PWS and remote drives that you have assigned with DFMDRIVE. Select the drive that you assigned to your MVS system and enter the names of the data sets that you want to use. You must specify fully qualified data set names. If you change only the high level qualifier (PJRES1), you do not have to change the FILE\_DESCRIPTOR\_MAP keywords in the CONFIG.DFM file.

As you can see in Figure 40, we assigned the E drive to MVS3DFM and the F drive to MVS1DFM. We specified the E drive for the customer file and the F drive for the account file; that is, we had the two files on different systems.

You can also see in Figure 40 that we marked the check boxes, ADL Associated with Customer and Account files. As we describe in 2.3.1.2, "Customization" on page 33, the record lengths of the files change if you request data conversion. When the application creates the files through a DDM call, it must supply the record lengths. The application chooses the record lengths based on the ADL

check boxes. The record lengths are longer when you use data conversion. The RECORDSIZE parameter for a VSAM data set indicates the maximum record size. If the target system is MVS, the application always works when you mark the check boxes for ADL. Data conversion is, however, controlled only by the FILE\_DESCRIPTOR\_MAP keywords in the CONFIG.DFM file.

Now select **Create**, and the data sets are allocated. Using Interactive Storage Management Facility (ISMF), you can verify that the data sets have been allocated. Figure 41 shows an ISMF display of our DFMSAMP.ACCT data set. Columns 34 and 35 in the ISMF data set list show DDM attributes and the coded character set identifier (CCSID) description (a list of CCSIDs can be found in Appendix D of the *DFSMS/MVS Version 1 Release 2 DFSMSdfp Storage Administration Reference*). You can see that DDM attributes exist for the data set and that the CCSID ID=00000 is set.

DATA SET LIST						COMMAND ==>		SCROLL ==> CSR	
								Entries 1-3 of 3	
ENTER LINE OPERATORS BELOW:								Data Columns 33-35 of 35	
LINE	OPERATOR	DATA SET NAME	REBLK	DDM	IND	ATTR	CCSID	DESCRIPTION	
---(1)---		----- (2) -----	(33)-	(34)	---	---	----- (35) -----		
		PJRES1.DFMSAMP.ACCT	---	---			ID=00000,	NO DESC	
		PJRES1.DFMSAMP.ACCT.DATA	NO	YES			ID=00000,	NO DESC	
		PJRES1.DFMSAMP.ACCT.INDEX	NO	---			-----		
-----		-----	BOTTOM	OF	DATA	-----	-----		

Figure 41. ISMF Panel with DFMSAMP.ACCT Data Set

Now select Load Keyed Files from the File Settings pull-down menu and you get to the DFMSAMP - Load Keyed Files window shown in Figure 42. If you did not place the sample on the D drive and in the DFMSAMP directory, you must change the drive ID and directory to those you chose. Select **Load** and your files will be loaded with sample data.

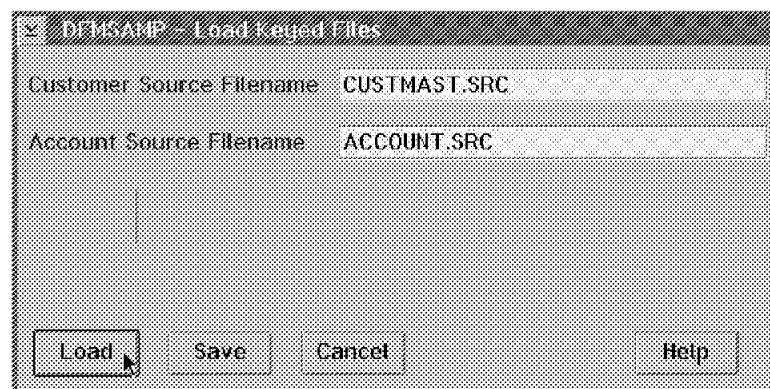


Figure 42. DFMSAMP - Load Keyed Files

Now you can open the files. Select Open Keyed Files from the File Settings pull-down menu. The Get and Add push buttons are now activated. Select **Get** to get to the DFMSAMP - Retrieve by Customer Number window. See Figure 43. Select **List Keys** and the fields are filled with data as shown in Figure 44.

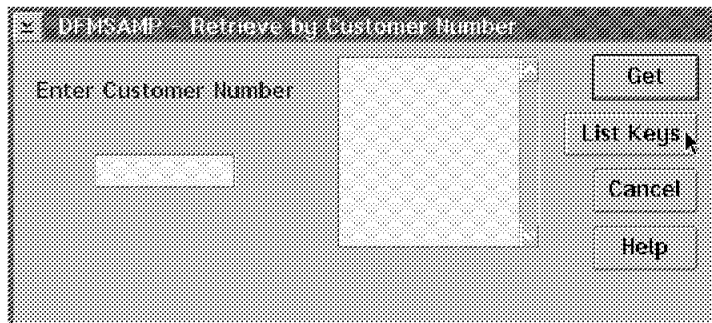


Figure 43. DFMSAMP - Retrieve by Customer Number

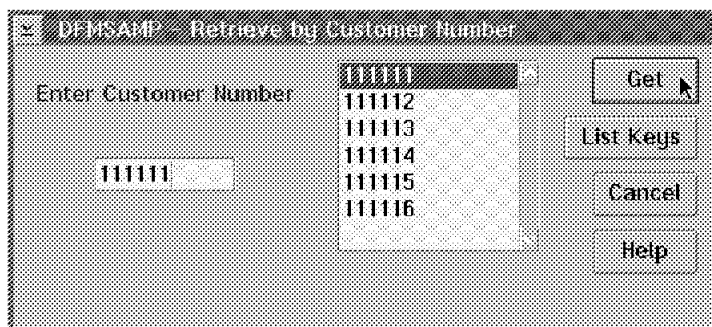


Figure 44. DFMSAMP - Customer Numbers Retrieved

Customer number 11111 has been selected; if you select **Get**, data is read from the customer master file and presented in the DFMSAMP main window. See Figure 45 on page 38.

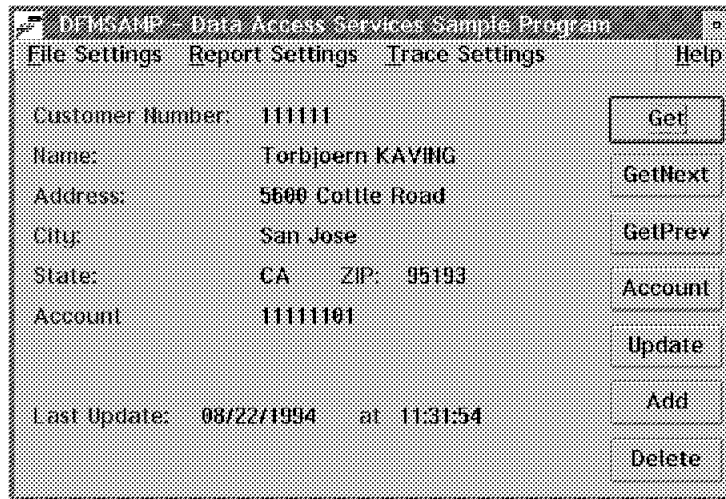


Figure 45. DFMSAMP - Main Window with Data

Now all of the push buttons are activated, and you can display, add, delete, and update data.

We used the data set names PJRES1.DFMSAMP.CUST and PJRES1.DFMSAMP.ACCT. After we updated the data twice, we closed the files. Then we ran LISTCAT ALL for the files. You can find the output from the LISTCATs in C.6, "LISTCAT Output for Customer Master File" on page 104 and C.7, "LISTCAT Output for Account Details File" on page 105. In the statistics part of the LISTCATs you can see that there is a total of six records, two records have been updated, and a number of records have been retrieved.

We also ran PRINT to see the contents of the files (see C.8, "Customer Master File with Data Conversion" on page 106 and C.9, "Account Details File with Data Conversion" on page 106, where the data is converted to EBCDIC).

Data conversion only works if the data set names match the specifications in the FILE\_DESCRIPTOR\_MAPs in the CONFIG.DFM file. To demonstrate that we repeated the exercise with the data set names PJRES1.DFMS.CUST and PJRES1.DFMS.ACCT. Then we made a PRINT of those data sets (see C.10, "Customer Master File without Data Conversion" on page 107 and C.11, "Account Details File without Data Conversion" on page 107). In those data sets, data is stored in ASCII; that is, there was no conversion. Compare C.8, "Customer Master File with Data Conversion" on page 106 with C.10, "Customer Master File without Data Conversion" on page 107 and compare C.9, "Account Details File with Data Conversion" on page 106 with C.11, "Account Details File without Data Conversion" on page 107. You can also see that the record lengths are different.



## 2.3.2 PL/I Application

To use the PL/I sample application, you need PL/I Package/2, program number 5601-388, on your PWS.

### Note !

Before you start the PL/I application make sure that in the LIBPATH statement in the CONFIG.SYS file, the DLL directory that is part of DAS precedes the DLL directory that is part of PL/I.

The PL/I sample application contains three simple PL/I programs, which illustrate how a high level language can access data through Distributed FileManager. You can find the source code of the programs in Appendix D, "PL/I Sample Application Listings" on page 109. The programs use ordinary OPEN, READ, WRITE, and CLOSE statements; that is, they are not modified at all to use Distributed FileManager. The PL/I run-time routines detect that DFM/2 must be called to access the data.

The application maintains a keyed file of manuals using the form number as the key. The sample also provides Restructured Extended Executor (REXX) language procedures that set up the environment for the programs and then call the programs. Table 2 presents an overview of the REXX procedures, the programs, and the functions they perform.

Table 2. PL/I Sample Application: REXX Procedures Overview

REXX Procedure	Program	Function	Source Code
LOAD	PLILOAD	Loads data into keyed file	D.1, "PLILOAD Source Code" on page 109
READ	PLIREAD	Reads keyed file	D.2, "PLIREAD Source Code" on page 110
MAINT	PLIMAINT	Updates keyed file	D.3, "PLIMAINT Source Code" on page 111

There is also a SETDIREC.CMD REXX file, which is used to set a reference to the target system data set that the programs use.

### 2.3.2.1 Installation

You install the PL/I sample application in the same simple way as you install the C sample application. Follow these steps:

1. Insert the diskette that accompanies this book (see Appendix A, "Sample Diskette Description" on page 71).
2. Create a new directory on your hard disk. We used D:\PLISAMP.
3. Make the new directory the current directory.
4. Copy all of the files from the PLISAMP directory on the diskette to your new directory:

```
COPY A:\PLISAMP\*.* *.* *
```

The following files will be copied to your current directory.

DDMOUTB	ADL	MAINT	CMD	PLIMAIN	EXE
DDMOUTV	ADL	READ	CMD	PLIREAD	EXE
ADDBOOKS	CAT	SETDIREC	CMD	PLILOAD	PLI
MAINTCAT	CAT	DDMOUTB	DDF	PLIMAIN	PLI
COMPLINK	CMD	DDMOUTV	DDF	PLIREAD	PLI
LOAD	CMD	PLILOAD	EXE		

You have now installed the sample application.

### 2.3.2.2 Customization

As we describe in 2.3.1.2, “Customization” on page 33, DFM/2 can do data conversion for you. Conversion requires a set of ADL and DDF files. The PL/I sample provides the following ADL and DDF files for conversion of its data file:

**DDMOUTB** Base sequence for the data file

**DDMOUTV** View sequence for the data file

To use the DDF files for data conversion, you must have FILE\_DESCRIPTOR\_MAP keywords for them in your CONFIG.DFM file. Figure 46 shows the FILE\_DESCRIPTOR\_MAP keywords that we added to the CONFIG.DFM file for the PL/I sample application. All data sets with the name \*.PLISAMP.DATA on the MVS1DFM and MVS3DFM systems are subject to conversion. The wildcard character, \*, represents any high level qualifier. The CONFIG.PLI file on the sample diskette contains these FILE\_DESCRIPTOR\_MAP keywords. If you created your own CONFIG.DFM file and want to convert data, you must add FILE\_DESCRIPTOR\_MAP keywords to your CONFIG.DFM file.

```
; PLISAMP ADL for sample data set cluster for MVS1DFM
FILE_DESCRIPTOR_MAP(
    REMOTE_LU(MVS1DFM)
    TARGET_FILENAME(*.PLISAMP.DATA)
    BASE_DDF(D:\PLISAMP\DDMOUTB.DDF)
    VIEW_DDF(D:\PLISAMP\DDMOUTV.DDF)
)

; PLISAMP ADL for sample data set cluster for MVS3DFM
FILE_DESCRIPTOR_MAP(
    REMOTE_LU(MVS3DFM)
    TARGET_FILENAME(*.PLISAMP.DATA)
    BASE_DDF(D:\PLISAMP\DDMOUTB.DDF)
    VIEW_DDF(D:\PLISAMP\DDMOUTV.DDF)
)
```

Figure 46. FILE\_DESCRIPTOR\_MAPs Added for the PL/I Sample

To activate the changes you make to the CONFIG.DFM file, you must issue the following commands to start DFM/2 again:

```
STRTDPMC
STRTDFMR
```

You can issue these commands when DFM/2 is active, and changes will take effect immediately.

### 2.3.2.3 Execution Scenarios

Before you can use the PL/I sample programs, you must create the VSAM data set that the application uses. Figure 47 shows the job step we used to allocate the data set. If you change only the high level qualifier (PJRES1), your data set name matches the FILE\_DESCRIPTOR\_MAP keywords shown in Figure 46 on page 40, and you do not have to change the CONFIG.DFM file to convert the data.

```
//DEFINE EXEC  PGM=IDCAMS
//SYSPRINT DD  SYSOUT=*
//SYSIN      DD  *
  DEFINE CLUSTER(
    NAME(PJRES1.PLISAMP.DATA) -
    TRACKS(1 1) -
    RECSZ(80 80) -
    KEYS(14 0) )
```

Figure 47. JCL to Create the VSAM Data Set

The directory where you installed the PL/I sample application must be the current directory. DFM/2 must be active; that is, you must have issued these commands:

```
STRTDPMC
STRTDFMR
```

Start DFMDRIVE to verify that you have a drive assigned to your MVS system. If a drive is not assigned, assign one. To specify your drive ID and data set name for the PL/I sample, enter SETDIREC. The REXX procedure SETDIREC prompts you for the drive ID and data set name and then issues the SET DD:DIREC= command that is needed to run the PL/I programs. Figure 48 shows the execution of the SETDIREC command.

```
[D:\PLISAMP]setdirec
Enter the drive letter that you assigned to your target system:
e
Enter your data set name:
pjres1.plisamp.data

[D:\PLISAMP]SET DD:DIREC=E:PJRES1.PLISAMP.DATA,AMTHD(DDM)

[D:\PLISAMP]
```

Figure 48. Execution of the SETDIREC Command

Now you can load data into your data set. The ADDBOOKS.CAT file (Figure 49 on page 42) contains sample data that you can use to load your data set.

GG243720	1	93/02/17	ESCON Manager Presentation	I
PROASET	0	93/06/17	Asset Management System Best Process Rollout Guide	I
A3S13M01	0	93/02/23	NTS/2 Messages and Problem Determination Guide	I
GG243549	1	93/02/17	Using NetView R3 PPI	I
GG243751	0	93/02/17	POWER Networking Guide	I
GG243917	0	93/02/17	LANDP Networking with APPN	I
GG244087	0	93/09/15	Using NFS in a Multivendor Environment	I
DMSE6AZZ	1	93/02/17	USING ONLINE DOCUMENTATION	I
GG663239	1.0	93/02/17	ESCON Operator Prob Deter in the MVS/ESA Environment	J
IBMBK005	0	93/08/22	PS/2, OS/2, PC, Printer, Modem, X.25	I

Figure 49. Sample Data for the PL/I Application

Use the REXX procedure LOAD to call the PLILOAD program. Just enter LOAD, and your data set will be loaded as shown in Figure 50. The REXX procedure LOAD issues the SET DD:SYSIN= command so that DD:SYSIN refers to the file that contains the sample data.

```
[D:\PLISAMP]load
[D:\PLISAMP]SET DD:SYSIN=ADDBOOKS.CAT,RECSIZE(81)
[D:\PLISAMP]PLILOAD

GG243720 1      93/02/17  ESCON Manager Presentation
PROASET  0      93/06/17  Asset Management System Best Process Rollout Guide
A3S13M01 0      93/02/23  NTS/2 Messages and Problem Determination Guide
GG243549 1      93/02/17  Using NetView R3 PPI
GG243751 0      93/02/17  POWER Networking Guide
GG243917 0      93/02/17  LANDP Networking with APPN
GG244087 0      93/09/15  Using NFS in a Multivendor Environment
DMSE6AZZ 1      93/02/17  USING ONLINE DOCUMENTATION
GG663239 1.0    93/02/17  ESCON Operator Prob Deter in the MVS/ESA Environment
IBMBK005 0      93/08/22  PS/2, OS/2, PC, Printer, Modem, X.25

                        10 Total records processed
                        10 Total records read

[D:\PLISAMP]
```

Figure 50. PL/I Sample: Loading Data

Enter READ to read the data set. The data set is read and the records are presented on your PWS screen as shown in Figure 51 on page 43.

```

[D:\PLISAMP]read
[D:\PLISAMP]PLIREAD

A3S13M01 0    93/02/23 NTS/2 Messages and Problem Determination Guide
DMSE6AZZ 1    93/02/17 USING ONLINE DOCUMENTATION
GG243549 1    93/02/17 Using NetView R3 PPI
GG243720 1    93/02/17 ESCON Manager Presentation
GG243751 0    93/02/17 POWER Networking Guide
GG243917 0    93/02/17 LANDP Networking with APPN
GG244087 0    93/09/15 Using NFS in a Multivendor Environment
GG663239 1.0  93/02/17 ESCON Operator Prob Deter in the MVS/ESA Environment
IBMBK005 0    93/08/22 PS/2, OS/2, PC, Printer, Modem, X.25
PROASET 0    93/06/17 Asset Management System Best Process Rollout Guide

                                10 Total records processed

[D:\PLISAMP]

```

Figure 51. PL/I Sample: Reading Data

The MAINTCAT.CAT file (Figure 52) contains sample data that you can use to update the data set. Enter MAINT, and the PLIMAIN program is run as shown in Figure 53 on page 44. The REXX procedure MAINT issues the SET DD:SYSIN= command so that DD:SYSIN refers to the file that contains the sample data for updating the VSAM data set.

GG243720 1	93/02/17 ESCON Manager Presentation Graphics	UPD
PROASET 0	93/06/17 Asset Management System Best Process Rollout Guide	DEL
A3S13M01 0	93/02/23 NTS/2 Messages and Problem Determination Guide	DEL
GG243806 0	93/10/05 Application Development on the AS/400	UPD
KADWAW		DEL
\$DB2 0	93/02/16 DB/2 Performance Monitor R2.1 Library	ADD
GG243878 0		DEL
GG663163 1.0	93/10/05 DFSMS Early Support Program Experiences	UPD
\$REL2ACI 2.0	93/02/16 AS/400 Communications Library Release 2.0	ADD
IBMBK005 0	93/10/05 PS/2, OS/2, PC, Printer, Modem and X.25 Catalog	UPD

Figure 52. Sample Update Data for the PL/I Application

```

[D:\PLISAMP]maint

[D:\PLISAMP]SET DD:SYSIN=MAINTCAT.CAT,RECSIZE(83)

[D:\PLISAMP]PLIMAIN

Updating .
GG243720 1      93/02/17 ESCON Manager Presentation Graphics

Deleting .
PROASET  0      93/06/17 Asset Management System Best Process Rollout Guide

Deleting .
A3S13M01 0      93/02/23 NTS/2 Messages and Problem Determination Guide

FormNo [GG243806 0 ] is not on file
Record will not be Updated

FormNo [KADWAWS  ] is not on file
Record will not be deleted

Adding ...
$DB2     0      93/02/16 DB/2 Performance Monitor R2.1 Library

FormNo [GG243878 0 ] is not on file
Record will not be deleted

FormNo [GG663163 1.0 ] is not on file
Record will not be Updated

Adding ...
$REL2ACI 2.0    93/02/16 AS/400 Communications Library Release 2.0

Updating .
IBMBK005 0      93/10/05 PS/2, OS/2, PC, Printer, Modem and X.25 Catalog

[D:\PLISAMP]

```

*Figure 53. PL/I Sample: Updating Data*

If you want, you can now read the data set again and see that changes have been made.

We ran the sample as it is described here. After that we ran a LISTCAT and a PRINT to see catalog information and the data contents of the data set. The LISTCAT output is in D.6, "LISTCAT Output of PL/I Data Set after Running Sample" on page 114, and the PRINT output is in D.7, "PL/I Sample Data Set after Running Sample" on page 115.

We also used ISMF to display data set information. Figure 54 shows an ISMF display of our PLISAMP.DATA data set. Columns 34 and 35 in the ISMF data set list show DDM attributes and the CCSID description. You can see that there are no DDM attributes for the data set and the CCSID is not set. This is correct because Distributed FileManager did not create the data set.

DATA SET LIST				
COMMAND ==>		SCROLL ==> CSR		
		Entries 1-3 of 3		
ENTER LINE OPERATORS BELOW:		Data Columns 33-35 of 35		
LINE OPERATOR	DATA SET NAME	REBLK IND	DDM ATTR	CCSID DESCRIPTION
---(1)---	------(2)-----	(33)-	(34)	------(35)-----
	PJRES1.PLISAMP.DATA	---	---	-----
	PJRES1.PLISAMP.DATA.DATA	NO	NO	-----
	PJRES1.PLISAMP.DATA.INDEX	NO	---	-----
-----	-----	BOTTOM	OF DATA	-----

Figure 54. ISMF Panel with PLISAMP.DATA Data Set

---

## 2.4 Stream I/O Samples

In this section we describe how you can extend the storage capacity of your PWS by using direct access storage devices (DASD) on an MVS/ESA system. We cover the following topics:

- Stream I/O support
- Stream I/O setup
- Usage scenarios.

### 2.4.1 Stream I/O Support

DFM/MVS supports stream files in sequential access method (SAM) or partitioned data set extended (PDSE) members. DFM/MVS also supports stream access to record files in other types of MVS/ESA data sets.

You can create, rename, delete, and modify DDM attributes, and retrieve them for stream files in the following data set types:

- SMS-managed SAM data sets on DASD
- SMS-managed PDSE members.

### 2.4.2 Stream I/O Setup

To use stream I/O to access stream files on MVS you use DFMDRIVE to assign a drive letter to your stream file data sets. The data set does not have to exist when you assign a drive letter. If the data set does not exist, it will be created when you store the first OS/2 file in it. When you start DFMDRIVE and select a drive to assign, you get to the Assign a Drive to a System window. To assign the drive letter to a data set, you select the system and then enter the PDSE name or the data set name prefix in the Directory entry field as shown in Figure 55.

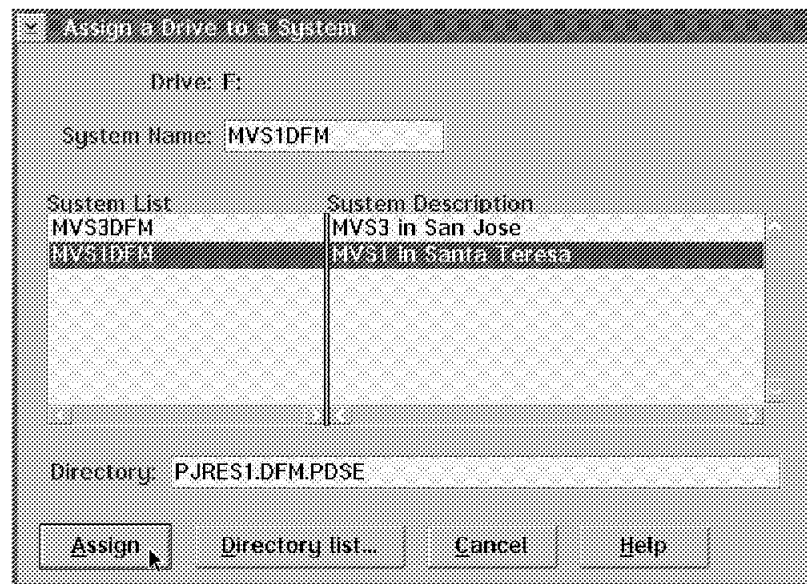


Figure 55. Assign a Drive to an MVS Data Set



When you select **Assign**, you return to the DFMDRIVE - Drive Control window, and you see your data set name in the Directory column to the right of your MVS system alias name. In Figure 56 on page 47 you can see the DFMDRIVE - Drive Control window from our PWS, after we assigned the F-drive to the PJRES1.DFM.PDSE data set and the G-drive to the PJRES1.DFM.SAM data set prefix on the MVS1DFM system.

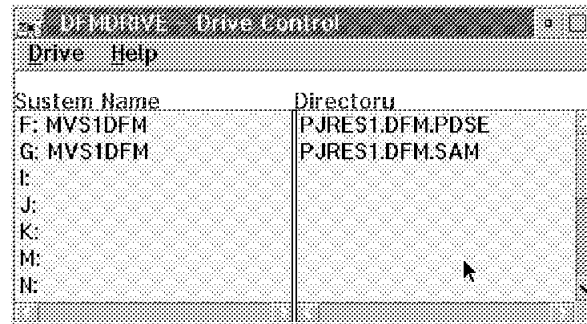


Figure 56. Drives Assigned to MVS Data Sets

You can now use the MVS data sets as a hard disk on your PWS. Depending on which file system you are using on your PWS, file allocation table (FAT) or high performance file system (HPFS), the files on your PWS hard disks have names in one of these formats:

```
[drive:] [path] filename.ext
[drive:] [path] filename
```

The file name format that you use on the assigned drives also depends on the MVS data set type. For SAM data sets the file name must comply with the MVS rules for data sets, and the total length, including your data set prefix, cannot exceed 44 characters. For PDSE members, names can be a maximum of eight characters and may not contain a period, so the normal file name format is not acceptable for PDSEs. You must put the member name in parentheses. The file name format you use to access a PDSE member is:

```
[drive:](member name)
```

### 2.4.3 Stream I/O Usage Scenarios

To illustrate the use of MVS data sets for stream I/O, we copied the LOAD.CMD and READ.CMD files from the PLISAMP directory to the MVS assigned drives. Figure 57 on page 48 shows the OS/2 COPY commands that we used. The MVS data sets (PDSE and SAM) were created when we issued the first copy command to each drive.

```

[D:\PLISAMP] dir *.cmd

The volume label in drive D is PJRES1-DATA.
The Volume Serial Number is 669E:4815
Directory of D:\PLISAMP

2-24-94   2:57p       742         859  COMPLINK.CMD
8-23-94   2:14p        96         640  LOAD.CMD
8-23-94   2:15p        97         641  MAINT.CMD
8-23-94   2:15p        50         583  READ.CMD
8-23-94   2:17p       744         942  SETDIREC.CMD
          5 file(s)          1729 bytes used
                               236608512 bytes free

[D:\PLISAMP] copy ?????.cmd g:
LOAD.CMD
The extended attributes for the file or directory
were discarded because the target file system does not
support extended attributes.
READ.CMD
The extended attributes for the file or directory
were discarded because the target file system does not
support extended attributes.
          2 file(s) copied.

[D:\PLISAMP] copy load.cmd f:(LOAD)
The extended attributes for the file or directory
were discarded because the target file system does not
support extended attributes.
          1 file(s) copied.

[D:\PLISAMP] copy read.cmd f:(READ)
The extended attributes for the file or directory
were discarded because the target file system does not
support extended attributes.
          1 file(s) copied.

[D:\PLISAMP]

```

Figure 57. Copy Files to SAM Data Sets and PDSE Members

To verify that the files were copied, we issued the DIR command for the assigned drives. As you can see in Figure 58 on page 49, the files exist either as members in the PDSE or as MVS sequential data sets.

**Note:** You can ignore the SYS0317 and SYS0050 error messages that are displayed when you execute the DIR command. The developers are aware of the problem and working on a fix.

```

[D:\PLISAMP] dir f:

The volume label in drive F is MVS1DFM.
Directory of F:\

10-20-94  4:21a      96          0 (LOAD)
10-20-94  4:21a      50          0 (READ)
SYS0317: The system cannot find message 0328
in message file OS0001.MSG.
        2 file(s)          0 bytes used
SYS0050: The network request is not supported.

[D:\PLISAMP] dir g:

The volume label in drive I is MVS1DFM.
Directory of I:\

10-20-94  2:42p      96          0 LOAD.CMD
10-20-94  2:42p      50          0 READ.CMD
SYS0317: The system cannot find message 0328
in message file OS0001.MSG.
        2 file(s)      146 bytes used
SYS0050: The network request is not supported.

[D:\PLISAMP]

```

Figure 58. DIR Command for the Assigned Drives

Issue the TYPE command to read a member, as shown in Figure 59.

```

[D:\PLISAMP]type f:(load)
/* */

trace o

' SET DD:SYSIN=ADDBOOKS.CAT,RECSIZE(81)'

' PLILOAD'

Exit 0

[D:\PLISAMP]

```

Figure 59. Reading a Stream-File from an MVS Data Set

We also logged on to Time Sharing Option (TSO) and verified that the data sets were created on the MVS system. Figure 60 on page 50 shows the ISPF data set list. When we entered the BROWSE line operator next to the PDSE name, we got the member list shown in Figure 61 on page 50.

DSLST - DATA SETS BEGINNING WITH PJRES1.DFM -----					ROW 1 OF 3
COMMAND ==>					SCROLL ==> CSR
COMMAND	NAME	DSORG	RECFM	LRECL	BLKSZ
-----					
	PJRES1.DFM.PDSE	PO-E	VB	8196	32760
	PJRES1.DFM.SAM.LOAD.CMD	PS	VB	8196	23476
	PJRES1.DFM.SAM.READ.CMD	PS	VB	8196	23476
***** END OF DATA SET LIST *****					

Figure 60. ISPF Data Set List of the MVS Data Sets

BROWSE - PJRES1.DFM.PDSE -----					ROW 00001 OF 00002
COMMAND ==>					SCROLL ==> CSR
NAME	VV.MM	CREATED	CHANGED	SIZE	INIT MOD ID
LOAD					
READ					
**END**					

Figure 61. Member List of the PDSE

We selected the LOAD member and the PJRES1.DFM.SAM.LOAD.CMD data set to look at the data contents, which are shown in Figure 62. As expected they are unreadable because data conversion is not done for stream I/O files. Data is stored in ASCII code.

BROWSE -- PJRES1.DFM.PDSE(LOAD) -----				
COMMAND ==>				
***** TOP OF DATA *****				
-----				
!.....ëê/Äá.?.....ëäë.ää.ëBēñ+.äää  .ë.äë.ëääēñ!á.....&<ñ<				
22222000027766626000022225452443555443444444452445254454542332200002222544444				
FA00AFDADA0421350FDADA0007354044A3939ED1442FFB3E314C25339A588197DADA00070C9CF1				
***** BOTTOM OF DATA ****				
BROWSE -- PJRES1.DFM.SAM.LOAD.CMD -----				
COMMAND ==>				
***** TOP OF DATA *****				
-----				
!.....ëê/Äá.?.....ëäë.ää.ëBēñ+.äää  .ë.äë.ëääēñ!á.....&<ñ<				
22222000027766626000022225452443555443444444452445254454542332200002222544444				
FA00AFDADA0421350FDADA0007354044A3939ED1442FFB3E314C25339A588197DADA00070C9CF1				
***** BOTTOM OF DATA ****				

Figure 62. Contents of the PDSE Member and the SAM Data Set

---

## Chapter 3. DFM/2 and AS/400 Distributed Data Management

After we completed the implementation of the samples in the MVS/ESA environment, we added an AS/400 system to our configuration. We used the AS/400 as a Distributed FileManager target system. We ran the C sample again, now accessing data in the AS/400 system.

In this chapter we describe the AS/400 environment where we implemented the sample. We also describe how to add support for an AS/400 as a target system in the Distributed FileManager configuration. The chapter contains the following topics:

- AS/400 environment
- OS/2 environment
- Record level I/O sample
- Stream I/O sample

---

### 3.1 AS/400 Environment

We used an AS/400 system 9404 model E20 connected to the same TRLAN as our PWS. The AS/400 used the OS/400 Version 2.3 operating system. Distributed FileManager source and target system support is integrated in the OS/400 operating system. We used the AS/400 only as a Distributed FileManager target system.

To use the AS/400 system as a Distributed FileManager target system you must acquire some information related to the AS/400 network definitions. You will use this information later for Communications Manager Setup. First you need the AS/400 network address, which you can obtain by issuing the DSPNETA command from an AS/400 system user ID. Figure 63 on page 52 shows the output of the DSPNETA command.

```

                                Display Network Attributes
                                System:  SJAS400A
Current system name . . . . . : SJAS400A
  Pending system name . . . . . :
Local network ID . . . . . : USIBMSN
Local control point name . . . . . : SJAS400A
Default local location . . . . . : SJAS400A
Default mode . . . . . : QPCSUPP
APPN node type . . . . . : *NETNODE
Data compression . . . . . : *NONE
Intermediate data compression . . . . . : *NONE
Maximum number of intermediate sessions . . . . : 200
Route addition resistance . . . . . : 128
Server network ID/control point name . . . . . : *LCLNETID  *ANY

More...

Press Enter to continue.

F3=Exit  F12=Cancel

```

Figure 63. AS/400 Host: Display Network Attributes Command Output

You also need the Local network ID (USIBMSN) and the Default local location (SJAS400A) when you define the partner LU in Communications Manager Setup.

The other information that you need is the TRLAN destination address of the AS/400 system. To get the token-ring address do the following:

1. Issue the WRKLIND command to get to the Work With Line Descriptions.
2. Enter option 5, Display, next to the active token-ring line.

The output of the display command includes the LAN adapter address of the AS/400 as shown in Figure 64 on page 53.

```

                                Display Line Description
                                01/09/95 13:46:52 SJAS400A
Line description . . . . . : LIND      TRLINE
Option . . . . . : OPTION    *BASIC
Category of line . . . . . :          *TRLAN

Resource name . . . . . : RSRNAME  LINO21
Online at IPL . . . . . : ONLINE   *YES
Vary on wait . . . . . : VRYWAIT  *NOWAIT
Network controller . . . . . : NETCTL TRLINNET
Maximum controllers . . . . . : MAXCTL 40
Line speed . . . . . : LINESPEED 4M
Maximum frame size . . . . . : MAXFRAME 4060
TRLAN manager logging level . . . : TRNLOGLVL *OFF
Current logging level . . . . . :          *OFF
TRLAN manager mode . . . . . : TRNMGRMODE *OBSERVING
Log configuration changes . . . . : LOGCFGCHG *LOG
Token-ring inform of beacon . . . : TRNINFCN *YES
Local adapter address . . . . . : ADPTADR 400052047158

Press Enter to continue.

F3=Exit  F11=Nondisplay keywords  F12=Cancel

```

Figure 64. AS/400 Host: Display Line Description Command Output

There are few other definitions that you need to have on the AS/400 system if you want to use it as a Distributed FileManager target system. You should ask your AS/400 system administrator to verify that:

- You have a userid on the AS/400 system
- Your AS/400 userid is enrolled as a PC support user
- Your OS/2 has a controller and a device description on the token-ring line.

The controller and device descriptions are created either manually, using configuration commands, or automatically, if the system is enabled to automatic configuration. (The controller and device descriptions are created automatically when a partner tries to start a conversation with the AS/400.)

On our AS/400 system, the PJRES1 userid is enrolled as a PC support user, and the token-ring line is defined with automatic configuration. Figure 65 on page 54 shows the controller and device descriptions that were created for our OS/2 workstation.

Display Controller Description

SJAS400A

01/09/95 13:54:32

Controller description . . . . . : SNC4879I  
Option . . . . . : \*BASIC  
Category of controller . . . . . : \*APPC  
  
Link type . . . . . : \*LAN  
Online at IPL . . . . . : \*NO  
Active switched line . . . . . : TRLINE  
Character code . . . . . : \*EBCDIC  
Maximum frame size . . . . . : 16393  
Remote network identifier . . . . . : USIBMSN  
Remote control point . . . . . : SNC4879I  
Initial connection . . . . . : \*DIAL  
Dial initiation . . . . . : \*LINKTYPE  
Switched disconnect . . . . . : \*YES  
Data link role . . . . . : \*NEG  
LAN remote adapter address . . . . . : 400052047163

More...

Press Enter to continue.

F3=Exit F11=Display keywords F12=Cancel

Display Device Description

SJAS400A

01/09/94 13:54:53

Device description . . . . . : SNC4879I  
Option . . . . . : \*BASIC  
Category of device . . . . . : \*APPC  
  
Automatically created . . . . . : YES  
Remote location . . . . . : SNC4879I  
Online at IPL . . . . . : \*NO  
Local location . . . . . : SJAS400A  
Remote network identifier . . . . . : \*NETATR  
Attached controller . . . . . : SNC4879I  
Message queue . . . . . : QSYSOPR  
Library . . . . . : \*LIBL  
Local location address . . . . . : 00  
APPN-capable . . . . . : \*YES  
Single session:  
Single session capable . . . . . : \*NO

More...

Press Enter to continue

F3=Exit F11=Display keywords F12=Cancel

Figure 65. AS/400 Host: Display Controller and Device Description Command Output

We also created a folder on the AS/400 system to hold all of the stream data files that we create from our OS/2 workstation.

To create a folder we used the CRTFLR command. The folder name is PJRES1.



---

## 3.2 OS/2 Environment

On the OS/2 workstation you need to add your AS/400 target system to the CM/2 and DFM/2 configuration files.

### 3.2.1 Communications Manager/2 Definitions

If you want to communicate with an AS/400 system, the AS/400 system must be defined to CM/2. When you set up a communication link from your PWS to an AS/400 system, you probably also want to be able to log on to the AS/400 system using your PWS as a terminal. Therefore you must add 5250 emulation support to CM/2. For that you need your CM/2 installation diskettes. To add 5250 emulation support and define the AS/400 system to CM/2 in our workstation, we used Communications Manager Setup. The description that follows guides you through the Communications Manager Setup windows to add an AS/400 to your Distributed FileManager configuration.

Start Communications Manager Setup as described in 2.2.1, “Communications Manager/2 Definitions” on page 11. Follow the description until you get to the Communications Manager Configuration Definition window shown in Figure 16 on page 13. At this point, it is a good idea to make a new backup copy of your configuration files.

On the Communications Manager Configuration Definition window, select the Workstation Connection Type that matches your connection to the AS/400, select 5250 emulation, and then select **Configure...** If you are using a token-ring connection, you get to the 5250 Emulation through the Token-Ring window. See Figure 66 on page 56. In our case this window showed one of the MVS system names in the Partner LU name field. We had to place the cursor at the beginning of the field and delete all characters with the keyboard delete key before we could enter the network name of the AS/400 system. Enter #INTER in the Mode name field, or select it from the drop-down list. Enter the token-ring address in the LAN destination address field. You also have to decide on and then enter the number of terminal sessions you want. We wanted only one terminal session. Figure 66 on page 56 shows the window after we entered the values we received from the display commands on the AS/400 system (Figure 63 on page 52 and Figure 64 on page 53).

**5250 Emulation through Token-Ring**

Network ID: USIBMSN

Local node name: SNC4879I

Local node ID (hex): 05D C4879

Partner LU name: USIBMSN.SJAS400A

Mode name: #INTER

Host type information: ☒ AS/400 ☐ S/36

LAN destination address (hex): 400052047158

Number of terminal sessions: 1

OK Advanced... Cancel Help

Figure 66. AS/400 Host: 5250 Emulation through the Token-Ring Window

Select **Advanced...** and you get to the Communications Manager Profile List Sheet window. This window is shown in Figure 17 on page 14 but now 5250 emulation has been added to the action list.

When we defined the AS/400 to CM/2 in this way, Communications Manager Setup generated the LU alias name 5250PLU. If you want to change the alias name, select SNA features and **Configure...** on the Communications Manager Profile List Sheet. If your PWS and AS/400 are in different VTAM networks, you have to change the network ID of the AS/400 by selecting SNA features. Follow the description in Figure 25 on page 19 through Figure 27 on page 20 and change the network ID and/or the LU alias name. When you get back to the Communications Manager Profile List Sheet window, select 5250 emulation, and then select **Configure....** You get to the 5250 Emulation window. See Figure 67 on page 57. You will have one line for each terminal session. Since we chose one terminal session, there is only one line.

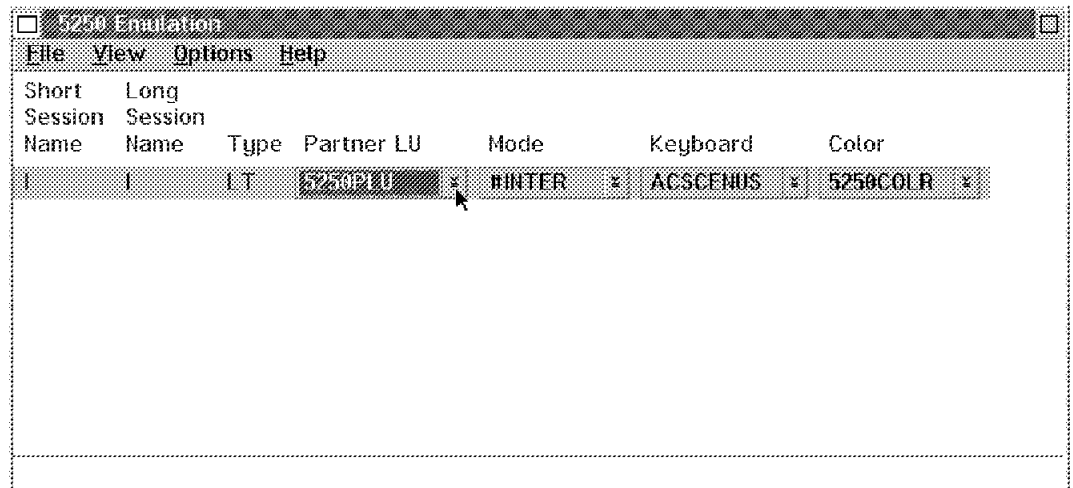


Figure 67. AS/400 Host: 5250 Emulation Window

This window shows 5250PLU as the partner LU alias name. If you changed the partner LU alias name earlier, you have to change it here as well. You can do that easily by selecting the alias name from the drop-down list as shown in Figure 68

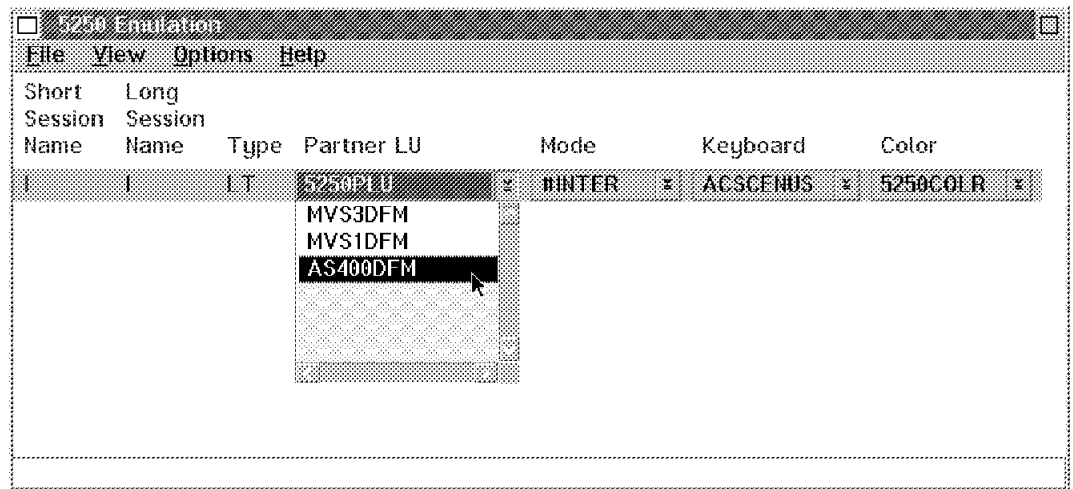


Figure 68. AS/400 Host: 5250 Emulation Window with Partner LU Drop-down List

**Note:** If you changed the local node alias name as described in Figure 17 on page 14 through Figure 19 on page 15, you must change it in the 5250 emulation. Select Options from the action bar on the 5250 Emulation window and then select Local LU alias name on the pull-down. The alias name must be the same as the name you entered on the Local Node Options window.

You have now defined the AS/400 system to CM/2. Close the 5250 Emulation window. Then exit Communications Manager Setup by selecting **Close** on each window until you see the Communications Manager - Checking Values window shown in Figure 30 on page 22. As always, when you make changes to Communications Manager Setup, you are asked whether you want to dynamically update your SNA resources.

The CM/2 configuration files are now changed. Figure 69 shows the NDF file from our workstation after we added the AS/400 to our CM/2 definitions.

```

DEFINE_LOCAL_CP  FQ_CP_NAME(USIBMSN.SNC4879I )
                  DESCRIPTION(Created on 07-05-94 AT 16:00)
                  CP_ALIAS(DFMRES )
                  NAU_ADDRESS(INDEPENDENT_LU)
                  NODE_TYPE(EN)
                  NODE_ID(X'05DC4879')
                  HOST_FP_SUPPORT(YES)
                  HOST_FP_LINK_NAME(LINK0002);

DEFINE_LOGICAL_LINK  LINK_NAME(LINK0005) 1
                    FQ_ADJACENT_CP_NAME(USIBMSN.SJAS400A )
                    ADJACENT_NODE_TYPE(LEN)
                    DLC_NAME(IBMTRNET)
                    ADAPTER_NUMBER(0)
                    DESTINATION_ADDRESS(X'400052047158')
                    CP_CP_SESSION_SUPPORT(NO)
                    ACTIVATE_AT_STARTUP(NO)
                    LIMITED_RESOURCE(USE_ADAPTER_DEFINITION)
                    LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
                    SOLICIT_SSCP_SESSION(NO)
                    EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
                    COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
                    COST_PER_BYTE(USE_ADAPTER_DEFINITION)
                    SECURITY(USE_ADAPTER_DEFINITION)
                    PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
                    USER_DEFINED_1(USE_ADAPTER_DEFINITION)
                    USER_DEFINED_2(USE_ADAPTER_DEFINITION)
                    USER_DEFINED_3(USE_ADAPTER_DEFINITION);

DEFINE_LOGICAL_LINK  LINK_NAME(LINK0002)
                    DESCRIPTION(Created on 07-05-95 at 04:35p)
                    FQ_ADJACENT_CP_NAME(USIBMSN.CP000003 )
                    ADJACENT_NODE_TYPE(LEN)
                    DLC_NAME(IBMTRNET)
                    ADAPTER_NUMBER(0)
                    DESTINATION_ADDRESS(X'40000ABC0016')
                    CP_CP_SESSION_SUPPORT(NO)
                    ACTIVATE_AT_STARTUP(NO)
                    LIMITED_RESOURCE(NO)
                    LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
                    SOLICIT_SSCP_SESSION(YES)
                    NODE_ID(X'05DC4879')
                    EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
                    COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
                    COST_PER_BYTE(USE_ADAPTER_DEFINITION)
                    SECURITY(USE_ADAPTER_DEFINITION)
                    PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
                    USER_DEFINED_1(USE_ADAPTER_DEFINITION)
                    USER_DEFINED_2(USE_ADAPTER_DEFINITION)
                    USER_DEFINED_3(USE_ADAPTER_DEFINITION);

```

Figure 69 (Part 1 of 3). AS/400 Host: CM/2 NDF with AS/400 Definitions

```

DEFINE_LOGICAL_LINK LINK_NAME(LINK0004)
                     ADJACENT_NODE_TYPE(LEN)
                     DLC_NAME(IBMTRNET)
                     ADAPTER_NUMBER(0)
                     DESTINATION_ADDRESS(X'40000ABC0200')
                     CP_SESSION_SUPPORT(NO)
                     ACTIVATE_AT_STARTUP(YES)
                     LIMITED_RESOURCE(USE_ADAPTER_DEFINITION)
                     LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
                     SOLICIT_SSCP_SESSION(YES)
                     PU_NAME(SJA2951 )
                     NODE_ID(X'05DA2051')
                     EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
                     COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
                     COST_PER_BYTE(USE_ADAPTER_DEFINITION)
                     SECURITY(USE_ADAPTER_DEFINITION)
                     PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_1(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_2(USE_ADAPTER_DEFINITION)
                     USER_DEFINED_3(USE_ADAPTER_DEFINITION);

DEFINE_PARTNER_LU FQ_PARTNER_LU_NAME(USIBMSJ.MVS3APPC )
PARTNER_LU_ALIAS(MVS3DFM)
PARTNER_LU_UNINTERPRETED_NAME(MVS3APPC)
MAX_MC_LL_SEND_SIZE(32767)
CONV_SECURITY_VERIFICATION(NO)
PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU FQ_PARTNER_LU_NAME(USIBMST.MVS1APPC )
PARTNER_LU_ALIAS(MVS1DFM)
PARTNER_LU_UNINTERPRETED_NAME(MVS1APPC)
MAX_MC_LL_SEND_SIZE(32767)
CONV_SECURITY_VERIFICATION(NO)
PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU FQ_PARTNER_LU_NAME(USIBMSN.SJAS400A ) 2
PARTNER_LU_ALIAS(AS400DFM)
MAX_MC_LL_SEND_SIZE(32767)
CONV_SECURITY_VERIFICATION(NO)
PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU_LOCATION FQ_PARTNER_LU_NAME(USIBMSJ.MVS3APPC )
WILDCARD_ENTRY(NO)
FQ_OWNING_CP_NAME(USIBMSN.CP000003 )
LOCAL_NODE_NN_SERVER(NO);

DEFINE_PARTNER_LU_LOCATION FQ_PARTNER_LU_NAME(USIBMSN.SJAS400A ) 3
WILDCARD_ENTRY(NO)
FQ_OWNING_CP_NAME(USIBMSN.SJAS400A )
LOCAL_NODE_NN_SERVER(YES);

DEFINE_PARTNER_LU_LOCATION FQ_PARTNER_LU_NAME(USIBMST.MVS1APPC )
WILDCARD_ENTRY(NO)
FQ_OWNING_CP_NAME(USIBMSN.CP000003 )
LOCAL_NODE_NN_SERVER(NO);

```

Figure 69 (Part 2 of 3). AS/400 Host: CM/2 NDF with AS/400 Definitions

```

DEFINE_MODE  MODE_NAME(QPCSUPP )
              DESCRIPTION(Created on 08-31-92 at 04:35p)
              COS_NAME(#CONNECT)
              DEFAULT_RU_SIZE(NO)
              MAX_RU_SIZE_UPPER_BOUND(1920)
              RECEIVE_PACING_WINDOW(7)
              MAX_NEGOTIABLE_SESSION_LIMIT(32767)
              PLU_MODE_SESSION_LIMIT(64)
              MIN_CONWINNERS_SOURCE(32);

DEFINE_DEFAULTS  IMPLICIT_INBOUND_PLU_SUPPORT(YES)
                  DESCRIPTION(Created on 08-31-92 at 04:35p)
                  DEFAULT_MODE_NAME(BLANK)
                  MAX_MC_LL_SEND_SIZE(32767)
                  DEFAULT_TP_OPERATION(NONQUEUED_AM_STARTED)
                  DEFAULT_TP_PROGRAM_TYPE(BACKGROUND)
                  DEFAULT_TP_CONV_SECURITY_RQD(NO)
                  MAX_HELD_ALERTS(10);

START_ATTACH_MANAGER;

```

Figure 69 (Part 3 of 3). AS/400 Host: CM/2 NDF with AS/400 Definitions

**Notes:**

- 1** Definition of the link to the AS/400.
- 2** Definition of the AS/400 LU name.
- 3** Definition of the AS/400 location in the network.

### 3.2.2 DFM/2 Definitions

You define the AS/400 system to DFM/2 by adding a DFM\_TARGET keyword to the CONFIG.DFM file. Figure 70 shows the DFM\_TARGET keyword that we added.

```

; Definition of an AS/400 system as target:
DFM_TARGET (
    remote_lu(AS400DFM)
    description(SJAS400A on the TR)
    conversation(HOLD)
    max_send_limit(4096)
    userid (PJRES1)
)

```

Figure 70. AS/400 Host: DFM\_TARGET Added for AS/400

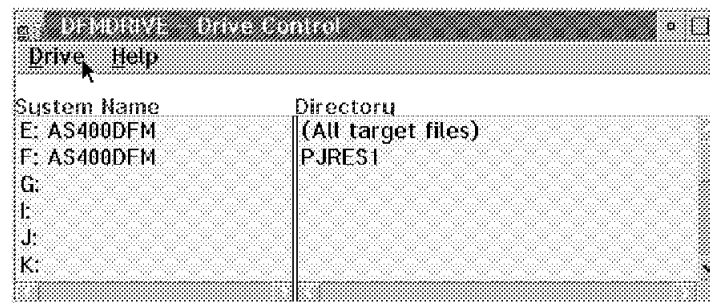
---

### 3.3 Record Level I/O Sample

In this section we explain how record level I/O is used when the data files reside on an AS/400 machine. We used the same C application that we used when the data files were on an MVS/ESA system.

In the first scenario that we ran we did not want to use conversion. Conversion is selected through the FILE\_DESCRIPTOR\_MAP keywords. The ADL check boxes are used only to select the record size when the files are created. With conversion the record sizes are larger. When you check the ADL boxes, the files are created with a larger record size. On the AS/400 there must be a correspondence between the ADL boxes and the FILE\_DESCRIPTOR\_MAPs. If they do not match, you will get a “Record Length Mismatch” message from the sample application when you try to load the data sets. By not selecting an ADL option and using data set names that are not mapped by the FILE\_DESCRIPTOR\_MAP keywords we opted not to have any translation or conversion of the data files on the AS/400 system.

Before we started the application, we assigned two drives to the AS/400 system as shown in Figure 71. We assigned the E drive to the libraries that are accessible to our user ID on the AS/400 and the F drive to one folder on the AS/400 system (the folder we used for our stream I/O demonstration).



---

Figure 71. AS/400 Host: DFMDRIVE - Drive Control Window

We started the C application and created the data files on the AS/400 system. Note that we placed all of the files on the E drive, which is now associated with the AS/400 system, and we did not check the ADL option for any of the files. Figure 72 on page 62 shows the creation of the data files on the AS/400 system.

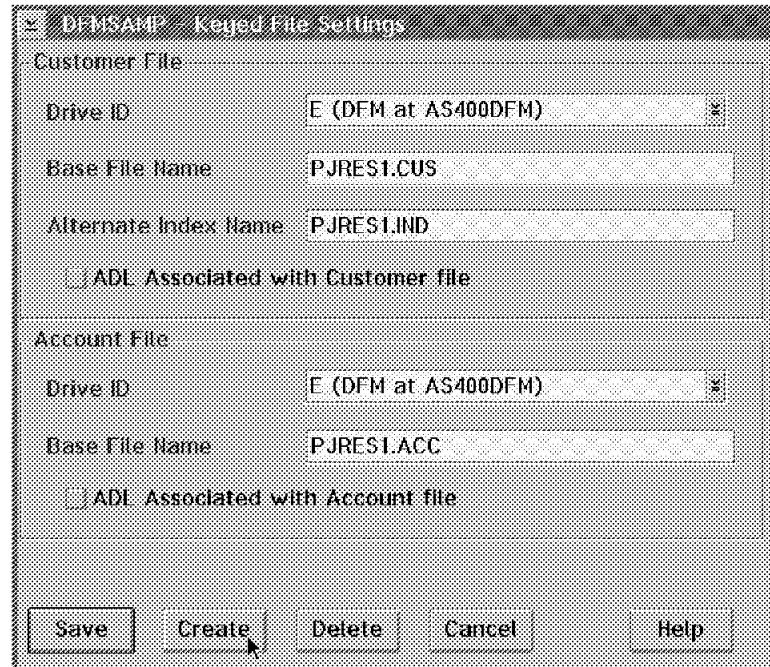


Figure 72. AS/400 Host: DFMSAMP - Keyed File Settings

After we created and loaded the data files from our OS/2 application, we could see those files on the AS/400 system. Figure 73 shows the files that were allocated on the AS/400 system.

Work with Files

Type options, press Enter.

1=Create 3=Copy 4=Delete 5=Display physical file member  
8=Display file description 9=Save 10=Restore 13=Change description

Opt	File	Library	Attribute	Text
5	PJRES1.ACC	QGPL	PF	
	PJRES1.CUS	QGPL	PF	
	PJRES1.IND	QGPL	LF	

Bottom

Parameters for options 1, 3, 4, 5, 8, 9, 10 and 13 or command  
===>

F3=Exit F4=Prompt F5=Refresh F9=Retrieve F11=Display names only  
F12=Cancel F16=Repeat position to F17=Position to

Figure 73. AS/400 Host: Work with Files Panel

On the next AS/400 panel we could see the records in the customer file. Because we did not use any translation, the records are stored in ASCII format



on the AS/400 system. Figure 74 shows the customer file in character and hexadecimal format.

Display Physical File Member
File . . . . . : PJRES1.CUS Library . . . . . : QGPL
Member . . . . . : PJRES1.CUS Record . . . . . : 1
Control . . . . . Column . . . . . : 1
Find . . . . .
\*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...
. + - ? ]? > ? % ?/
- > ? % ?/
< |+ - />
+ - > / ? % ?/
& < - ¢ ¢? / + ? % ?/
&| - /% ? % ?/
\*\*\*\*\* END OF DATA \*\*\*\*\*
F3=Exit F12=Cancel F19=Left F20=Right F24=More keys

Display Physical File Member
File . . . . . : PJRES1.CUS Library . . . . . : QGPL
Member . . . . . : PJRES1.CUS Record . . . . . : 1
Control . . . . . Column . . . . . : 1
Find . . . . .
\* . . . . + . . . . . 1 . . . . + . . . . . 2 . . . . + . . . . . 3 . . . . +
31313131 31314B41 56494E47 20202020 20202020 20202020 2020CA07 09000200 0E0029
31313131 31325349 53544120 20202020 20202020 20202020 2020CA07 09000200 0E0029
31313131 31334C41 52534F4E 20202020 20202020 20202020 2020CA07 09000200 0E0029
31313131 31344245 4852454E 53202020 20202020 20202020 2020CA07 09000200 0E0029
31313131 31355045 4C454720 20202020 20202020 20202020 2020CA07 09000200 0E0029
31313131 3136504F 47474920 20202020 20202020 20202020 2020CA07 09000200 0E0029
\*\*\*\*\* END OF DATA \*\*\*\*\*
F10=Display character F11=Display over/under F24=More keys

Figure 74. AS/400 Host: Display Physical File Member (Character and Hex Format)

After we created the files we started the application process. On the AS/400 system we could access the data sets by either the primary key, which is the customer number, or the alternate key, which is the last name. This process is demonstrated in Figure 75 on page 64 through Figure 78 on page 65.

In Figure 75 on page 64 you can see our first retrieval operation. We asked for a list of all primary keys; the output was a list of six customer numbers.

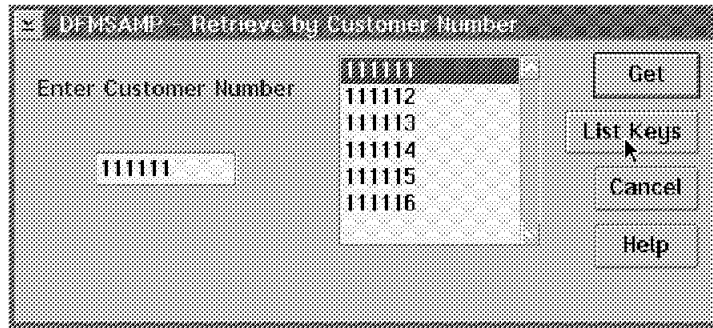


Figure 75. AS/400 Host: DFMSAMP - Retrieve by Customer Number Input Window

We selected customer number 111111 and clicked on **Get**. Figure 76 shows the output for this selection.

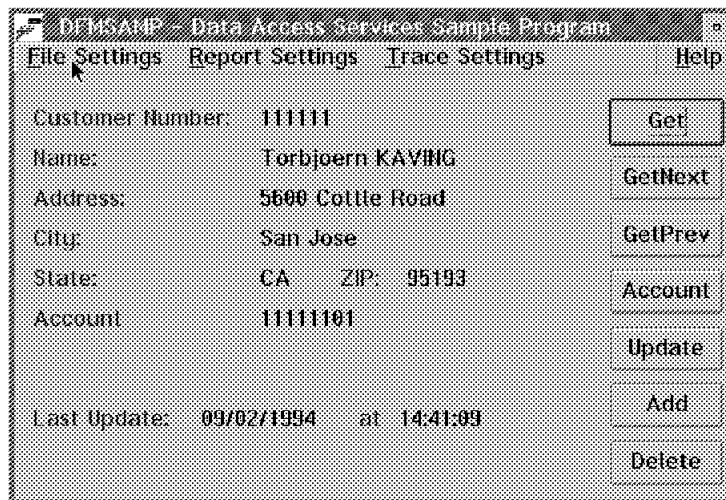


Figure 76. AS/400 Host: Record Retrieved by Customer Number

We then changed from retrieval by customer number to retrieval by customer's last name. To switch retrieval mode you need to select File Settings from the action bar on the DFMSAMP - Data Access Services Sample Program window and make your new selection. Using the alternate key we asked for the customer record with the key of SISTA (see Figure 77).

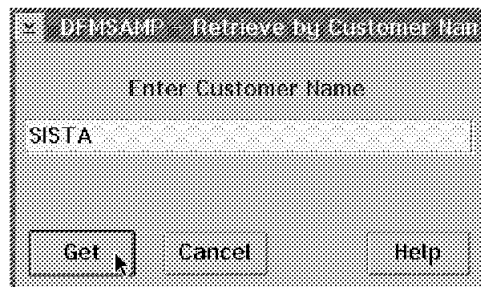


Figure 77. AS/400 Host: Get Record by an Alternate Key

Figure 78 shows the output for this request.

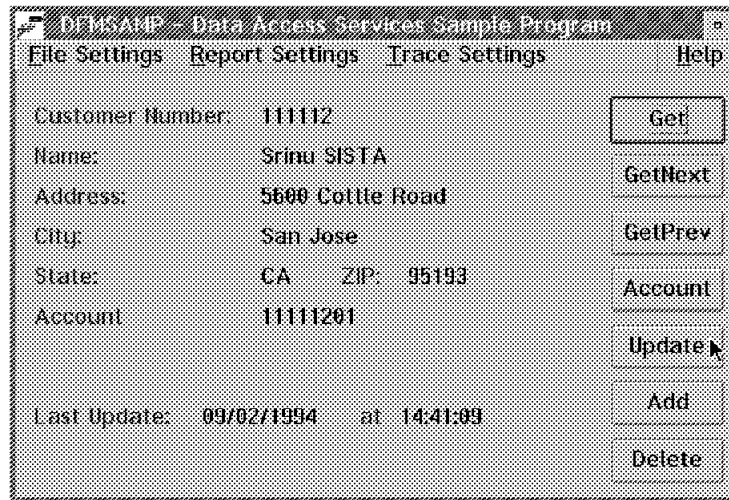


Figure 78. AS/400 Host: Record Retrieved by Customer's Last Name

To end the application we closed the files and the application's window.

In the second scenario we used AS/400 system data sets and converted the data when it was retrieved by the application. We added the FILE\_DESCRIPTOR\_MAP keywords shown in Figure 79 to the CONFIG.DFM file. We used the wildcard character, \*, as in the previous samples. We also added a FILE\_DESCRIPTOR\_MAP keyword for the alternate index file.

```
; DFMSAMP ADL for customer master file base cluster for AS400DFM
FILE_DESCRIPTOR_MAP(
    REMOTE_LU(AS400DFM)
    TARGET_FILENAME(*.CST)
    BASE_DDF(D:\DFMSAMP\DFMRCLCB.DDF)
    VIEW_DDF(D:\DFMSAMP\DFMRCLCV.DDF)
)

; DFMSAMP ADL for alternate index file base cluster for AS400DFM
FILE_DESCRIPTOR_MAP(
    REMOTE_LU(AS400DFM)
    TARGET_FILENAME(*.AIX)
    BASE_DDF(D:\DFMSAMP\DFMRCLCB.DDF)
    VIEW_DDF(D:\DFMSAMP\DFMRCLCV.DDF)
)

; DFMSAMP ADL for account detail file base cluster for AS400DFM
FILE_DESCRIPTOR_MAP(
    REMOTE_LU(AS400DFM)
    TARGET_FILENAME(*.ACT)
    BASE_DDF(D:\DFMSAMP\DFMRCLAB.DDF)
    VIEW_DDF(D:\DFMSAMP\DFMRCLAV.DDF)
)
```

Figure 79. AS/400 Host: FILE\_DESCRIPTOR\_MAP Keywords Added for AS/400

When we had the necessary pointers to the ADL files in our Distributed FileManager configuration file, we restarted Distributed FileManager by using the STRTDFMC command on the workstation, and restarted the sample application. This time we decided to use the ADL files. We also gave the data files names that DFM/2 maps as file names that are subject to data conversion (Figure 80).

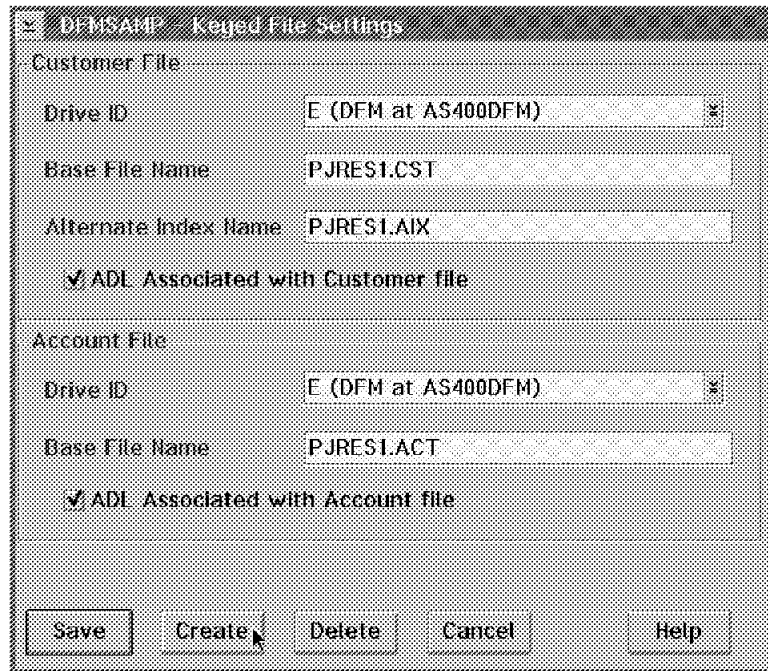


Figure 80. AS/400 Host: DFMSAMP - Keyed File Settings (with Data Conversion)

In Figure 81 on page 67 you can see the additional files that we created on the AS/400 system and a display of the records in the customer file when the data is converted to the AS/400 format.

Work with Files

Type options, press Enter.  
1=Create 3=Copy 4=Delete 5=Display physical file member  
8=Display file description 9=Save 10=Restore 13=Change description

Opt	File	Library	Attribute	Text
	PJRES1.ACC	QGPL	PF	
	PJRES1.ACT	QGPL	PF	
	PJRES1.AIX	QGPL	LF	
	PJRES1.CST	QGPL	PF	
	PJRES1.CUS	QGPL	PF	
	PJRES1.IND	QGPL	LF	

Bottom

Parameters for options 1, 3, 4, 5, 8, 9, 10 and 13 or command  
===>  
F3=Exit F4=Prompt F5=Refresh F9=Retrieve F11=Display names only  
F12=Cancel F16=Repeat position to F17=Position to

Display Physical File Member

File . . . . . : PJRES1.CST Library . . . . . : QGPL  
Member . . . . . : PJRES1.CST Record . . . . . : 1  
Control . . . . . Column . . . . . : 1  
Find . . . . .  
\*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...  
11111KAVING Torbjoern 5600 Cottle Road San Jos  
11112SISTA Srinu 5600 Cottle Road San Jos  
11113LARSON R. C. (Randy) 2000 West First Street Winston  
11114BEHRENS Cynthia 5600 Cottle Road San Jos  
11115PELEG J. (Joshua) 800 N. Frederick Avenue Gaither  
11116POGGI V. S. (Valerie) 5600 Cottle Road San Jos  
\*\*\*\*\* END OF DATA \*\*\*\*\*

F3=Exit F12=Cancel F19=Left F20=Right F24=More keys

Figure 81. AS/400 Host: Data Files on the AS/400 System (with Data Conversion )

In the third sample we created stream output and placed it in an AS/400 folder, not on the workstation’s local disk drive. This procedure is described in 3.4, “Stream I/O Sample.”

### 3.4 Stream I/O Sample

We also wanted to demonstrate how we could use the AS/400 file system as an extension to our local disk drive and save stream data in it.

In the first sample we used the same C application that we used for the record level I/O sample application. One of the options in the application is the writing

of a report record from the data files. To specify the data set name, select Report Settings from the action bar on the DFMSAMP - Data Access Services Sample Program window. We placed our report file on the F drive, which is actually a folder on the AS/400 system (see Figure 82).

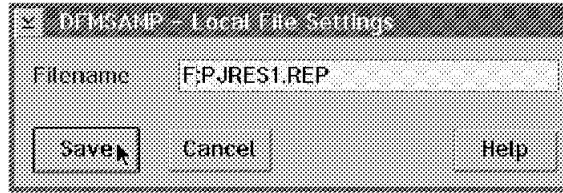


Figure 82. AS/400 Host: DFMSAMP - Local File Settings (Report File)

We wrote two records to the report file and then looked at the file from an OS/2 window and an AS/400 system terminal. Figure 83 shows how the report file looks from the OS/2 side.

**Note:** You can ignore the SYS0317 error message that is displayed when you execute the DIR command. The developers are aware of this problem and working on a fix.

```
[D:\DFMSAMP]dir f:

The volume label in drive F is PJRES1.
Directory of F:\

1-09-95  9:19p    <DIR>          0  .
1-09-95  9:19p    <DIR>          0  ..
1-10-95  1:53a    414           0  PJRES1.REP
SYS0317: The system cannot find message  0328
in message file OS0001.MSG.
          3 file(s)          414 bytes used
          2147482624 K bytes free

[D:\DFMSAMP]type f:pjres1.rep
Customer Number: 111111
Torbjoern KAVING
5600 Cottle Road
San Jose, CA 95193
Account Number: 11111101
The Kaving Corporate Account
5600 Cottle Road
San Jose, CA 95193
Balance: $5000.00

Customer Number: 111112
Srinu SISTA
5600 Cottle Road
San Jose, CA 95193
Account Number: 11111201
Sista Marketing Company
5600 Cottle Road
San Jose, CA 95193
Balance: $4000.00

[D:\DFMSAMP]
```

Figure 83. AS/400 Host: Stream File in an AS/400 Folder (OS/2 View)

The OS/2 file is stored in an AS/400 folder. When we looked at the file from an AS/400 terminal, we saw it as a document in a folder with a type value of PCFILE (see Figure 84 on page 69).

Work with Documents in Folders

Folder . . . PJRES1

Position to . . . . . Starting characters

Type options (and Document), press Enter.

7=Rename

3=Copy

4=Delete

Opt	Document	Document Description	Revised	Type
	PJRES1.REP	PJRES1.REP	01/10/95	PCFILE

F3=Exit

F4=Prompt

F5=Refresh

F11=Display names only

F12=Cancel

Bottom

F24=More keys

Figure 84. AS/400 Host: Stream File in an AS/400 Folder (AS/400 View)

In the second sample we used the AS/400 system to store OS/2 data of a standard commercial application. We used Distributed FileManager to store a presentation file for Freelance Graphics on the AS/400 system.

After we created a foil presentation we saved the presentation file on our F drive (this drive resides on the AS/400 system). Figure 85 on page 70 shows the presentation saved as an F:\DFM2.PRS file.

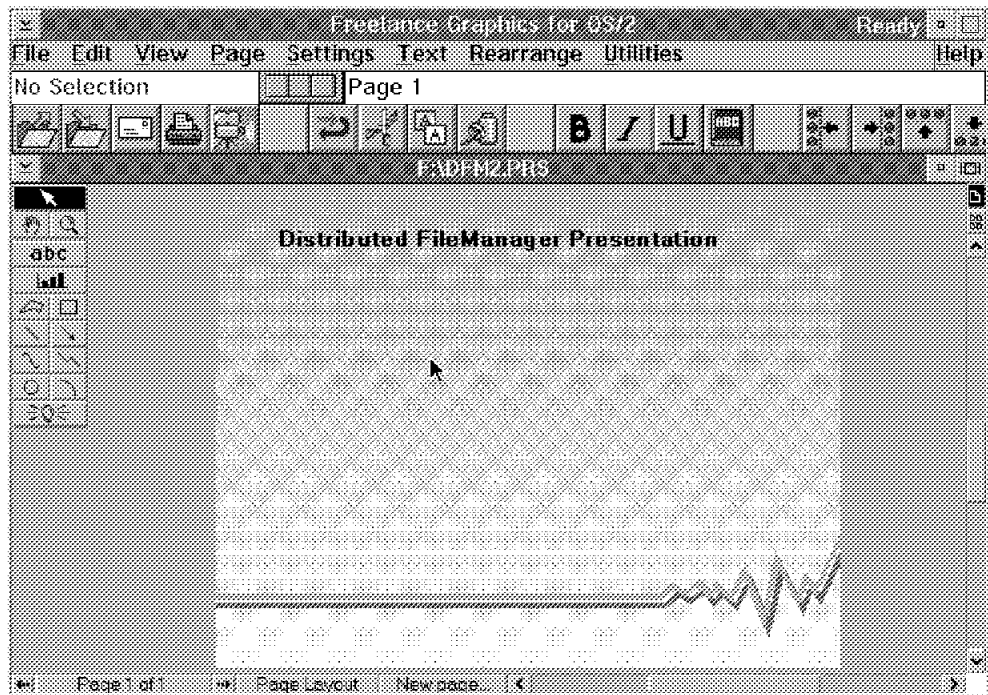


Figure 85. AS/400 Host: Freelance Graphics Using AS/400 Storage (OS/2 View)

On the AS/400 we could see that the new presentation file was added to the folder (Figure 86).

Work with Documents in Folders

Folder . . . PJRES1

Position to . . . . . Starting characters

Type options (and Document), press Enter.

7=Rename                      3=Copy                      4=Delete

Opt	Document	Document Description	Revised	Type
	DFM2.PRS	DFM2.PRS	01/10/95	PCFILE
	PJRES1.REP	PJRES1.REP	01/10/95	PCFILE

Bottom

F3=Exit      F4=Prompt      F5=Refresh
F11=Display names only      F12=Cancel
F24=More keys

Figure 86. AS/400 Host: Freelance Graphics Using AS/400 Storage (AS/400 View)

The logical drive in the AS/400 folder can be used for all standard stream I/O operations executed from the OS/2 workstation.



---

## Appendix A. Sample Diskette Description

The sample diskette accompanying this book contains the following files:

<b>CONFIG.BAS</b>	Sample configuration file for DFM/2, basic setup
<b>CONFIG.CPP</b>	Sample configuration file for DFM/2, including additions for the C sample application
<b>CONFIG.PLI</b>	Sample configuration file for DFM/2, including additions for the PL/I sample application
<b>CONFIG.DFM</b>	Sample configuration file for DFM/2, including additions for the AS/400 system
<b>\DFMSAMP</b>	A directory containing the files for the C sample application
<b>\PLISAMP</b>	A directory containing the files for the PL/I sample application



## Appendix B. Sample Network Definitions File for a Dialup Line

This appendix contains a sample CM/2 network definitions file (NDF) for connecting an OS/2 source system to two MVS target systems and one AS/400 target system using an SDLC dialup line.

```
DEFINE_LOCAL_CP FQ_CP_NAME(IBMINTSTDIL01 ) 1
                CP_ALIAS(STDIL01)
                NAU_ADDRESS(INDEPENDENT_LU)
                NODE_TYPE(EN)
                NODE_ID(X'05DABC01')
                NW_FP_SUPPORT(NONE)
                HOST_FP_SUPPORT(YES)
                MAX_COMP_LEVEL(NONE)
                MAX_COMP_TOKENS(0);

DEFINE_LOGICAL_LINK LINK_NAME(HOST0001) 2
                   FQ_ADJACENT_CP_NAME(USIBMSJ.MVS3APPC )
                   ADJACENT_NODE_TYPE(LEN)
                   DLC_NAME(SDLC )
                   ADAPTER_NUMBER(2)
                   DESTINATION_ADDRESS(X'FF')
                   OCDT_ENTRY_NAME(DASHOST_1)
                   CP_CP_SESSION_SUPPORT(NO)
                   SOLICIT_SSCP_SESSION(YES)
                   NODE_ID(X'05DABC01')
                   ACTIVATE_AT_STARTUP(YES)
                   USE_PUNAME_AS_CPNAME(NO)
                   INIT_WITH_SNRN(NO)
                   LIMITED_RESOURCE(USE_ADAPTER_DEFINITION)
                   LINK_STATION_ROLE(USE_ADAPTER_DEFINITION)
                   MAX_ACTIVATION_ATTEMPTS(USE_ADAPTER_DEFINITION)
                   EFFECTIVE_CAPACITY(USE_ADAPTER_DEFINITION)
                   COST_PER_CONNECT_TIME(USE_ADAPTER_DEFINITION)
                   COST_PER_BYTE(USE_ADAPTER_DEFINITION)
                   SECURITY(USE_ADAPTER_DEFINITION)
                   PROPAGATION_DELAY(USE_ADAPTER_DEFINITION)
                   USER_DEFINED_1(USE_ADAPTER_DEFINITION)
                   USER_DEFINED_2(USE_ADAPTER_DEFINITION)
                   USER_DEFINED_3(USE_ADAPTER_DEFINITION);

DEFINE_PARTNER_LU FQ_PARTNER_LU_NAME(USIBMSN.MVS1APPC ) 3
                 PARTNER_LU_ALIAS(MVS1DFM)
                 PARTNER_LU_UNINTERPRETED_NAME(MVS1APPC)
                 MAX_MC_LL_SEND_SIZE(32767)
                 CONV_SECURITY_VERIFICATION(NO)
                 PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU FQ_PARTNER_LU_NAME(USIBMSN.SJAS400A ) 3
                 PARTNER_LU_ALIAS(AS400DFM)
                 PARTNER_LU_UNINTERPRETED_NAME(SJAS400A)
                 MAX_MC_LL_SEND_SIZE(32767)
                 CONV_SECURITY_VERIFICATION(NO)
                 PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU FQ_PARTNER_LU_NAME(USIBMSJ.MVS3APPC ) 3
                 PARTNER_LU_ALIAS(MVS3DFM)
                 PARTNER_LU_UNINTERPRETED_NAME(MVS3APPC)
                 MAX_MC_LL_SEND_SIZE(32767)
                 CONV_SECURITY_VERIFICATION(NO)
                 PARALLEL_SESSION_SUPPORT(YES);

DEFINE_PARTNER_LU_LOCATION FQ_PARTNER_LU_NAME(USIBMSN.MVS1APPC ) 4
                          WILDCARD_ENTRY(NO)
                          FQ_OWNING_CP_NAME(USIBMSJ.MVS3APPC )
                          LOCAL_NODE_NN_SERVER(NO);

DEFINE_PARTNER_LU_LOCATION FQ_PARTNER_LU_NAME(USIBMSN.SJAS400A ) 4
                          WILDCARD_ENTRY(NO)
                          FQ_OWNING_CP_NAME(USIBMSJ.MVS3APPC )
                          LOCAL_NODE_NN_SERVER(NO);

DEFINE_PARTNER_LU_LOCATION FQ_PARTNER_LU_NAME(USIBMSJ.MVS3APPC ) 4
                          WILDCARD_ENTRY(NO)
                          FQ_OWNING_CP_NAME(USIBMSJ.MVS3APPC )
                          LOCAL_NODE_NN_SERVER(NO);

DEFINE_MODE MODE_NAME(QPCSUPP ) 5
            COS_NAME(#CONNECT)
```

```

        DEFAULT_RU_SIZE(NO)
        MAX_RU_SIZE_UPPER_BOUND(1920)
        RECEIVE_PACING_WINDOW(7)
        MAX_NEGOTIABLE_SESSION_LIMIT(32767)
        PLU_MODE_SESSION_LIMIT(64)
        MIN_CONWINNERS_SOURCE(32)
        COMPRESSION_NEED(PROHIBITED)
        PLU_SLU_COMPRESSION(NONE)
        SLU_PLU_COMPRESSION(NONE);

DEFINE_DEFAULTS  IMPLICIT_INBOUND_PLU_SUPPORT(YES)
                  DEFAULT_MODE_NAME(BLANK)
                  MAX_MC_LL_SEND_SIZE(32767)
                  DIRECTORY_FOR_INBOUND_ATTACHES(*)
                  DEFAULT_TP_OPERATION(NONQUEUED_AM_STARTED)
                  DEFAULT_TP_PROGRAM_TYPE(BACKGROUND)
                  DEFAULT_TP_CONV_SECURITY_RQD(NO)
                  MAX_HELD_ALERTS(10);

START_ATTACH_MANAGER;

```

**Notes:**

- 1** Definition of the local workstation.
- 2** Definition of the SDLC link to the focal-point MVS system.
- 3** Definition of the LU names of the target systems.
- 4** Definition of the location of the target systems.
- 5** Definition of the communication mode that DAS uses.

---

## Appendix C. C Sample Application Listings

This appendix contains the following listings from our implementation of the C sample application:

- Source code of the C program that issues the DDM calls  
All DDM calls are marked with <<<< in the right-hand margin.
- ADL for host view of customer master record
- ADL for application view of customer master record
- ADL for host view of account details record
- ADL for application view of account details record
- LISTCAT output for customer master file
- LISTCAT output for account details file
- Customer master file with data conversion
- Account details file with data conversion
- Customer master file without data conversion
- Account details file without data conversion.

The program was written by Randy Larson, IBM US Marketing and Support.

## C.1 C Sample Application Source Code

```
/*-----*/
/* Module: DFMTHRD.C */
/* (C) Copyright IBM Corporation 1994. All Rights reserved. */
/* */
/* Contents: All DDM Calls for the DAS Sample Application */
/* DFMWorker - thread which makes all the DDM calls */
/* Move_Cust_Buffer - moves customer record to I/O area */
/* Move_Account_buffer - moves account record to I/O area */
/* Move_Cust_Record - moves customer record from I/O area */
/* Move_Account_Record - move details recrod from I/O area */
/* Find_OS2_Error - formats error msg from OS/2 error code */
/* QueryDDMCount - get count of records in keyed file */
/*-----*/
#define INCL_BASE
#define INCL_WIN
#define INCL_DOSPROCESS
#define INCL_DOSFILEMGR /* File Manager values */
#include <os2.h>
#include <string.h>
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <ctype.h>
#include "dub.h"

#include "DFMRECRD.H"
#include "DFMSAMP.H"
#include "DFMHELP.H"

extern HDDMFILE FileHandle;
extern HDDMFILE FileHandleAlt;
extern HDDMFILE FileHandleAcct;

void _Optlink DFMWorker(PVOID pmp)
/* ***** */
/* This thread waits for the application semaphore to be posted */
/* it then checks pmp->DFM_Function for the requested code */
/* then loops to wait on the semaphore again */
/* when a shutdown request is received it exits the loop and closes */
/* the thread down */
/* ***** */
{
    BOOL ShutDown = FALSE;
    ULONG posted = 0;
    ULONG ulListCount;
    ULONG ulListSize;
    ULONG RecCount = 0;
    ULONG RecRtnCnt = 0;
    ULONG ulRecordLen;
    ULONG DDM_Flags = 0;
    ULONG FileInfoLevel = 1;
    ULONG ulKeyLen = 0;
    PVOID pCustList;
    DATETIME FunctionTime;
    MYKEYDEFBUF KeyDefBuf;
    HDDMFILE FileLoadHandle;
    HFILE Source_File;
    HFILE Target_File;
    BOOL end_of_file_flag = FALSE;
    CHAR szLoadBuffer[140] = "";
    ULONG ulStartTime;
    ULONG ulEndTime;
    ULONG byte_count;
    ULONG bytes_to_write;
    ULONG dos_rc;
    ULONG file_open_action = 0; /* open action return area */
    PMAIN_PARM pmp = pmp;
    CHAR szFullBaseName[259] = "";
    CHAR szFullIndexName[259] = "";
    CHAR szFullAcctName[259] = "";
    CHAR szTraceData[255] = "";
    FILESTATUS3 PathInfoBuf;
    ULONG PathInfoBufSize = sizeof(FILESTATUS3);
    ULONG ulRecCount;
    CHAR szEOF[2] = "\x1A";
    CHAR szDubErr[255] = "";
    CHAR szTempMsg[100] = "";
    TraceActivity(pmp, "DFMTHRD - Thread Initializing");
    while (ShutDown == FALSE)
    {

```

```

DosWaitEventSem(pmp->hevWorker, 0xFFFFFFFF);
sprintf(szTraceData,
        "DFMTHRD - Entry Function %d",
        pmp->DFM_Function);
TraceActivity(pmp, szTraceData);
pmp->DDM_rc = SC_NO_ERROR;
DosGetDateTime(&FunctionTime);
ulStartTime = FunctionTime.hundredths +
               (FunctionTime.seconds * 100) +
               (FunctionTime.minutes * 6000) +
               (FunctionTime.hours * 360000);
switch (pmp->DFM_Function)
{
case DFMSAMP_OPEN_CUST:
/* ***** */
/* Open Customer base file */
/* Process: */
/* - Build full name from Keyed file settings */
/* - Issue DDMOpen call */
/* - If open fails, set handle to NULLHANDLE */
/* ***** */
memset(szFullBaseName, '\0', sizeof(szFullBaseName));
memcpy(szFullBaseName, pmp->szKeyedCustDriveID, 1);
strcat(szFullBaseName, ".");
strcat(szFullBaseName, pmp->szKeyedBaseName);
sprintf(szTraceData, "DFMTRHD - Opening: %s", szFullBaseName);
TraceActivity(pmp, szTraceData);
strcpy(pmp->szDubErr, "DDMOpen Customer");
pmp->DDM_rc = DDMOpen(szFullBaseName,
                     &FileHandle,
                     CMBKEYAM,
                     DDM_GETAI | DDM_INSAI | DDM_MODAI | DDM_DELA,
                     DDM_UPDATERS,
                     NULL,
                     NULL);
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
if (pmp->DDM_rc != SC_NO_ERROR)
{
    FileHandle = NULLHANDLE;
}
break;

case DFMSAMP_OPEN_CUST_ALT:
/* ***** */
/* Open Customer alternate index file */
/* Process: */
/* - Build full name from Keyed file settings */
/* - Issue DDMOpen call */
/* - If open fails, set handle to NULLHANDLE */
/* ***** */
memset(szFullIndexName, '\0', sizeof(szFullIndexName));
memcpy(szFullIndexName, pmp->szKeyedCustDriveID, 1);
strcat(szFullIndexName, ".");
strcat(szFullIndexName, pmp->szKeyedIndexName);
sprintf(szTraceData, "DFMTRHD - Opening: %s", szFullIndexName);
TraceActivity(pmp, szTraceData);
strcpy(pmp->szDubErr, "DDMOpen Index");
pmp->DDM_rc = DDMOpen(szFullIndexName,
                     &FileHandleAlt,
                     CMBKEYAM,
                     DDM_GETAI | DDM_INSAI | DDM_MODAI | DDM_DELA,
                     DDM_UPDATERS,
                     NULL,
                     NULL);
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
if (pmp->DDM_rc != SC_NO_ERROR)
{
    FileHandleAlt = NULLHANDLE;
}
break;

case DFMSAMP_OPEN_ACCT:
/* ***** */
/* Open Account base file */
/* Process: */

```

```

/* - Build full name from Keyed file settings      */
/* - Issue DDMOpen call                          */
/* - If open fails, set handle to NULLHANDLE      */
/* ***** */
memset(szFullAcctName, '\0', sizeof(szFullAcctName));
memcpy(szFullAcctName, pmp->szKeyedAcctDriveID, 1);
strcat(szFullAcctName, ".");
strcat(szFullAcctName, pmp->szKeyedAcctName);
sprintf(szTraceData, "DFMTRHD - Opening: %s", szFullAcctName);
TraceActivity(pmp, szTraceData);
strcpy(pmp->szDubErr, "DDMOpen Account");
pmp->DDM_rc = DDMOpen(szFullAcctName,
                    &FileHandleAcct,
                    RNDKEYAM,
                    DDM_GETAI | DDM_INSAI | DDM_MODAI | DDM_DELAI,
                    DDM_UPDATERS,
                    NULL,
                    NULL);
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
if (pmp->DDM_rc != SC_NO_ERROR)
{
    FileHandleAcct = NULLHANDLE;
}
break;

case DFMSAMP_GET_CUST:
/* ***** */
/* Get a customer record from base file          */
/* Process:                                       */
/* - Build Key Value Buffer from requested key    */
/* - Issue DDMSetKey call using base handle      */
/* - If successful, move record contents to PMP   */
/* ***** */
pmp->KeyValBuf.cbKeyValBuf = (ULONG)sizeof(MYKEYVALNUM);
pmp->KeyValBuf.cpKeyValBuf = KEYVAL;
memcpy(pmp->KeyValBuf.szKeyVal,
        pmp->szGetKey,
        sizeof(pmp->KeyValBuf.szKeyVal));
strcpy(pmp->szDubErr, "DDMSetKey Customer");
pmp->DDM_rc = DDMSetKey(FileHandle,
                    (ULONG)0,
                    (PDDMOBJECT)&pmp->KeyValBuf,
                    KEYEQ,
                    (PDDMRECORD)pmp->pRecord,
                    (ULONG)pmp->RecordSize);
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
if (pmp->DDM_rc == SC_NO_ERROR) Move_Cust_Record(pmp);
break;

case DFMSAMP_GET_CUST_ALT:
/* ***** */
/* Get a customer record from alternate index file */
/* Process:                                       */
/* - Build Key Value Buffer from requested key    */
/* (consider length of name)                    */
/* - Issue DDMSetKey call using index handle      */
/* - If successful, move record contents to PMP   */
/* ***** */
pmp->KeyValBufN.cbKeyValBuf = (ULONG)sizeof(pmp->KeyValBufN.cbKeyValBuf) +
                    sizeof(pmp->KeyValBufN.cpKeyValBuf) +
                    strlen(pmp->szGetAltKey);
pmp->KeyValBufN.cpKeyValBuf = KEYVAL;
memcpy(pmp->KeyValBufN.szKeyVal,
        pmp->szGetAltKey,
        sizeof(pmp->KeyValBufN.szKeyVal));
strcpy(pmp->szDubErr, "DDMSetKey Index");
pmp->DDM_rc = DDMSetKey(FileHandleAlt,
                    (ULONG)0,
                    (PDDMOBJECT)&pmp->KeyValBufN,
                    KEYAE,
                    (PDDMRECORD)pmp->pRecord,
                    (ULONG)pmp->RecordSize);
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,

```



```

        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
if (pmp->DDM_rc == SC_NO_ERROR) Move_Cust_Record(pmp);
break;

case DFMSAMP_GET_ACCT:
/* ***** */
/* Get an account record */
/* Process: */
/* - Build Key Value Buffer from requested key */
/* - Issue DDMSetKey call using account handle */
/* - If successful, move record contents to PMP */
/* ***** */
pmp->KeyValBufA.cbKeyValBuf = (ULONG)sizeof(MYKEYVALACT);
pmp->KeyValBufA.cpKeyValBuf = KEYVAL;
memcpy(pmp->KeyValBufA.szKeyVal,
        pmp->szCustAccount,
        sizeof(pmp->KeyValBufA.szKeyVal));
strcpy(pmp->szDubErr, "DDMSetKey Account");
pmp->DDM_rc = DDMSetKey(FileHandleAcct,
        (ULONG)0,
        (PDDMOBJECT)&pmp->KeyValBufA,
        KEYEQ,
        (PDDMRECORD)pmp->pRecordA,
        (ULONG)pmp->RecordSizeA);
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
if (pmp->DDM_rc == SC_NO_ERROR) Move_Account_Record(pmp);
break;

case DFMSAMP_LIST_KEYS:
/* ***** */
/* Build a list of keys from customer base file */
/* Process: */
/* - Clear current key list array */
/* - There are two ways to continue based on entry */
/*   of a starting key */
/* - If no starting key is passed */
/* - Issue DDMSetKeyFirst to position at first */
/*   key in the file */
/* - Issue DDMSetKeyNext for 25 keys */
/* - If a starting key is passed */
/* - Issue DDMSetKeyFirst to position at that */
/*   key in the file using After or Equal strategy */
/* - If not found issue DDMSetKeyLast to position */
/*   at the last key in the file */
/* - Save the key we have found */
/* - Issue DDMSetKeyNext for 25 keys */
/* - Count the keys to position into array */
/* - Issue DDMSetKey to re-position at starting key */
/* - Issue DDMSetKeyPrevious for 25 keys */
/* - Count all the keys and save in PMP */
/* - Issue DDMSetKey to initial key so cursor remains */
/*   where it was when we started */
/* ***** */
/* First clear current list */
for (RecRtnCnt = 0; RecRtnCnt < 100; RecRtnCnt++)
{
    pmp->CustKeys->CustKey[RecRtnCnt].cpKeyValBuf = 0;
}
pCustList = &pmp->CustKeys->CustKey[0];
pmp->CustKeyCount = 0;
/* if no current customer number just position at first key */
/* and get it plus next KEYSEARCHCOUNT keys */
if (strlen(pmp->szGetKey) == 0)
{
    ulListSize = sizeof(MYKEYVALNUM);
    strcpy(pmp->szDubErr, "DDMSetKeyFirst Customer");
    pmp->DDM_rc = DDMSetKeyFirst(FileHandle,
        DDM_NODATA | DDM_KEYVALFB,
        (PDDMRECORD)pCustList,
        ulListSize);
    sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
    TraceActivity(pmp, szTraceData);
    if (pmp->DDM_rc != SC_NO_ERROR) break;
    pmp->CustKeyCount = 1;
    pCustList = &pmp->CustKeys->CustKey[1];
}

```

```

        ullistCount = KEYSEARCHCOUNT;
/* ***** */
/* DAS needs to provide space for the possible return of */
/* 2 RECAL objects - each contains RECAL and RECCNT */
/* RECAL has 4 byte length and 2 byte codepoint */
/* RECCNT has 4 byte len, 2 byte CP and 4 byte count */
/* for a total of 16 bytes each or 32 all together */
/* that is why the next line adds 32 to length */
/* See DOC APAR IC08180 */
/* ***** */
        ullistSize = (sizeof(MYKEYVALNUM) * ullistCount) + 32;
        strcpy(pmp->szDubErr,"DDMSetKeyNext Customer");
        pmp->DDM_rc = DDMSetKeyNext(FileHandle,
                                   DDM_NODATA | DDM_KEYVALFB,
                                   (PDDMRECORD)pCustList,
                                   ullistSize,
                                   ullistCount,
                                   &RecRtnCnt);
        sprintf(szTraceData,
                "DFMTRHD - %s RC = %d Record Count = %d",
                pmp->szDubErr,
                pmp->DDM_rc,
                RecRtnCnt);
        TraceActivity(pmp, szTraceData);
        pmp->CustKeyCount += RecRtnCnt;
    }
    else
/* if there is a requested customer number position at requested key */
/* get it plus next KEYSEARCHCOUNT keys then previous KEYSEARCHCOUNT keys */
    {
        ulKeyLen = strlen(pmp->szGetKey);
        if (ulKeyLen < 6)
        {
            memset(&pmp->szGetKey[ulKeyLen], '0', 6-ulKeyLen);
            memset(&pmp->szGetKey[6], '\0', 1);
        }
        pmp->KeyValBuf.cbKeyValBuf = (ULONG)sizeof(MYKEYVALNUM);
        pmp->KeyValBuf.cpKeyValBuf = KEYVAL;
        memcpy(pmp->KeyValBuf.szKeyVal,
               pmp->szGetKey,
               sizeof(pmp->KeyValBuf.szKeyVal));
        ullistSize = sizeof(MYKEYVALNUM);
        strcpy(pmp->szDubErr,"DDMSetKey Customer");
        pmp->DDM_rc = DDMSetKey(FileHandle,
                               DDM_NODATA | DDM_KEYVALFB,
                               (PDDMOBJECT)&pmp->KeyValBuf,
                               KEYAE,
                               (PDDMRECORD)pCustList,
                               ullistSize);
        sprintf(szTraceData,
                "DFMTRHD - %s RC = %d",
                pmp->szDubErr,
                pmp->DDM_rc);
        TraceActivity(pmp, szTraceData);
        if (pmp->DDM_rc != SC_NO_ERROR)
        {
            ullistSize = sizeof(MYKEYVALNUM);
            strcpy(pmp->szDubErr,"DDMSetKeyLast Customer");
            pmp->DDM_rc = DDMSetKeyLast(FileHandle,
                                       DDM_NODATA | DDM_KEYVALFB,
                                       (PDDMRECORD)pCustList,
                                       ullistSize);
            sprintf(szTraceData,
                    "DFMTRHD - %s RC = %d",
                    pmp->szDubErr,
                    pmp->DDM_rc);
            TraceActivity(pmp, szTraceData);
            if (pmp->DDM_rc != SC_NO_ERROR) break;
        }
/* save the key of the record we have found */
        memcpy(pmp->szGetKey, pmp->CustKeys->CustKey[0].szKeyVal, 6);
        pmp->CustKeyCount = 1;
        pCustList = &pmp->CustKeys->CustKey[1];
        ullistCount = KEYSEARCHCOUNT;
        ullistSize = (sizeof(MYKEYVALNUM) * ullistCount) + 32;
        strcpy(pmp->szDubErr,"DDMSetKeyNext Customer");
        pmp->DDM_rc = DDMSetKeyNext(FileHandle,
                                   DDM_NODATA | DDM_KEYVALFB,
                                   (PDDMRECORD)pCustList,
                                   ullistSize,
                                   ullistCount,
                                   &RecRtnCnt);
        sprintf(szTraceData,

```

```

        "DFMTRHD - %s RC = %d Record Count = %d",
        pmp->szDubErr,
        pmp->DDM_rc,
        RecRtnCnt);
TraceActivity(pmp, szTraceData);
if (pmp->DDM_rc != SC_NO_ERROR)
{
    pmp->CodePoint = ReplyMsg(szDubErr,
                             pmp->szDubMsg,
                             pmp->szSRVDGN);
    TraceActivity(pmp, pmp->szDubErr);
    TraceActivity(pmp, pmp->szDubMsg);
}
pmp->CustKeyCount += RecRtnCnt;
RecRtnCnt = 0;
while (pmp->CustKeys->CustKey[RecRtnCnt].cpKeyValBuf == KEYVAL)
{
    RecRtnCnt++;
}
pmp->KeyValBuf.cbKeyValBuf = (ULONG)sizeof(MYKEYVALNUM);
pmp->KeyValBuf.cpKeyValBuf = KEYVAL;
memcpy(pmp->KeyValBuf.szKeyVal,
        pmp->szGetKey,
        sizeof(pmp->KeyValBuf.szKeyVal));
strcpy(pmp->szDubErr, "DDMSetKey Customer");
pmp->DDM_rc = DDMSetKey(FileHandle,
                        DDM_NODATA,
                        (PDDMOBJECT)&pmp->KeyValBuf,
                        KEYEQ,
                        (PDDMRECORD)pCustList,
                        (ULONG)sizeof(CUSTKEYLIST));
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
if (pmp->DDM_rc != SC_NO_ERROR) break;
pCustList = &pmp->CustKeys->CustKey[RecRtnCnt];
ulListCount = KEYSEARCHCOUNT;
ulListSize = (sizeof(MYKEYVALNUM) * ulListCount) + 32;
strcpy(pmp->szDubErr, "DDMSetKeyPrevious Customer");
pmp->DDM_rc = DDMSetKeyPrevious(FileHandle,
                                DDM_NODATA | DDM_KEYVALFB,
                                (PDDMRECORD)pCustList,
                                ulListSize,
                                ulListCount,
                                &RecRtnCnt);
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d Record Count = %d",
        pmp->szDubErr,
        pmp->DDM_rc,
        RecRtnCnt);
TraceActivity(pmp, szTraceData);
pmp->CustKeyCount += RecRtnCnt;
}
pmp->CustKeyCount = 0;
while (pmp->CustKeys->CustKey[pmp->CustKeyCount].cpKeyValBuf == KEYVAL)
{
    pmp->CustKeyCount++;
}
/* now re-position cursor if there is a current record */
if (pmp->fRecordGotten == TRUE)
{
    pmp->KeyValBuf.cbKeyValBuf = (ULONG)sizeof(MYKEYVALNUM);
    pmp->KeyValBuf.cpKeyValBuf = KEYVAL;
    memcpy(pmp->KeyValBuf.szKeyVal,
          pmp->szCustNumber,
          sizeof(pmp->KeyValBuf.szKeyVal));
    strcpy(pmp->szDubErr, "DDMSetKey Customer");
    pmp->DDM_rc = DDMSetKey(FileHandle,
                            DDM_NODATA,
                            (PDDMOBJECT)&pmp->KeyValBuf,
                            KEYEQ,
                            (PDDMRECORD)pCustList,
                            (ULONG)sizeof(CUSTKEYLIST));
    sprintf(szTraceData,
            "DFMTRHD - %s RC = %d",
            pmp->szDubErr,
            pmp->DDM_rc);
    TraceActivity(pmp, szTraceData);
}
break;

```

```

case DFMSAMP_LIST_KEYS_ALT:
/* ***** */
/* Build a list of customer names from alternate index */
/* Note: Not implemented */
/* ***** */
strcpy(pmp->szDubErr, "DDMSetKeyFirst Index");
break;

case DFMSAMP_GETNEXT_CUST:
/* ***** */
/* Get next customer record from base file */
/* Process: */
/* - Issue DDMSetKeyNext using handle for base file */
/* ***** */
strcpy(pmp->szDubErr, "DDMSetKeyNext Customer");
pmp->DDM_rc = DDMSetKeyNext(FileHandle,
    (ULONG)0,
    (PDDMRECORD)pmp->pRecord,
    (ULONG)pmp->RecordSize,
    (ULONG)1,
    &RecRtnCnt);
sprintf(szTraceData,
    "DFMTRHD - %s RC = %d",
    pmp->szDubErr,
    pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
if (pmp->DDM_rc == SC_NO_ERROR) Move_Cust_Record(pmp);
break;

case DFMSAMP_GETNEXT_CUST_ALT:
/* ***** */
/* Get next customer record from alternate index file */
/* Process: */
/* - Issue DDMSetKeyNext using handle for alternate */
/* index file */
/* ***** */
strcpy(pmp->szDubErr, "DDMSetKeyNext Index");
pmp->DDM_rc = DDMSetKeyNext(FileHandleAlt,
    (ULONG)0,
    (PDDMRECORD)pmp->pRecord,
    (ULONG)pmp->RecordSize,
    (ULONG)1,
    &RecRtnCnt);
sprintf(szTraceData,
    "DFMTRHD - %s RC = %d",
    pmp->szDubErr,
    pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
if (pmp->DDM_rc == SC_NO_ERROR) Move_Cust_Record(pmp);
break;

case DFMSAMP_GETPREV_CUST:
/* ***** */
/* Get previous customer record from base file */
/* Process: */
/* - Issue DDMSetKeyPrevious using handle for base */
/* file */
/* ***** */
strcpy(pmp->szDubErr, "DDMSetKeyPrevious Customer");
pmp->DDM_rc = DDMSetKeyPrevious(FileHandle,
    (ULONG)0,
    (PDDMRECORD)pmp->pRecord,
    (ULONG)pmp->RecordSize,
    (ULONG)1,
    &RecRtnCnt);
sprintf(szTraceData,
    "DFMTRHD - %s RC = %d",
    pmp->szDubErr,
    pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
if (pmp->DDM_rc == SC_NO_ERROR) Move_Cust_Record(pmp);
break;

case DFMSAMP_GETPREV_CUST_ALT:
/* ***** */
/* Get next customer record from alternate index file */
/* Process: */
/* - Issue DDMSetKeyPrevious using handle for */
/* alternate index file */
/* ***** */
strcpy(pmp->szDubErr, "DDMSetKeyPrevious Index");
pmp->DDM_rc = DDMSetKeyPrevious(FileHandleAlt,
    (ULONG)0,

```

```

        (PDDMRECORD)pmp->pRecord,
        (ULONG)pmp->RecordSize,
        (ULONG)1,
        &RecRtnCnt);
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
if (pmp->DDM_rc == SC_NO_ERROR) Move_Cust_Record(pmp);
break;

case DFMSAMP_GET_CUST_UPDATE:
/* ***** */
/* Get customer record for update */
/* Process: */
/* - Build key value buffer from key of current */
/* customer record */
/* - Issue DDMSetKey using handle for base file and */
/* specifying update intent and return no data */
/* (We already have the record) */
/* ***** */
pmp->KeyValBuf.cbKeyValBuf = (ULONG)sizeof(MYKEYVALNUM);
pmp->KeyValBuf.cpKeyValBuf = KEYVAL;
memcpy(pmp->KeyValBuf.szKeyVal,
        pmp->szCustNumber,
        sizeof(pmp->KeyValBuf.szKeyVal));
strcpy(pmp->szDubErr, "DDMSetKey Customer");
pmp->DDM_rc = DDMSetKey(FileHandle,
        DDM_UPDINT | DDM_NODATA,
        (PDDMOBJECT)&pmp->KeyValBuf,
        KEYEQ,
        NULL,
        (ULONG)0);
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
break;

case DFMSAMP_GET_ACCT_UPDATE:
/* ***** */
/* Get account record for update */
/* Process: */
/* - Build key value buffer from key of current */
/* account record */
/* - Issue DDMSetKey using handle for account file */
/* specifying update intent and return no data */
/* (We already have the record) */
/* ***** */
pmp->KeyValBufA.cbKeyValBuf = (ULONG)sizeof(MYKEYVALACT);
pmp->KeyValBufA.cpKeyValBuf = KEYVAL;
memcpy(pmp->KeyValBufA.szKeyVal,
        pmp->szCustAccount,
        sizeof(pmp->KeyValBufA.szKeyVal));
strcpy(pmp->szDubErr, "DDMSetKey Account");
pmp->DDM_rc = DDMSetKey(FileHandleAcct,
        DDM_UPDINT | DDM_NODATA,
        (PDDMOBJECT)&pmp->KeyValBufA,
        KEYEQ,
        NULL,
        (ULONG)0);
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
break;

case DFMSAMP_MOD_CUST:
/* ***** */
/* Modify a customer record already gotten for update */
/* Process: */
/* - We have already gotten the current customer */
/* record with update intent */
/* - Issue DDMModifyRec using INHMODKY to changing */
/* the key of the record */
/* ***** */
strcpy(pmp->szDubErr, "DDMModifyRec Customer");
pmp->DDM_rc = DDMModifyRec(FileHandle,
        DDM_INHMODKY,
        (PDDMRECORD)pmp->pRecord);

```

```

sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
break;

case DFMSAMP_MOD_ACCT:
/* ***** */
/* Modify an account record already gotten for update */
/* Process: */
/* - We have already gotten the current account */
/* record with update intent */
/* - Issue DDMModifyRec using INHMODKY to changing */
/* the key of the record */
/* ***** */
strcpy(pmp->szDubErr, "DDMModifyRec Account");
pmp->DDM_rc = DDMModifyRec(FileHandleAcct,
                           DDM_INHMODKY,
                           (PDDMRECORD)pmp->pRecordA);
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
break;

case DFMSAMP_INSERT_CUST:
/* ***** */
/* Insert a new customer record */
/* - Issue DDMInsertRecKey to add record at pRecord */
/* ***** */
strcpy(pmp->szDubErr, "DDMInsertRecKey Customer");
pmp->DDM_rc = DDMInsertRecKey
              (FileHandle,
               (ULONG)0,
               pmp->pRecord,
               NULL,
               (ULONG)1
              );
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
break;

case DFMSAMP_INSERT_ACCT:
/* ***** */
/* Insert a new account record */
/* - Issue DDMInsertRecKey to add record at pRecordA */
/* ***** */
strcpy(pmp->szDubErr, "DDMInsertRecKey Account");
pmp->DDM_rc = DDMInsertRecKey
              (FileHandleAcct,
               (ULONG)0,
               pmp->pRecordA,
               NULL,
               (ULONG)1
              );
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
break;

case DFMSAMP_DELETE_CUST:
/* ***** */
/* Delete a customer record gotten for update */
/* Process: */
/* - We have already gotten the current customer */
/* record with update intent */
/* - Issue DDMDeleteRec to delete the currently held */
/* record */
/* ***** */
strcpy(pmp->szDubErr, "DDMDeleteRec Customer");
pmp->DDM_rc = DDMDeleteRec(FileHandle,
                           (ULONG)0);
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);

```

```

TraceActivity(pmp, szTraceData);
break;

case DFMSAMP_DELETE_ACCT:
/* ***** */
/* Delete an account record gotten for update */
/* Process: */
/* - We have already gotten the current account */
/* record with update intent */
/* - Issue DDMDeleteRec to delete the currently held */
/* record */
/* ***** */
strcpy(pmp->szDubErr,"DDMDeleteRec Account");
pmp->DDM_rc = DDMDeleteRec(FileHandleAcct,
                           (ULONG)0);
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
break;

case DFMSAMP_CLOSE_CUST:
/* ***** */
/* Close customer base file */
/* Process: */
/* - Check that file handle is not NULLHANDLE */
/* - Close the handle of the base customer file */
/* ***** */
strcpy(pmp->szDubErr,"DDMClose Customer");
if (FileHandle != NULLHANDLE)
{
    pmp->DDM_rc = DDMClose(FileHandle);
    sprintf(szTraceData,
            "DFMTRHD - %s RC = %d",
            pmp->szDubErr,
            pmp->DDM_rc);
    TraceActivity(pmp, szTraceData);
}
else
{
    sprintf(szTraceData,
            "DFMTRHD - Customer File Not Open");
    TraceActivity(pmp, szTraceData);
}
FileHandle = NULLHANDLE;
break;

case DFMSAMP_CLOSE_CUST_ALT:
/* ***** */
/* Close customer alternate index file */
/* Process: */
/* - Check that file handle is not NULLHANDLE */
/* - Close the handle of the alternate index file */
/* ***** */
strcpy(pmp->szDubErr,"DDMClose Index");
if (FileHandleAlt != NULLHANDLE)
{
    pmp->DDM_rc = DDMClose(FileHandleAlt);
    sprintf(szTraceData,
            "DFMTRHD - %s RC = %d",
            pmp->szDubErr,
            pmp->DDM_rc);
    TraceActivity(pmp, szTraceData);
}
else
{
    sprintf(szTraceData,
            "DFMTRHD - Index File Not Open");
    TraceActivity(pmp, szTraceData);
}
FileHandleAlt = NULLHANDLE;
break;

case DFMSAMP_CLOSE_ACCT:
/* ***** */
/* Close account base file */
/* Process: */
/* - Check that file handle is not NULLHANDLE */
/* - Close the handle of the base account file */
/* ***** */
strcpy(pmp->szDubErr,"DDMClose Account");
if (FileHandleAcct != NULLHANDLE)

```

```

    {
        pmp->DDM_rc = DDMClose(FileHandleAcct);
        sprintf(szTraceData,
            "DFMTRHD - %s RC = %d",
            pmp->szDubErr,
            pmp->DDM_rc);
        TraceActivity(pmp, szTraceData);
    }
    else
    {
        sprintf(szTraceData,
            "DFMTRHD - Account File Not Open");
        TraceActivity(pmp, szTraceData);
    }
    FileHandleAcct = NULLHANDLE;
    break;

case DFMSAMP_CUST_LOAD:
    /* ***** */
    /* Load the customer base file from a flat file */
    /* Process: */
    /* - Calculate number of source records */
    /* - Open the source file */
    /* - Get the first source record */
    /* - Get the second source record */
    /* - Issue DDMLoadFileFirst with chaining if the */
    /* record exists */
    /* - Loop reading source and issue DDMLoadFileNext */
    /* with no flags */
    /* - When end-of-file is reached on source file, */
    /* set flags to DDM_CLOSE to complete process */
    /* ***** */
    /* Build target file name */
    memset(szFullBaseName, '\0', sizeof(szFullBaseName));
    memcpy(szFullBaseName, pmp->szKeyedCustDriveID, 1);
    strcat(szFullBaseName, ".");
    strcat(szFullBaseName, pmp->szKeyedBaseName);
    /* calculate count of source records */
    DosQueryPathInfo(pmp->szLoadCustomerFileName,
        1,
        &PathInfoBuf,
        PathInfoBufSize);
    ulRecCount = PathInfoBuf.cbFile/CUST_SRC_SIZE;
    if (ulRecCount == 0) ulRecCount = 1;
    pmp->GetCount = 0;
    /* Open source file */
    dos_rc = DosOpen(pmp->szLoadCustomerFileName,
        &Source_File,
        &file_open_action,
        (ULONG) 0,
        FILE_NORMAL,
        OPEN_ACTION_OPEN_IF_EXISTS,
        OPEN_SHARE_DENYWRITE,
        0L);
    if (dos_rc != NO_ERROR)
    {
        strcpy(pmp->szDubErr, "OS/2 Open Error on Customer Source");
        Find_OS2_Error(pmp, dos_rc);
        pmp->GetCount = -1;
        break;
    }
    end_of_file_flag = FALSE;
    /* Read the first record */
    dos_rc = DosRead(Source_File,
        szLoadBuffer,
        CUST_SRC_SIZE,
        &byte_count);
    if (dos_rc != NO_ERROR)
    {
        strcpy(pmp->szDubErr, "OS/2 Read Error on Customer Source");
        Find_OS2_Error(pmp, dos_rc);
        pmp->GetCount = -1;
        break;
    }
    /* check for eof */
    if (byte_count == 0)
        end_of_file_flag = TRUE;
    if (memcmp(&szLoadBuffer[0], &szEOF[0], 1) == 0)
        end_of_file_flag = TRUE;
    /* build target record */
    Move_Cust_Buffer(pmp, szLoadBuffer);
    /* get the second record */
    dos_rc = DosRead(Source_File,

```



```

        szLoadBuffer,
        CUST_SRC_SIZE,
        &byte_count);
if (dos_rc != NO_ERROR)
{
    strcpy(pmp->szDubErr, "OS/2 Read Error on Customer Source");
    Find_OS2_Error(pmp, dos_rc);
    pmp->GetCount = -1;
    break;
}
if (byte_count == 0)
    end_of_file_flag = TRUE;
if (memcmp(&szLoadBuffer[0], &szEOF[0], 1) == 0)
    end_of_file_flag = TRUE;
if (end_of_file_flag == FALSE)
    DDM_Flags = DDM_CHAIN;
else
    DDM_Flags = 0;
strcpy(pmp->szDubErr, "DDMLoadFileFirst Customer");
pmp->DDM_rc = DDMLoadFileFirst
    (szFullBaseName,
     &FileLoadHandle,
     DDM_Flags,
     pmp->pRecord,
     1UL
    );
if (pmp->DDM_rc != SC_NO_ERROR)
{
    sprintf(szTraceData,
            "DFMTRHD - %s RC = %d",
            pmp->szDubErr,
            pmp->DDM_rc);
    TraceActivity(pmp, szTraceData);
}
else
{
    ++pmp->GetCount;
    WinPostMsg(pmp->hwndCaller,
               WM_USER_SET_TEXT,
               MPFROMLONG(u1RecCount),
               MPFROMLONG(pmp->GetCount));
}
/* now loop sending data to target system until end of file */
while ((end_of_file_flag == FALSE) && (pmp->DDM_rc == SC_NO_ERROR))
{
    Move_Cust_Buffer(pmp, szLoadBuffer);
    dos_rc = DosRead(Source_File,
                     szLoadBuffer,
                     CUST_SRC_SIZE,
                     &byte_count);
    if (dos_rc != NO_ERROR)
    {
        strcpy(pmp->szDubErr, "OS/2 Read Error on Customer Source");
        Find_OS2_Error(pmp, dos_rc);
        pmp->GetCount = -1;
        break;
    }
    if (byte_count == 0)
        end_of_file_flag = TRUE;
    if (memcmp(&szLoadBuffer[0], &szEOF[0], 1) == 0)
        end_of_file_flag = TRUE;
    if (end_of_file_flag == FALSE)
        DDM_Flags = 0;
    else
        DDM_Flags = DDM_CLOSE;
/* Update counter prior to DDMLoad so slider will show full when done */
    ++pmp->GetCount;
    WinPostMsg(pmp->hwndCaller,
               WM_USER_SET_TEXT,
               MPFROMLONG(u1RecCount),
               MPFROMLONG(pmp->GetCount));
    strcpy(pmp->szDubErr, "DDMLoadFileNext Customer");
    pmp->DDM_rc = DDMLoadFileNext
        (FileLoadHandle,
         DDM_Flags,
         pmp->pRecord,
         1UL
        );
    if (pmp->DDM_rc != SC_NO_ERROR)
    {
        sprintf(szTraceData,
                "DFMTRHD - %s RC = %d",
                pmp->szDubErr,

```

```

        pmp->DDM_rc);
        TraceActivity(pmp, szTraceData);
    }
}
dos_rc = DosClose(Source_File);
break;

case DFMSAMP_ACCT_LOAD:
/* ***** */
/* Load the account base file from a flat file */
/* Process: */
/* - Calculate number of source records */
/* - Open the source file */
/* - Get the first source record */
/* - Get the second source record */
/* - Issue DDMLoadFileFirst with chaining if the */
/*   record exists */
/* - Loop reading source and issue DDMLoadFileNext */
/*   with no flags */
/* - When end-of-file is reached on source file, */
/*   set flags to DDM_CLOSE to complete process */
/* ***** */
memset(szFullAcctName, '\0', sizeof(szFullAcctName));
memcpy(szFullAcctName, pmp->szKeyedAcctDriveID, 1);
strcat(szFullAcctName, ".");
strcat(szFullAcctName, pmp->szKeyedAcctName);
DosQueryPathInfo(pmp->szLoadAccountFileName,
    1,
    &PathInfoBuf,
    PathInfoBufSize);
ulRecCount = PathInfoBuf.cbFile/ACCT_SRC_SIZE;
if (ulRecCount == 0) ulRecCount = 1;
pmp->GetCount = 0;
dos_rc = DosOpen(pmp->szLoadAccountFileName,
    &Source_File,
    &file_open_action,
    (ULONG) 0,
    FILE_NORMAL,
    OPEN_ACTION_OPEN_IF_EXISTS,
    OPEN_SHARE_DENYWRITE,
    0L);
if (dos_rc != NO_ERROR)
{
    strcpy(pmp->szDubErr, "OS/2 Open Error on Account Source");
    Find_OS2_Error(pmp, dos_rc);
    pmp->GetCount = -1;
    break;
}
end_of_file_flag = FALSE;
dos_rc = DosRead(Source_File,
    szLoadBuffer,
    ACCT_SRC_SIZE,
    &byte_count);
if (dos_rc != NO_ERROR)
{
    strcpy(pmp->szDubErr, "OS/2 Read Error on Account Source");
    Find_OS2_Error(pmp, dos_rc);
    pmp->GetCount = -1;
    break;
}
if (byte_count == 0) end_of_file_flag = TRUE;
if (memcmp(&szLoadBuffer[0], &szEOF[0], 1) == 0) end_of_file_flag = TRUE;
Move_Account_Buffer(pmp, szLoadBuffer);
dos_rc = DosRead(Source_File,
    szLoadBuffer,
    ACCT_SRC_SIZE,
    &byte_count);
if (dos_rc != NO_ERROR)
{
    strcpy(pmp->szDubErr, "OS/2 Read Error on Account Source");
    Find_OS2_Error(pmp, dos_rc);
    pmp->GetCount = -1;
    break;
}
if (byte_count == 0) end_of_file_flag = TRUE;
if (memcmp(&szLoadBuffer[0], &szEOF[0], 1) == 0) end_of_file_flag = TRUE;
if (end_of_file_flag == FALSE)
    DDM_Flags = DDM_CHAIN;
else
    DDM_Flags = 0;
strcpy(pmp->szDubErr, "DDMLoadFileFirst Account");
pmp->DDM_rc = DDMLoadFileFirst
    (szFullAcctName,

```

<<<<

```

        &FileLoadHandle,
        DDM_Flags,
        pmp->pRecordA,
        1UL
    );
    if (pmp->DDM_rc != SC_NO_ERROR)
    {
        sprintf(szTraceData,
            "DFMTRHD - %s RC = %d",
            pmp->szDubErr,
            pmp->DDM_rc);
        TraceActivity(pmp, szTraceData);
    }
    else
    {
        ++pmp->GetCount;
        WinPostMsg(pmp->hwndCaller,
            WM_USER_SET_TEXT,
            MPFROMLONG(u1RecCount),
            MPFROMLONG(pmp->GetCount));
    }
    while ((end_of_file_flag == FALSE) && (pmp->DDM_rc == SC_NO_ERROR))
    {
        Move_Account_Buffer(pmp, szLoadBuffer);
        dos_rc = DosRead(Source_File,
            szLoadBuffer,
            ACCT_SRC_SIZE,
            &byte_count);
        if (dos_rc != NO_ERROR)
        {
            strcpy(pmp->szDubErr, "OS/2 Read Error on Account Source");
            Find_OS2_Error(pmp, dos_rc);
            pmp->GetCount = -1;
            break;
        }
        if (byte_count == 0)
            end_of_file_flag = TRUE;
        if (memcmp(&szLoadBuffer[0], &szEOF[0], 1) == 0)
            end_of_file_flag = TRUE;
        if (end_of_file_flag == FALSE)
            DDM_Flags = 0;
        else
            DDM_Flags = DDM_CLOSE;
        /* Update counter prior to DDMLoad so slider will show full when done */
        ++pmp->GetCount;
        WinPostMsg(pmp->hwndCaller,
            WM_USER_SET_TEXT,
            MPFROMLONG(u1RecCount),
            MPFROMLONG(pmp->GetCount));
        strcpy(pmp->szDubErr, "DDMLoadFileNext Account");
        pmp->DDM_rc = DDMLoadFileNext
            (FileLoadHandle,
             DDM_Flags,
             pmp->pRecordA,
             1UL
            );
        if (pmp->DDM_rc != SC_NO_ERROR)
        {
            sprintf(szTraceData,
                "DFMTRHD - %s RC = %d",
                pmp->szDubErr,
                pmp->DDM_rc);
            TraceActivity(pmp, szTraceData);
        }
    }
    dos_rc = DosClose(Source_File);
    break;

case DFMSAMP_CUST_UNLOAD:
    /* ***** */
    /* Unload the customer base file into a flat file */
    /* Process: */
    /* - Calculate number of source records */
    /* (see routine QueryDDMCount) */
    /* - Open the target file */
    /* - Issue DDMUnLoadFileFirst in key sequence */
    /* - Write first target record */
    /* - Loop using DDMUnLoadFileNext if MORE_DATA flag */
    /* is returned on previous call then write next */
    /* target record */
    /* - Close target file */
    /* ***** */
    memset(szFullBaseName, '\0', sizeof(szFullBaseName));

```

```

memcpy(szFullBaseName,pmp->szKeyedCustDriveID,1);
strcat(szFullBaseName,".");
strcat(szFullBaseName,pmp->szKeyedBaseName);
ulRecCount = QueryDDMCount(pmp, szFullBaseName);
pmp->GetCount = 0;
dos_rc = DosOpen(pmp->szUnloadCustomerFileName,
                 &Target_File,
                 &file_open_action,
                 (ULONG) 0,
                 FILE_NORMAL,
                 OPEN_ACTION_REPLACE_IF_EXISTS |
                 OPEN_ACTION_CREATE_IF_NEW,
                 OPEN_ACCESS_WRITEONLY |
                 OPEN_SHARE_DENYWRITE |
                 OPEN_FLAGS_SEQUENTIAL,
                 0L);
if (dos_rc != NO_ERROR)
{
    strcpy(pmp->szDubErr,"OS/2 Open Error on Customer Target");
    Find_OS2_Error(pmp, dos_rc);
    pmp->GetCount = -1;
    break;
}
strcpy(pmp->szDubErr,"DDMUnLoadFileFirst Customer");
pmp->DDM_rc = DDMUnLoadFileFirst(szFullBaseName,
                                &FileLoadHandle,
                                (ULONG)0,
                                &DDM_Flags,
                                (PDDMRECORD)pmp->pRecord,
                                (ULONG)pmp->RecordSize,
                                (CODEPOINT)KEYORD,
                                &RecCount);
if (pmp->DDM_rc != SC_NO_ERROR)
{
    sprintf(szTraceData,
            "DFMTRHD - %s RC = %d",
            pmp->szDubErr,
            pmp->DDM_rc);
    TraceActivity(pmp, szTraceData);
}
else
{
    Move_Cust_Record(pmp);
    sprintf(szLoadBuffer,
            "%6s%-20s%-20s%-25s%-25s%-2s%-5s%-8s\r\n",
            pmp->szCustNumber,
            pmp->szCustLName,
            pmp->szCustFName,
            pmp->szCustAddress,
            pmp->szCustCity,
            pmp->szCustState,
            pmp->szCustZip,
            pmp->szCustAccount);
    bytes_to_write = strlen(szLoadBuffer);
    dos_rc = DosWrite(Target_File,
                     szLoadBuffer,
                     bytes_to_write,
                     &byte_count);
    if (dos_rc != NO_ERROR)
    {
        strcpy(pmp->szDubErr,"OS/2 Write Error on Customer Target");
        Find_OS2_Error(pmp, dos_rc);
        pmp->GetCount = -1;
        break;
    }
    pmp->GetCount = 1;
    WinPostMsg(pmp->hwndCaller,
               WM_USER_SET_TEXT,
               MPFROMLONG(ulRecCount),
               MPFROMLONG(pmp->GetCount));
}
while ((pmp->DDM_rc == SC_NO_ERROR) && (DDM_Flags == DDM_MOREDATA))
{
    strcpy(pmp->szDubErr,"DDMUnLoadFileNext Customer");
    pmp->DDM_rc = DDMUnLoadFileNext(FileLoadHandle,
                                    (ULONG)0,
                                    &DDM_Flags,
                                    (PDDMRECORD)pmp->pRecord,
                                    (ULONG)pmp->RecordSize,
                                    &RecCount);
    if (pmp->DDM_rc != SC_NO_ERROR)
    {
        sprintf(szTraceData,

```

```

        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
    TraceActivity(pmp, szTraceData);
}
else Move_Cust_Record(pmp);
sprintf(szLoadBuffer,
        "%6s%-20s%-20s%-25s%-25s%-2s%-5s%-8s\r\n",
        pmp->szCustNumber,
        pmp->szCustLName,
        pmp->szCustFName,
        pmp->szCustAddress,
        pmp->szCustCity,
        pmp->szCustState,
        pmp->szCustZip,
        pmp->szCustAccount);
bytes_to_write = strlen(szLoadBuffer);
dos_rc = DosWrite(Target_File,
        szLoadBuffer,
        bytes_to_write,
        &byte_count);
if (dos_rc != NO_ERROR)
{
    strcpy(pmp->szDubErr, "OS/2 Write Error on Customer Target");
    Find_OS2_Error(pmp, dos_rc);
    pmp->GetCount = -1;
    break;
}
else
{
    ++pmp->GetCount;
    WinPostMsg(pmp->hwndCaller,
        WM_USER_SET_TEXT,
        MPFROMLONG(u1RecCount),
        MPFROMLONG(pmp->GetCount));
}
}
dos_rc = DosClose(Target_File);
break;

case DFMSAMP_ACCT_UNLOAD:
/* ***** */
/* Unload the account base file into a flat file */
/* - Calculate number of source records */
/* (see routine QueryDDMCount) */
/* - Open the target file */
/* - Issue DDMUnLoadFileFirst in key sequence */
/* - Write first target record */
/* - Loop using DDMUnLoadFileNext if MORE_DATA flag */
/* is returned on previous call then write next */
/* target record */
/* - Close target file */
/* ***** */
memset(szFullAcctName, '\0', sizeof(szFullAcctName));
memcpy(szFullAcctName, pmp->szKeyedAcctDriveID, 1);
strcat(szFullAcctName, ".");
strcat(szFullAcctName, pmp->szKeyedAcctName);
u1RecCount = QueryDDMCount(pmp, szFullAcctName);
pmp->GetCount = 0;
dos_rc = DosOpen(pmp->szUnloadAccountFileName,
        &Target_File,
        &file_open_action,
        (ULONG) 0,
        FILE_NORMAL,
        OPEN_ACTION_REPLACE_IF_EXISTS |
        OPEN_ACTION_CREATE_IF_NEW,
        OPEN_ACCESS_WRITEONLY |
        OPEN_SHARE_DENYWRITE |
        OPEN_FLAGS_SEQUENTIAL,
        0L);
if (dos_rc != NO_ERROR)
{
    strcpy(pmp->szDubErr, "OS/2 Open Error on Account Target");
    Find_OS2_Error(pmp, dos_rc);
    pmp->GetCount = -1;
    break;
}
strcpy(pmp->szDubErr, "DDMUnLoadFileFirst Account");
pmp->DDM_rc = DDMUnLoadFileFirst(szFullAcctName,
        &FileLoadHandle,
        (ULONG) 0,
        &DDM_Flags,
        (PDDMRECORD) pmp->pRecordA,

```

<<<<<

```

        (ULONG)pmp->RecordSizeA,
        (CODEPOINT)KEYORD,
        &RecCount);
if (pmp->DDM_rc != SC_NO_ERROR)
{
    sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
    TraceActivity(pmp, szTraceData);
}
else
{
    Move_Account_Record(pmp);
    sprintf(szLoadBuffer,
        "%0-8s%-40s%-25s%-25s%-2s%-5s%09u\r\n",
        pmp->szAcctNumber,
        pmp->szAcctName,
        pmp->szAcctAddress,
        pmp->szAcctCity,
        pmp->szAcctState,
        pmp->szAcctZip,
        pmp->AccountBuf.ulAcctBalance);
    bytes_to_write = strlen(szLoadBuffer);
    dos_rc = DosWrite(Target_File,
        szLoadBuffer,
        bytes_to_write,
        &byte_count);
    if (dos_rc != NO_ERROR)
    {
        strcpy(pmp->szDubErr, "OS/2 Write Error on Account Target");
        Find_OS2_Error(pmp, dos_rc);
        pmp->GetCount = -1;
        break;
    }
    pmp->GetCount = 1;
    WinPostMsg(pmp->hwndCaller,
        WM_USER_SET_TEXT,
        MPFROMLONG(ulRecCount),
        MPFROMLONG(pmp->GetCount));
}
while ((pmp->DDM_rc == SC_NO_ERROR) && (DDM_Flags == DDM_MOREDATA))
{
    strcpy(pmp->szDubErr, "DDMUnLoadFileNext Account");
    pmp->DDM_rc = DDMUnLoadFileNext(FileLoadHandle,
        (ULONG)0,
        &DDM_Flags,
        (PDDMRECORD)pmp->pRecordA,
        (ULONG)pmp->RecordSizeA,
        &RecCount);
    if (pmp->DDM_rc != SC_NO_ERROR)
    {
        sprintf(szTraceData,
            "DFMTRHD - %s RC = %d",
            pmp->szDubErr,
            pmp->DDM_rc);
        TraceActivity(pmp, szTraceData);
    }
    else Move_Account_Record(pmp);
    sprintf(szLoadBuffer,
        "%0-8s%-40s%-25s%-25s%-2s%-5s%09u\r\n",
        pmp->szAcctNumber,
        pmp->szAcctName,
        pmp->szAcctAddress,
        pmp->szAcctCity,
        pmp->szAcctState,
        pmp->szAcctZip,
        pmp->AccountBuf.ulAcctBalance);
    bytes_to_write = strlen(szLoadBuffer);
    dos_rc = DosWrite(Target_File,
        szLoadBuffer,
        bytes_to_write,
        &byte_count);
    if (dos_rc != NO_ERROR)
    {
        strcpy(pmp->szDubErr, "OS/2 Write Error on Account Target");
        Find_OS2_Error(pmp, dos_rc);
        pmp->GetCount = -1;
        break;
    }
    else
    {
        ++pmp->GetCount;
    }
}

```

```

        WinPostMsg(pmp->hwndCaller,
            WM_USER_SET_TEXT,
            MPFROMLONG(u1RecCount),
            MPFROMLONG(pmp->GetCount));
    }
}
dos_rc = DosClose(Target_File);
break;

case DFMSAMP_CREATE_CUST:
/* ***** */
/* Create a customer base file */
/* Process: */
/* - Build file name from values on Keyed File */
/* Settings Window */
/* - Build Key Field Definition Buffer */
/* - Adjust record length if ADL will be used */
/* - Issue DDMCreateRecFile with long list of values */
/* ***** */
memset(szFullBaseName, '\0', sizeof(szFullBaseName));
memcpy(szFullBaseName, pmp->szKeyedCustDriveID, 1);
strcat(szFullBaseName, ".");
strcat(szFullBaseName, pmp->szKeyedBaseName);
sprintf(szTraceData, "DFMTRHD - Creating: %s", szFullBaseName);
TraceActivity(pmp, szTraceData);
KeyDefBuf.cbKeyDefBuf = (ULONG)sizeof(MYKEYDEFBUF);
KeyDefBuf.cpKeyDefBuf = KEYDEF;
KeyDefBuf.KeyFldDef[0].cbKeyFldDef = (ULONG)sizeof(KEYFLDDEF);
KeyDefBuf.KeyFldDef[0].cpKeyFldDef = KEYFLDDF;
KeyDefBuf.KeyFldDef[0].cpSequence = SEQASC;
KeyDefBuf.KeyFldDef[0].cpKeyClass = BYTSTRDR;
KeyDefBuf.KeyFldDef[0].cbKeyField = (USHORT)6;
KeyDefBuf.KeyFldDef[0].oKeyField = (ULONG)0;
if(pmp->fCust_ADL_Assoc)
    u1RecordLen = CUSTRECORDLEN + 2;
else
    u1RecordLen = CUSTRECORDLEN;
strcpy(pmp->szDubErr, "DDMCreateRecFile Customer");
pmp->DDM_rc = DDMCreateRecFile
/* File Name */
/* Create Flags */
/* Create Flags */
/* RecLen */
/* RecLenCls */
/* KeyDefBuf */
/* InitFilSiz */
/* MaxFileSiz */
/* DupFilOpt */
/* DftRecOp */
/* RecCnt */
/* EABuf */
/* FileClass */
/* DftRecBuf */
    (szFullBaseName,
        DDM_GETCP | DDM_INSCP |
        DDM_MODCP | DDM_DELCF,
        u1RecordLen,
        RECFIX,
        (PKEYDEFBUF)&KeyDefBuf,
        (ULONG)10,
        (LONG)-1,
        DUPFILDO,
        NIL,
        (ULONG)0,
        NULL,
        KEYFIL,
        NULL);
    sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
    TraceActivity(pmp, szTraceData);
    break;

case DFMSAMP_CREATE_CUST_ALT:
/* ***** */
/* Create a customer alternate index file */
/* Process: */
/* - Build alternate index file name from values on */
/* Keyed File Settings Window */
/* - Build base file name from values on Keyed File */
/* Settings Window */
/* - Build Key Field Definition Buffer for name field */
/* - Issue DDMCreateAltIndex pointing to base file */
/* ***** */
memset(szFullIndexName, '\0', sizeof(szFullIndexName));
memcpy(szFullIndexName, pmp->szKeyedCustDriveID, 1);
strcat(szFullIndexName, ".");
strcat(szFullIndexName, pmp->szKeyedIndexName);
memset(szFullBaseName, '\0', sizeof(szFullBaseName));
memcpy(szFullBaseName, pmp->szKeyedCustDriveID, 1);
strcat(szFullBaseName, ".");
strcat(szFullBaseName, pmp->szKeyedBaseName);
sprintf(szTraceData, "DFMTRHD - Creating: %s", szFullIndexName);
TraceActivity(pmp, szTraceData);
KeyDefBuf.cbKeyDefBuf = (ULONG)sizeof(MYKEYDEFBUF);
KeyDefBuf.cpKeyDefBuf = KEYDEF;

```

```

KeyDefBuf.KeyFldDef[0].cbKeyFldDef = (ULONG)sizeof(KEYFLDDEF);
KeyDefBuf.KeyFldDef[0].cpKeyFldDef = KEYFLDDF;
KeyDefBuf.KeyFldDef[0].cpSequence = SEQASC;
KeyDefBuf.KeyFldDef[0].cpKeyClass = BYTSTRDR;
KeyDefBuf.KeyFldDef[0].cbKeyField = (USHORT)20;
KeyDefBuf.KeyFldDef[0].oKeyField = (ULONG)6;
strcpy(pmp->szDubErr,"DDMCreateAltIndex");
pmp->DDM_rc = DDMCreateAltIndex                                     <<<<<
    (szFullIndexName, /* FileName */
     szFullBaseName, /* BaseLName */
     DDM_ALDUPKEY, /* CreateFlags */
     (PKEYDEFBUF)&KeyDefBuf, /* KeyDefBuf */
     DUPFILDO, /* DupFilOpt */
     NULL); /* DftRecBuf */
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
break;

case DFMSAMP_CREATE_ACCT:
/* ***** */
/* Create an account base file */
/* Process: */
/* - Build file name from values on Keyed File */
/* Settings Window */
/* - Build Key Field Definition Buffer */
/* - Adjust record length if ADL will be used */
/* - Issue DDMCreateRecFile with long list of values */
/* ***** */
memset(szFullAcctName,'\\0', sizeof(szFullAcctName));
memcpy(szFullAcctName,pmp->szKeyedAcctDriveID,1);
strcat(szFullAcctName,".");
strcat(szFullAcctName,pmp->szKeyedAcctName);
sprintf(szTraceData,"DFMTRHD - Creating: %s",szFullAcctName);
TraceActivity(pmp, szTraceData);
KeyDefBuf.cbKeyDefBuf = (ULONG)sizeof(MYKEYDEFBUF);
KeyDefBuf.cpKeyDefBuf = KEYDEF;
KeyDefBuf.KeyFldDef[0].cbKeyFldDef = (ULONG)sizeof(KEYFLDDEF);
KeyDefBuf.KeyFldDef[0].cpKeyFldDef = KEYFLDDF;
KeyDefBuf.KeyFldDef[0].cpSequence = SEQASC;
KeyDefBuf.KeyFldDef[0].cpKeyClass = BYTSTRDR;
KeyDefBuf.KeyFldDef[0].cbKeyField = (USHORT)8;
KeyDefBuf.KeyFldDef[0].oKeyField = (ULONG)0;
if(pmp->fAcct_ADL_Assoc)
    ulRecordLen = ACCTRECORDLEN + 3;
else
    ulRecordLen = ACCTRECORDLEN;
strcpy(pmp->szDubErr,"DDMCreateRecFile Account");
pmp->DDM_rc = DDMCreateRecFile                                     <<<<<
    (szFullAcctName, /* FileName */
     DDM_GETCP | DDM_INSCP | /* CreateFlags */
     DDM_MODCP | DDM_DELCP,
     ulRecordLen, /* RecLen */
     RECFIX, /* RecLenCls */
     (PKEYDEFBUF)&KeyDefBuf, /* KeyDefBuf */
     (ULONG)10, /* InitFilSiz */
     (LONG)-1, /* MaxFileSiz */
     DUPFILDO, /* DupFilOpt */
     NIL, /* DftRecOp */
     (ULONG)0, /* RecCnt */
     NULL, /* EABuf */
     KEYFIL, /* FileClass */
     NULL); /* DftRecBuf */
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
break;

case DFMSAMP_DELETE_CUST_FILE:
/* ***** */
/* Delete a customer base file */
/* Process: */
/* - Build base file name from values on Keyed File */
/* Settings Window */
/* - Issue DDMDelete using base file name */
/* ***** */
memset(szFullBaseName,'\\0', sizeof(szFullBaseName));
memcpy(szFullBaseName,pmp->szKeyedCustDriveID,1);
strcat(szFullBaseName,".");

```



```

strcat(szFullBaseName,pmp->szKeyedBaseName);
sprintf(szTraceData,"DFMTRHD - Deleting: %s",szFullBaseName);
TraceActivity(pmp, szTraceData);
strcpy(pmp->szDubErr,"DDMDelete Customer");
pmp->DDM_rc = DDMDelete
    (szFullBaseName,
     (ULONG)0);
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
break;

case DFMSAMP_DELETE_CUST_ALT_FILE:
/* ***** */
/* Delete a customer alternate index file */
/* Process: */
/* - Build alternate index file name from values on */
/* Keyed File */
/* Settings Window */
/* - Issue DDMDelete using alternate index file name */
/* ***** */
memset(szFullIndexName,'\0',sizeof(szFullIndexName));
memcpy(szFullIndexName,pmp->szKeyedCustDriveID,1);
strcat(szFullIndexName,".");
strcat(szFullIndexName,pmp->szKeyedIndexName);
sprintf(szTraceData,"DFMTRHD - Deleting: %s",szFullIndexName);
TraceActivity(pmp, szTraceData);
strcpy(pmp->szDubErr,"DDMDelete Index");
pmp->DDM_rc = DDMDelete
    (szFullIndexName,
     (ULONG)0);
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
break;

case DFMSAMP_DELETE_ACCT_FILE:
/* ***** */
/* Delete an account base file */
/* Process: */
/* - Build Account file name from values on Keyed */
/* File Settings Window */
/* - Issue DDMDelete using account file name */
/* ***** */
memset(szFullAcctName,'\0',sizeof(szFullAcctName));
memcpy(szFullAcctName,pmp->szKeyedAcctDriveID,1);
strcat(szFullAcctName,".");
strcat(szFullAcctName,pmp->szKeyedAcctName);
sprintf(szTraceData,"DFMTRHD - Deleting: %s",szFullAcctName);
TraceActivity(pmp, szTraceData);
strcpy(pmp->szDubErr,"DDMDelete Account");
pmp->DDM_rc = DDMDelete
    (szFullAcctName,
     (ULONG)0);
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
break;

case DFMSAMP_QUERY_CUST:
/* ***** */
/* Query a customer base file */
/* Process: */
/* - Build base file name from values on Keyed File */
/* Settings Window */
/* - eaop2 structure has already been built by */
/* query window proc */
/* - Issue DDMQueryPathInfo */
/* ***** */
memset(szFullBaseName,'\0',sizeof(szFullBaseName));
memcpy(szFullBaseName,pmp->szKeyedCustDriveID,1);
strcat(szFullBaseName,".");
strcat(szFullBaseName,pmp->szKeyedBaseName);
sprintf(szTraceData,
        "DFMTRHD - Querying %s for %s",
        pmp->szKeyedCustDriveID,
        pmp->szKeyedBaseName);

```

```

        TraceActivity(pmp, szTraceData);
/* get info */
        FileInfoLevel = 1;
        strcpy(pmp->szDubErr, "DDMQueryPathInfo Customer");
        pmp->DDM_rc = DDMQueryPathInfo
            (szFullBaseName,
             FileInfoLevel,
             (PBYTE)&pmp->eaop2,
             sizeof(pmp->eaop2));
            /* On successful return, the Level 1 */
            /* file information is in the */
            /* pfea2list */
            /*
        sprintf(szTraceData,
            "DFMTRHD - %s RC = %d",
            pmp->szDubErr,
            pmp->DDM_rc);
        TraceActivity(pmp, szTraceData);
        break;

case DFMSAMP_QUERY_INDEX:
/* ***** */
/* Query a customer alternate index file */
/* Process: */
/* - Build alternate index file name from values on */
/* Keyed File Settings Window */
/* - eaop2 structure has already been built by */
/* query window proc */
/* - Issue DDMQueryPathInfo */
/* ***** */
memset(szFullIndexName, '\0', sizeof(szFullIndexName));
memcpy(szFullIndexName, pmp->szKeyedCustDriveID, 1);
strcat(szFullIndexName, ".");
strcat(szFullIndexName, pmp->szKeyedIndexName);
sprintf(szTraceData,
    "DFMTRHD - Querying %s for %s",
    pmp->szKeyedCustDriveID,
    pmp->szKeyedIndexName);
    TraceActivity(pmp, szTraceData);
/* get info */
        FileInfoLevel = 1;
        strcpy(pmp->szDubErr, "DDMQueryPathInfo Index");
        pmp->DDM_rc = DDMQueryPathInfo
            (szFullIndexName,
             FileInfoLevel,
             (PBYTE)&pmp->eaop2,
             sizeof(pmp->eaop2));
            /* On successful return, the Level 1 */
            /* file information is in the */
            /* pfea2list */
            /*
        sprintf(szTraceData,
            "DFMTRHD - %s RC = %d",
            pmp->szDubErr,
            pmp->DDM_rc);
        TraceActivity(pmp, szTraceData);
        break;

case DFMSAMP_QUERY_ACCT:
/* ***** */
/* Query an account base file */
/* Process: */
/* - Build account file name from values on Keyed */
/* File Settings Window */
/* - eaop2 structure has already been built by */
/* query window proc */
/* - Issue DDMQueryPathInfo */
/* ***** */
memset(szFullAcctName, '\0', sizeof(szFullAcctName));
memcpy(szFullAcctName, pmp->szKeyedAcctDriveID, 1);
strcat(szFullAcctName, ".");
strcat(szFullAcctName, pmp->szKeyedAcctName);
sprintf(szTraceData,
    "DFMTRHD - Querying %s for %s",
    pmp->szKeyedAcctDriveID,
    pmp->szKeyedAcctName);
    TraceActivity(pmp, szTraceData);
/* get info */
        FileInfoLevel = 1;
        strcpy(pmp->szDubErr, "DDMQueryPathInfo Account");
        pmp->DDM_rc = DDMQueryPathInfo
            (szFullAcctName,
             FileInfoLevel,
             (PBYTE)&pmp->eaop2,
             sizeof(pmp->eaop2));

```

```

/* On successful return, the Level 1 */
/* file information is in the      */
/* pfea2list                       */
sprintf(szTraceData,
        "DFMTRHD - %s RC = %d",
        pmp->szDubErr,
        pmp->DDM_rc);
TraceActivity(pmp, szTraceData);
break;

case DFMSAMP_SHUTDOWN:
/* ***** */
/* Thread shutdown request */
/* Process: */
/* - Set Shutdown flag to end loop */
/* ***** */
TraceActivity(pmp, "DFMTRHD - Thread Shutting Down");
ShutDown = TRUE;
break;

default:
break;
} /* endswitch */
/* ***** */
/* We have finished with whatever was the last DDM */
/* Request */
/* Process: */
/* - If an error has occurred get the reply message */
/* and trace its contents including the full */
/* SRVDGN */
/* - Calculate the time it took for the complete DDM */
/* process and trace it */
/* - Reset semaphore and post caller that we are done */
/* ***** */
if (pmp->DDM_rc != SC_NO_ERROR)
{
    pmp->CodePoint = ReplyMsg(szDubErr,
                             pmp->szDubMsg,
                             pmp->szSRVDGN);
    TraceActivity(pmp, pmp->szDubErr);
    TraceActivity(pmp, pmp->szDubMsg);
    TraceActivity(pmp, pmp->szSRVDGN);
}
DosGetDateTime(&FunctionTime);
ulEndTime = ((FunctionTime.hundredths +
              (FunctionTime.seconds * 100) +
              (FunctionTime.minutes * 6000) +
              (FunctionTime.hours * 360000)) -
             ulStartTime) * 10;

sprintf(szTraceData,
        "DFMTRHD - Exit RC = %d (%d ms)",
        pmp->DDM_rc,
        ulEndTime);
TraceActivity(pmp, szTraceData);
DosResetEventSem(pmp->hevWorker, &posted);
WinPostMsg(pmp->hwndCaller, WM_USER_DFM_DONE, NULL, NULL);
}
return;
}

VOID Move_Cust_Buffer(PMAIN_PARM pmp, PSZ szLoadBuffer)
{
    DATETIME CurrentDate;
    pmp->pRecord->cbRecord = pmp->RecordSize;
    pmp->pRecord->cpRecord = RECORD;
    memset(&pmp->RecordBuf, ' ', sizeof(pmp->RecordBuf));
    memcpy(pmp->RecordBuf.szBaseNumber,
           szLoadBuffer,
           sizeof(pmp->RecordBuf.szBaseNumber));
    memcpy(pmp->RecordBuf.szBaseLName,
           &szLoadBuffer[6],
           sizeof(pmp->RecordBuf.szBaseLName));
    memcpy(pmp->RecordBuf.szBaseFName,
           &szLoadBuffer[26],
           sizeof(pmp->RecordBuf.szBaseFName));
    memcpy(pmp->RecordBuf.szBaseAddress,
           &szLoadBuffer[46],
           sizeof(pmp->RecordBuf.szBaseAddress));
    memcpy(pmp->RecordBuf.szBaseCity,
           &szLoadBuffer[71],
           sizeof(pmp->RecordBuf.szBaseCity));
    memcpy(pmp->RecordBuf.szBaseState,

```

```

        &szLoadBuffer[96],
        sizeof(pmp->RecordBuf.szBaseZip));
memcpy(pmp->RecordBuf.szBaseZip,
        &szLoadBuffer[98],
        sizeof(pmp->RecordBuf.szBaseZip));
memcpy(pmp->RecordBuf.szBaseAccount,
        &szLoadBuffer[103],
        sizeof(pmp->RecordBuf.szBaseAccount));
DosGetDateTime(&CurrentDate);
pmp->RecordBuf.usBaseYear = CurrentDate.year;
pmp->RecordBuf.usBaseMonth = CurrentDate.month;
pmp->RecordBuf.usBaseDay = CurrentDate.day;
pmp->RecordBuf.usBaseHour = CurrentDate.hours;
pmp->RecordBuf.usBaseMinute = CurrentDate.minutes;
pmp->RecordBuf.usBaseSecond = CurrentDate.seconds;
memcpy(pmp->pRecord->pRecord,
        &pmp->RecordBuf,
        sizeof(pmp->RecordBuf));
}

VOID Move_Account_Buffer(PMAIN_PARM pmp, PSZ szLoadBuffer)
{
    DATETIME    CurrentDate;
    CHAR        szTempBalance[10];
    PCH         szEndString = NULL;
    pmp->pRecordA->cbRecord = pmp->RecordSizeA;
    pmp->pRecordA->cpRecord = RECORD;
    memset(&pmp->AccountBuf, ' ', sizeof(pmp->AccountBuf));
    memcpy(pmp->AccountBuf.szAcctNumber,
        szLoadBuffer,
        sizeof(pmp->AccountBuf.szAcctNumber)); // 8
    memcpy(pmp->AccountBuf.szAcctName,
        &szLoadBuffer[8],
        sizeof(pmp->AccountBuf.szAcctName)); // 40
    memcpy(pmp->AccountBuf.szAcctAddress,
        &szLoadBuffer[48],
        sizeof(pmp->AccountBuf.szAcctAddress)); // 25
    memcpy(pmp->AccountBuf.szAcctCity,
        &szLoadBuffer[73],
        sizeof(pmp->AccountBuf.szAcctCity)); // 25
    memcpy(pmp->AccountBuf.szAcctState,
        &szLoadBuffer[98],
        sizeof(pmp->AccountBuf.szAcctState)); // 2
    memcpy(pmp->AccountBuf.szAcctZip,
        &szLoadBuffer[100],
        sizeof(pmp->AccountBuf.szAcctZip)); // 5
    memset(szTempBalance, '0', sizeof(szTempBalance));
    memcpy(szTempBalance,
        &szLoadBuffer[105],
        sizeof(szTempBalance)-1);
    pmp->AccountBuf.ulAcctBalance = (ULONG)strtol(szTempBalance, &szEndString, 10);
    DosGetDateTime(&CurrentDate);
    pmp->AccountBuf.usAcctYear = CurrentDate.year;
    pmp->AccountBuf.usAcctMonth = CurrentDate.month;
    pmp->AccountBuf.usAcctDay = CurrentDate.day;
    pmp->AccountBuf.usAcctHour = CurrentDate.hours;
    pmp->AccountBuf.usAcctMinute = CurrentDate.minutes;
    pmp->AccountBuf.usAcctSecond = CurrentDate.seconds;
    memcpy(pmp->pRecordA->pRecord,
        &pmp->AccountBuf,
        sizeof(pmp->AccountBuf));
}

VOID Move_Cust_Record(PMAIN_PARM pmp)
{
    memcpy(pmp->RecordBuf.szBaseNumber,
        pmp->pRecord->pRecord,
        sizeof(pmp->RecordBuf));
    memcpy(pmp->szCustNumber,
        pmp->RecordBuf.szBaseNumber,
        sizeof(pmp->RecordBuf.szBaseNumber));
    memcpy(pmp->szCustLName,
        pmp->RecordBuf.szBaseLName,
        sizeof(pmp->RecordBuf.szBaseLName));
    memcpy(pmp->szCustFName,
        pmp->RecordBuf.szBaseFName,
        sizeof(pmp->RecordBuf.szBaseFName));
    memcpy(pmp->szCustAddress,
        pmp->RecordBuf.szBaseAddress,
        sizeof(pmp->RecordBuf.szBaseAddress));
    memcpy(pmp->szCustCity,
        pmp->RecordBuf.szBaseCity,
        sizeof(pmp->RecordBuf.szBaseCity));
}

```

```

memcpy(pmp->szCustState,
       pmp->RecordBuf.szBaseState,
       sizeof(pmp->RecordBuf.szBaseState));
memcpy(pmp->szCustZip,
       pmp->RecordBuf.szBaseZip,
       sizeof(pmp->RecordBuf.szBaseZip));
memcpy(pmp->szCustAccount,
       pmp->RecordBuf.szBaseAccount,
       sizeof(pmp->RecordBuf.szBaseAccount));
StripString(pmp->szCustNumber);
StripString(pmp->szCustLName);
StripString(pmp->szCustFName);
StripString(pmp->szCustAddress);
StripString(pmp->szCustCity);
StripString(pmp->szCustState);
StripString(pmp->szCustZip);
StripString(pmp->szCustAccount);
sprintf(pmp->szCustDate,
        "%.2d/%.2d/%.4d",
        pmp->RecordBuf.usBaseMonth,
        pmp->RecordBuf.usBaseDay,
        pmp->RecordBuf.usBaseYear);
sprintf(pmp->szCustTime,
        "%.2d:%.2d:%.2d",
        pmp->RecordBuf.usBaseHour,
        pmp->RecordBuf.usBaseMinute,
        pmp->RecordBuf.usBaseSecond);
}

VOID Move_Account_Record(PMAIN_PARM pmp)
{
    ULONG    ulDollars;
    ULONG    ulCents;
    memcpy(pmp->AccountBuf.szAcctNumber,
           pmp->pRecordA->pRecord,
           sizeof(pmp->AccountBuf));
    memcpy(pmp->szAcctNumber,
           pmp->AccountBuf.szAcctNumber,
           sizeof(pmp->AccountBuf.szAcctNumber));
    memcpy(pmp->szAcctName,
           pmp->AccountBuf.szAcctName,
           sizeof(pmp->AccountBuf.szAcctName));
    StripString(pmp->szAcctName);
    memcpy(pmp->szAcctAddress,
           pmp->AccountBuf.szAcctAddress,
           sizeof(pmp->AccountBuf.szAcctAddress));
    StripString(pmp->szAcctAddress);
    memcpy(pmp->szAcctCity,
           pmp->AccountBuf.szAcctCity,
           sizeof(pmp->AccountBuf.szAcctCity));
    StripString(pmp->szAcctCity);
    memcpy(pmp->szAcctState,
           pmp->AccountBuf.szAcctState,
           sizeof(pmp->AccountBuf.szAcctState));
    StripString(pmp->szAcctState);
    memcpy(pmp->szAcctZip,
           pmp->AccountBuf.szAcctZip,
           sizeof(pmp->AccountBuf.szAcctZip));
    StripString(pmp->szAcctZip);
    ulCents = pmp->AccountBuf.ulAcctBalance%(ULONG)100;
    ulDollars = pmp->AccountBuf.ulAcctBalance/(ULONG)100;
    sprintf(pmp->szAcctBalance, "$u.%02u", ulDollars, ulCents);
    sprintf(pmp->szAcctDate,
           "%.2d/%.2d/%.4d",
           pmp->AccountBuf.usAcctMonth,
           pmp->AccountBuf.usAcctDay,
           pmp->AccountBuf.usAcctYear);
    sprintf(pmp->szAcctTime,
           "%.2d:%.2d:%.2d",
           pmp->AccountBuf.usAcctHour,
           pmp->AccountBuf.usAcctMinute,
           pmp->AccountBuf.usAcctSecond);
}

VOID Find_OS2_Error(PMAIN_PARM pmp, ULONG RetCode)
{
    /* fields for DosGetMsg usage */
    PCHAR    *DosMsg_table;          /* table of variables to insert */
    USHORT   DosMsg_count = 0;       /* number of variables to insert = 0 */
    CHAR     DosMsg_msgarea[120];    /* buffer address to return message */
    USHORT   DosMsg_msgarea_len = 119; /* length of the input buffer */
    USHORT   DosMsg_msgnum;          /* message number to expand */
    static char DosMsg_msgfile[] = "OS0001.MSG"; /* system message file */

```

```

ULONG      DosMsg_msg_len;          /* length of the returned message */
ULONG      inv_char_loc = 0;        /* index to invalid character */
DosMsg_table = (char * *)NULL;     /* set up message table */
DosMsg_count = 0;                  /* no variables to insert */
DosMsg_msgarea_len = 119;          /* length to accept */
DosMsg_msgnum = (USHORT) RetCode;   /* message number to find */
DosGetMessage(DosMsg_table,        /* table address */
              DosMsg_count,         /* variable count */
              DosMsg_msgarea,       /* where to put it */
              DosMsg_msgarea_len,   /* what we accept */
              DosMsg_msgnum,        /* which message */
              DosMsg_msgfile,       /* in which file */
              &DosMsg_msg_len);    /* len of return msg */
memset(&DosMsg_msgarea[DosMsg_msg_len-1], '\0', 1);
do
{
    inv_char_loc = strcspn(DosMsg_msgarea, "\n");
    if (inv_char_loc != 0 & inv_char_loc < (DosMsg_msg_len - 1))
        memset(&DosMsg_msgarea[inv_char_loc], '\0', 1);
} while (inv_char_loc != 0 & inv_char_loc < (DosMsg_msg_len - 1));
do
{
    inv_char_loc = strcspn(DosMsg_msgarea, "\r");
    if (inv_char_loc != 0 & inv_char_loc < (DosMsg_msg_len - 1))
        memset(&DosMsg_msgarea[inv_char_loc], '\0', 1);
} while (inv_char_loc != 0 & inv_char_loc < (DosMsg_msg_len - 1));
sprintf(pmp->szDubMsg,
        "OS/2 Message: %s",
        DosMsg_msgarea);
TraceActivity(pmp, pmp->szDubErr);
TraceActivity(pmp, pmp->szDubMsg);
}

ULONG QueryDDMCount(PMAIN_PARM pmp, PSZ szQueryFileName)
{
    CHAR      szTraceData[255] = "";
    ULONG     FileInfoLevel = 1;
    PGEA2LIST pgea2list;
    PFEA2LIST pfea2list;
    PGEA2      pgea2;
    PBYTE      ptrgea2;
    PFEA2      pfea2;
    PBYTE      pData;
    PDDMOBJECT pObject;
    ULONG      ulListLength = 0;
    ULONG      ulEntryLength = 0;
    ULONG      ulRecCount = 1;
    ULONG      cplen;
    CHAR      szDubErr[255] = "";

    /* create the list with 1 GEA items */
    pgea2list = (PGEA2LIST) calloc ( 1, 1024);          // bigger
    pgea2list->cbList = 1024;                            // ... area
    ptrgea2 = (PBYTE)&pgea2list->list;
    pgea2 = (PGEA2) calloc ( 1, 20 );
    ulListLength = 4;

    /* create one list item at a time for GEA list */
    /* define codepoints for basic file information */
    ulEntryLength = AddLastEntry(ptrgea2, pgea2, ".DDM_EOFNBR"); // 1110
    ptrgea2 += ulEntryLength;
    ulListLength += ulEntryLength;
    /* create the list where DosQueryFileInfo places result */
    pfea2list = (PFEA2LIST) calloc ( 1, 4096 );
    pfea2list->cbList = 4096;

    /* put both lists in a eaop2 structure */
    pmp->eaop2.fpGEA2List = pgea2list;
    pmp->eaop2.fpFEA2List = pfea2list;
    free(pgea2);
    sprintf(szTraceData,
            "DFMTRHD - Querying %s for record count",
            szQueryFileName);
    TraceActivity(pmp, szTraceData);
    /* get info */
    FileInfoLevel = 1;
    strcpy(pmp->szDubErr, "DDMQueryPathInfo Customer");
    pmp->DDM_rc = DDMQueryPathInfo
        (szQueryFileName,
         FileInfoLevel,
         (PBYTE)&pmp->eaop2,
         sizeof(pmp->eaop2));
    /* On successful return, the Level 1 */
    /* file information is in the */
}

```

```

/* pfea2list */
if (pmp->DDM_rc != SC_NO_ERROR)
{
    pmp->CodePoint = ReplyMsg(szDubErr,
                             pmp->szDubMsg,
                             pmp->szSRVDGN);
    TraceActivity(pmp, pmp->szDubErr);
    TraceActivity(pmp, pmp->szDubMsg);
    TraceActivity(pmp, pmp->szSRVDGN);
}
else
{
    pFEA = pmp->eaop2.fpFEA2List->list;
    while (pFEA->oNextEntryOffset != 0)
    {
        pData = (PBYTE)&pFEA->szName + pFEA->cbName + 1;
        pObj = (PDDMOBJECT)pData;
        cpLen = pObj->cbObject - 6; // get true length of cp Value
        if (cpLen > 0)
        {
            if (pObj->cpObject == EOFNBR)
            {
                ulRecCount = *(PULONG)((PDDMOBJECT)pObj)->pData;
            }
        }
        pData = (PBYTE)&pFEA->oNextEntryOffset + pFEA->oNextEntryOffset;
        pFEA = (PFEA2)pData;
    } /* endwhile */
    pData = (PBYTE)&pFEA->szName + pFEA->cbName + 1;
    pObj = (PDDMOBJECT)pData;
    cpLen = pObj->cbObject - 6; // get true length of cp Value
    if (cpLen > 0)
    {
        if (pObj->cpObject == EOFNBR)
        {
            ulRecCount = *(PULONG)((PDDMOBJECT)pObj)->pData;
        }
    }
}
free(pgea2list);
free(pgea2);
free(pfea2list);
return (ULONG)max((ulRecCount - 1),1);
}

```

---

## C.2 ADL for Host View of Customer Master Record

```
/******  
/* Copyright (c) IBM Corporation 1994 - All Rights Reserved */  
/* */  
/* DFMSAMP Sample Data Access Services program */  
/* */  
/* File Name: DFMSMPCB.ADL */  
/* Application View: DFMSMPCV.ADL */  
/******  
/******  
/* Start of a host view of customer master record */  
/******  
DECLARE  
BEGIN;  
CustMastB: SEQUENCE  
BEGIN;  
CustNum: CHAR LENGTH(6) CCSID(00037);  
CustLName: CHAR LENGTH(20) CCSID(00037);  
CustFName: CHAR LENGTH(20) CCSID(00037);  
CustAddr: CHAR LENGTH(25) CCSID(00037);  
CustCity: CHAR LENGTH(25) CCSID(00037);  
CustState: CHAR LENGTH(2) CCSID(00037);  
CustZip: CHAR LENGTH(5) CCSID(00037);  
CustAcct: CHAR LENGTH(8) CCSID(00037);  
CustUpDYr: BINARY PRECISION(9) BYTRVS(FALSE) SCALE(0) RADIX(10);  
CustUpDMn: BINARY PRECISION(2) BYTRVS(FALSE) SCALE(0) RADIX(10);  
CustUpDDy: BINARY PRECISION(2) BYTRVS(FALSE) SCALE(0) RADIX(10);  
CustUpTHr: BINARY PRECISION(2) BYTRVS(FALSE) SCALE(0) RADIX(10);  
CustUpTMn: BINARY PRECISION(2) BYTRVS(FALSE) SCALE(0) RADIX(10);  
CustUpTSc: BINARY PRECISION(2) BYTRVS(FALSE) SCALE(0) RADIX(10);  
END;  
END;  
/******  
/* End of a host view of customer master record */  
/******
```

---

## C.3 ADL for Application View of Customer Master Record

```
/******  
/* Copyright (c) IBM Corporation 1994 - All Rights Reserved */  
/* */  
/* DFMSAMP Sample Data Access Services program */  
/* */  
/* File Name: DFMSMPCV.ADL */  
/* Host View: DFMSMPCB.ADL */  
/******  
/******  
/* Start of Application view of customer master record */  
/******  
DECLARE  
BEGIN;  
CustMastV: SEQUENCE  
BEGIN;  
CustNum: CHAR LENGTH(6);  
CustLName: CHAR LENGTH(20);  
CustUpDYr: BINARY PRECISION(2) BYTRVS(TRUE) SCALE(0) RADIX(10);  
CustUpDMn: BINARY PRECISION(2) BYTRVS(TRUE) SCALE(0) RADIX(10);  
CustUpDDy: BINARY PRECISION(2) BYTRVS(TRUE) SCALE(0) RADIX(10);  
CustUpTHr: BINARY PRECISION(2) BYTRVS(TRUE) SCALE(0) RADIX(10);  
CustUpTMn: BINARY PRECISION(2) BYTRVS(TRUE) SCALE(0) RADIX(10);  
CustUpTSc: BINARY PRECISION(2) BYTRVS(TRUE) SCALE(0) RADIX(10);  
CustFName: CHAR LENGTH(20);  
CustAddr: CHAR LENGTH(25);  
CustCity: CHAR LENGTH(25);  
CustState: CHAR LENGTH(2);  
CustZip: CHAR LENGTH(5);  
CustAcct: CHAR LENGTH(8);  
END;  
END;  
/******  
/* End of Application view of customer master record */  
/******
```



---

## C.4 ADL for Host View of Account Details Record

```

/*****
/* Copyright (c) IBM Corporation 1994 - All Rights Reserved */
/*
/* DFMSAMP Sample Data Access Services program */
/*
/* File Name: DFMSMPAB.ADL */
/* Application View: DFMSMPAV.ADL */
*****/
/*****
/* Start of Host view of Account details record */
*****/
DECLARE
BEGIN;
  AcctMastB: SEQUENCE
  BEGIN;
    AcctNum: CHAR LENGTH(8) CCSID(00037);
    AcctName: CHAR LENGTH(40) CCSID(00037);
    AcctAddr: CHAR LENGTH(25) CCSID(00037);
    AcctCity: CHAR LENGTH(25) CCSID(00037);
    AcctState: CHAR LENGTH(2) CCSID(00037);
    AcctZip: CHAR LENGTH(5) CCSID(00037);
    AcctBalanc: PACKED PRECISION(9) SCALE(0);
    AcctUpDyr: BINARY PRECISION(9) BYTRVS(FALSE) SCALE(0) RADIX(10);
    AcctUpDMn: BINARY PRECISION(2) BYTRVS(FALSE) SCALE(0) RADIX(10);
    AcctUpDDy: BINARY PRECISION(2) BYTRVS(FALSE) SCALE(0) RADIX(10);
    AcctUpThr: BINARY PRECISION(2) BYTRVS(FALSE) SCALE(0) RADIX(10);
    AcctUpTMn: BINARY PRECISION(2) BYTRVS(FALSE) SCALE(0) RADIX(10);
    AcctUpTSc: BINARY PRECISION(2) BYTRVS(FALSE) SCALE(0) RADIX(10);

  END;
END;
/*****
/* End of Host view of Account details record */
*****/

```

---

## C.5 ADL for Application View of Account Details Record

```

/*****
/* Copyright (c) IBM Corporation 1994 - All Rights Reserved */
/*
/* DFMSAMP Sample Data Access Services program */
/*
/* File Name: DFMSMPAV.ADL */
/* Host View: DFMSMPAB.ADL */
*****/
/*****
/* Start Application view of Account details record */
*****/
DECLARE
BEGIN;
  AcctMastV: SEQUENCE
  BEGIN;
    AcctNum: CHAR LENGTH(8);
    AcctBalanc: BINARY PRECISION(9) BYTRVS(TRUE) SCALE(0) RADIX(10);
    AcctUpDyr: BINARY PRECISION(2) BYTRVS(TRUE) SCALE(0) RADIX(10);
    AcctUpDMn: BINARY PRECISION(2) BYTRVS(TRUE) SCALE(0) RADIX(10);
    AcctUpDDy: BINARY PRECISION(2) BYTRVS(TRUE) SCALE(0) RADIX(10);
    AcctUpThr: BINARY PRECISION(2) BYTRVS(TRUE) SCALE(0) RADIX(10);
    AcctUpTMn: BINARY PRECISION(2) BYTRVS(TRUE) SCALE(0) RADIX(10);
    AcctUpTSc: BINARY PRECISION(2) BYTRVS(TRUE) SCALE(0) RADIX(10);
    AcctName: CHAR LENGTH(40);
    AcctAddr: CHAR LENGTH(25);
    AcctCity: CHAR LENGTH(25);
    AcctState: CHAR LENGTH(2);
    AcctZip: CHAR LENGTH(5);

  END;
END;
/*****
/* End Application view of Account details record */
*****/

```

## C.6 LISTCAT Output for Customer Master File

IDCAMS SYSTEM SERVICES

TIME: 10:42:21

08/19/94

PAGE

1

```
CLUSTER ----- PJRES1.DFMSAMP.CUST
IN-CAT --- SYS1.ICFCAT.VSYS366
HISTORY
  DATASET-OWNER----- (NULL)      CREATION-----1994.231
  RELEASE-----2      EXPIRATION-----0000.000
SMSDATA
  STORAGECLASS -----NORMAL      MANAGEMENTCLASS--PRIMARY
  DATACLASS ----- (NULL)      LBACKUP ---0000.000.0000
  BWO STATUS-----00000000      BWO TIMESTAMP---00000 00:00:00.0
  PROTECTION-PSWD----- (NULL)    RACF----- (NO)
ASSOCIATIONS
  DATA-----PJRES1.DFMSAMP.CUST.DATA
  INDEX-----PJRES1.DFMSAMP.CUST.INDEX

DATA ----- PJRES1.DFMSAMP.CUST.DATA
IN-CAT --- SYS1.ICFCAT.VSYS366
HISTORY
  DATASET-OWNER----- (NULL)      CREATION-----1994.231
  RELEASE-----2      EXPIRATION-----0000.000
  PROTECTION-PSWD----- (NULL)    RACF----- (NO)
ASSOCIATIONS
  CLUSTER--PJRES1.DFMSAMP.CUST
ATTRIBUTES
  KEYLEN-----6      AVGLRECL-----125      BUFSIZE-----45568      CFSIZE-----22528
  RKP-----0      MAXLRECL-----125      EXCEXIT----- (NULL)    CI/CA-----2
  SHROPTNS (1,3)      SPEED      UNIQUE      NOERASE      INDEXED      NOWRITECHK      NOIMBED      NOREPLICAT
  UNORDERED      REUSE      NONSPANNED      BINARY
  CCSID-----0
STATISTICS
  REC-TOTAL-----6      SPLITS-CI-----0      EXCPS-----7
  REC-DELETED-----0      SPLITS-CA-----0      EXTENTS-----1
  REC-INSERTED-----0      FREESPACE-%CI-----0      SYSTEM-TIMESTAMP:
  REC-UPDATED-----2      FREESPACE-%CA-----0      X' A9C01E2C66135781'
  REC-RETRIEVED-----55      FREESPC-BYTES-----22528
ALLOCATION
  SPACE-TYPE-----TRACK      HI-ALLOC-RBA-----45056
  SPACE-PRI-----1      HI-USED-RBA-----45056
  SPACE-SEC-----0
VOLUME
  VOLSER-----SYS308      PHYREC-SIZE-----22528      HI-ALLOC-RBA-----45056      EXTENT-NUMBER-----1
  DEVTYPE-----X'3010200E'      PHYRECS/TRK-----2      HI-USED-RBA-----45056      EXTENT-TYPE-----X'40'
  VOLFLAG-----PRIME      TRACKS/CA-----1
  EXTENTS:
  LOW-CCHH-----X'000F000E'      LOW-RBA-----0      TRACKS-----1
  HIGH-CCHH-----X'000F000E'      HIGH-RBA-----45055

INDEX ----- PJRES1.DFMSAMP.CUST.INDEX
IN-CAT --- SYS1.ICFCAT.VSYS366
HISTORY
  DATASET-OWNER----- (NULL)      CREATION-----1994.231
  RELEASE-----2      EXPIRATION-----0000.000
  PROTECTION-PSWD----- (NULL)    RACF----- (NO)
ASSOCIATIONS
  CLUSTER--PJRES1.DFMSAMP.CUST
ATTRIBUTES
  KEYLEN-----6      AVGLRECL-----0      BUFSIZE-----0      CFSIZE-----512
  RKP-----0      MAXLRECL-----505      EXCEXIT----- (NULL)    CI/CA-----46
  SHROPTNS (1,3)      RECOVERY      UNIQUE      NOERASE      NOWRITECHK      NOIMBED      NOREPLICAT      UNORDERED
  REUSE
STATISTICS
  REC-TOTAL-----1      SPLITS-CI-----0      EXCPS-----5      INDEX:
  REC-DELETED-----0      SPLITS-CA-----0      EXTENTS-----1      LEVELS-----1
  REC-INSERTED-----0      FREESPACE-%CI-----0      SYSTEM-TIMESTAMP:      ENTRIES/SECT-----1
  REC-UPDATED-----0      FREESPACE-%CA-----0      X' A9C01E2C66135781'      SEQ-SET-RBA-----0
  REC-RETRIEVED-----0      FREESPC-BYTES-----23040      HI-LEVEL-RBA-----0
ALLOCATION
  SPACE-TYPE-----TRACK      HI-ALLOC-RBA-----23552
  SPACE-PRI-----1      HI-USED-RBA-----512
  SPACE-SEC-----0
VOLUME
  VOLSER-----SYS308      PHYREC-SIZE-----512      HI-ALLOC-RBA-----23552      EXTENT-NUMBER-----1
  DEVTYPE-----X'3010200E'      PHYRECS/TRK-----46      HI-USED-RBA-----512      EXTENT-TYPE-----X'00'
  VOLFLAG-----PRIME      TRACKS/CA-----1
  EXTENTS:
  LOW-CCHH-----X'001C000A'      LOW-RBA-----0      TRACKS-----1
  HIGH-CCHH-----X'001C000A'      HIGH-RBA-----23551
```

## C.7 LISTCAT Output for Account Details File

IDCAMS SYSTEM SERVICES

TIME: 11:29:20

08/19/94

PAGE

1

```
CLUSTER ----- PJRES1.DFMSAMP.ACCT
IN-CAT --- SYS1.ICFCAT.VSYS107
HISTORY
  DATASET-OWNER----- (NULL)      CREATION-----1994.231
  RELEASE-----2        EXPIRATION-----0000.000
SMSDATA
  STORAGECLASS -----NORMAL      MANAGEMENTCLASS--PRIMARY
  DATACLASS ----- (NULL)      LBACKUP ---0000.000.0000
  BWO STATUS-----00000000      BWO TIMESTAMP---00000 00:00:00.0
  PROTECTION-PSWD----- (NULL)    RACF----- (NO)
ASSOCIATIONS
  DATA-----PJRES1.DFMSAMP.ACCT.DATA
  INDEX-----PJRES1.DFMSAMP.ACCT.INDEX

DATA ----- PJRES1.DFMSAMP.ACCT.DATA
IN-CAT --- SYS1.ICFCAT.VSYS107
HISTORY
  DATASET-OWNER----- (NULL)      CREATION-----1994.231
  RELEASE-----2        EXPIRATION-----0000.000
  PROTECTION-PSWD----- (NULL)    RACF----- (NO)
ASSOCIATIONS
  CLUSTER--PJRES1.DFMSAMP.ACCT
ATTRIBUTES
  KEYLEN-----8          AVGLRECL-----124      BUFSIZE-----45568      CFSIZE-----22528
  RKP-----0            MAXLRECL-----124      EXCEXIT----- (NULL)    CI/CA-----2
  SHROPTNS (1,3)  SPEED  UNIQUE          NOERASE      INDEXED          NOWRITECHK  NOIMBED          NOREPLICAT
  UNORDERED      REUSE  NONSPANNED      BINARY
  CCSID-----0
STATISTICS
  REC-TOTAL-----6        SPLITS-CI-----0      EXCPS-----6
  REC-DELETED-----0      SPLITS-CA-----0      EXTENTS-----1
  REC-INSERTED-----0      FREESPACE-%CI-----0  SYSTEM-TIMESTAMP:
  REC-UPDATED-----2      FREESPACE-%CA-----0  X' A9C01E1F951C1914'
  REC-RETRIEVED-----6    FREESPC-BYTES-----22528
ALLOCATION
  SPACE-TYPE-----TRACK    HI-ALLOC-RBA-----45056
  SPACE-PRI-----1        HI-USED-RBA-----45056
  SPACE-SEC-----0
VOLUME
  VOLSER-----SYS105      PHYREC-SIZE-----22528  HI-ALLOC-RBA-----45056  EXTENT-NUMBER-----1
  DEVTYPE-----X'3010200E' PHYRECS/TRK-----2    HI-USED-RBA-----45056  EXTENT-TYPE-----X'40'
  VOLFLAG-----PRIME      TRACKS/CA-----1
  EXTENTS:
  LOW-CCHH-----X'0011000B' LOW-RBA-----0      TRACKS-----1
  HIGH-CCHH-----X'0011000B' HIGH-RBA-----45055

INDEX ----- PJRES1.DFMSAMP.ACCT.INDEX
IN-CAT --- SYS1.ICFCAT.VSYS107
HISTORY
  DATASET-OWNER----- (NULL)      CREATION-----1994.231
  RELEASE-----2        EXPIRATION-----0000.000
  PROTECTION-PSWD----- (NULL)    RACF----- (NO)
ASSOCIATIONS
  CLUSTER--PJRES1.DFMSAMP.ACCT
ATTRIBUTES
  KEYLEN-----8          AVGLRECL-----0      BUFSIZE-----0      CFSIZE-----512
  RKP-----0            MAXLRECL-----505      EXCEXIT----- (NULL)    CI/CA-----46
  SHROPTNS (1,3)  RECOVERY UNIQUE          NOERASE      NOWRITECHK      NOIMBED      NOREPLICAT      UNORDERED
  REUSE
STATISTICS
  REC-TOTAL-----1        SPLITS-CI-----0      EXCPS-----4      INDEX:
  REC-DELETED-----0      SPLITS-CA-----0      EXTENTS-----1      LEVELS-----1
  REC-INSERTED-----0      FREESPACE-%CI-----0  SYSTEM-TIMESTAMP:      ENTRIES/SECT-----1
  REC-UPDATED-----0      FREESPACE-%CA-----0  X' A9C01E1F951C1914'  SEQ-SET-RBA-----0
  REC-RETRIEVED-----0    FREESPC-BYTES-----23040  HI-LEVEL-RBA-----0
ALLOCATION
  SPACE-TYPE-----TRACK    HI-ALLOC-RBA-----23552
  SPACE-PRI-----1        HI-USED-RBA-----512
  SPACE-SEC-----0
VOLUME
  VOLSER-----SYS105      PHYREC-SIZE-----512  HI-ALLOC-RBA-----23552  EXTENT-NUMBER-----1
  DEVTYPE-----X'3010200E' PHYRECS/TRK-----46  HI-USED-RBA-----512  EXTENT-TYPE-----X'00'
  VOLFLAG-----PRIME      TRACKS/CA-----1
  EXTENTS:
  LOW-CCHH-----X'004A0007' LOW-RBA-----0      TRACKS-----1
  HIGH-CCHH-----X'004A0007' HIGH-RBA-----23551
```

## C.8 Customer Master File with Data Conversion

IDCAMS	SYSTEM SERVICES	TIME: 14:54:48	08/22/94	PAGE	1
KEY OF RECORD - F1F1F1F1F1					
000000	F1F1F1F1 F1F1D2C1 E5C9D5C7 40404040	40404040 40404040 4040E396 99829196	*11111KAVING	T....*	
000020	85999540 40404040 40404040 4040F5F6	F0F040C3 96A3A393 8540D996 81844040	*... 5600 C.... R...	*	
000040	40404040 404040E2 819540D1 96A28540	40404040 40404040 40404040 40404040	* S.. J...	*	
000060	C3C1F9F5 F1F9F3F1 F1F1F1F1 F1F0F100	0007CA00 08001600 0E003200 1D	*CA9519311111101.....	*	
KEY OF RECORD - F1F1F1F1F1F2					
000000	F1F1F1F1 F1F2E2C9 E2E3C140 40404040	40404040 40404040 4040E299 8995A440	*11112SISTA	S....*	
000020	40404040 40404040 40404040 4040F5F6	F0F040C3 96A3A393 8540D996 81844040	* 5600 C.... R...	*	
000040	40404040 404040E2 819540D1 96A28540	40404040 40404040 40404040 40404040	* S.. J...	*	
000060	C3C1F9F5 F1F9F3F1 F1F1F1F1 F2F0F100	0007CA00 08001600 0E003200 1D	*CA9519311111201.....	*	
KEY OF RECORD - F1F1F1F1F1F3					
000000	F1F1F1F1 F1F3D3C1 D9E2D6D5 40404040	40404040 40404040 4040D94B 40C34B40	*11113LARSON	R. C. *	
000020	4DD98195 84A85D40 40404040 4040F2F0	F0F040E6 85A2A340 C68999A2 A340E2A3	*(R....) 2000 W... F.... S.*		
000040	998585A3 404040E6 8995A2A3 969560E2	81938594 40404040 40404040 40404040	*.... W.....-S....	*	
000060	D5C3F2F7 F1F0F4F1 F1F1F1F1 F3F0F100	0007CA00 08001600 0E003200 1D	*NC2710411111301.....	*	
KEY OF RECORD - F1F1F1F1F1F4					
000000	F1F1F1F1 F1F4C2C5 C8D9C5D5 E2404040	40404040 40404040 4040C3A8 95A38889	*11114BEHRENS	C....*	
000020	81404040 40404040 40404040 4040F5F6	F0F040C3 96A3A393 8540D996 81844040	*. 5600 C.... R...	*	
000040	40404040 404040E2 819540D1 96A28540	40404040 40404040 40404040 40404040	* S.. J...	*	
000060	C3C1F9F5 F1F9F3F1 F1F1F1F1 F4F0F100	0007CA00 08001600 0E003200 1D	*CA9519311111401.....	*	
KEY OF RECORD - F1F1F1F1F1F5					
000000	F1F1F1F1 F1F5D7C5 D3C5C740 40404040	40404040 40404040 4040D14B 4040404D	*11115PELEG	J. (*	
000020	D196A288 A4815D40 40404040 4040F8F0	F040D54B 40C69985 84859989 839240C1	*J.....) 800 N. F..... A*		
000040	A58595A4 854040C7 8189A388 8599A282	A4998740 40404040 40404040 40404040	*..... G.....	*	
000060	D4C4F2F0 F8F7F9F1 F1F1F1F1 F5F0F100	0007CA00 08001600 0E003200 1D	*MD2087911111501.....	*	
KEY OF RECORD - F1F1F1F1F1F6					
000000	F1F1F1F1 F1F6D7D6 C7C7C940 40404040	40404040 40404040 4040E54B 40E24B40	*11116POGGI	V. S. *	
000020	4DE58193 85998985 5D040400 4040F5F6	F0F040C3 96A3A393 8540D996 81844040	*(V.....) 5600 C.... R...	*	
000040	40404040 404040E2 819540D1 96A28540	40404040 40404040 40404040 40404040	* S.. J...	*	
000060	C3C1F9F5 F1F9F3F1 F1F1F1F1 F6F9F900	0007CA00 08001600 0E003200 1D	*CA9519311111699.....	*	

## C.9 Account Details File with Data Conversion

IDCAMS	SYSTEM SERVICES	TIME: 14:56:32	08/22/94	PAGE	1
KEY OF RECORD - F1F1F1F1F1F0F1					
000000	F1F1F1F1 F1F1F0F1 E3888540 D281A589	958740C3 96999796 9981A385 40C18383	*11111101T.. K.... C..... A..*		
000020	96A495A3 40404040 40404040 40404040	F5F6F0F0 40C396A3 A3938540 D9968184	*.... 5600 C.... R...	*	
000040	40404040 40404040 40E28195 40D196A2	85404040 40404040 40404040 40404040	* S.. J...	*	
000060	4040C3C1 F9F5F1F9 F3000500 000F0000	07CA0008 0016000E 0032001E	* CA95193.....	*	
KEY OF RECORD - F1F1F1F1F1F2F0F1					
000000	F1F1F1F1 F1F2F0F1 E289A2A3 8140D481	999285A3 89958740 C3969497 8195A840	*11111201S.... M..... C..... *		
000020	40404040 40404040 40404040 40404040	F5F6F0F0 40C396A3 A3938540 D9968184	* 5600 C.... R...	*	
000040	40404040 40404040 40E28195 40D196A2	85404040 40404040 40404040 40404040	* S.. J...	*	
000060	4040C3C1 F9F5F1F9 F3000400 000F0000	07CA0008 0016000E 0032001E	* CA95193.....	*	
KEY OF RECORD - F1F1F1F1F1F3F0F1					
000000	F1F1F1F1 F1F3F0F1 E3888540 D38199A2	969540C4 85A58593 96979485 95A340C3	*11111301T.. L.... D..... C*		
000020	96999748 40404040 40404040 40404040	F3F7F7F0 40D28999 92938585 A240D996	*.... 3770 K..... R.*		
000040	81844040 40404040 40E68995 A2A39695	60E28193 85944040 40404040 40404040	*.. W.....-S....	*	
000060	4040D5C3 F2F7F1F0 F4000000 002F0000	07CA0008 0016000E 0032001E	* NC27104.....	*	
KEY OF RECORD - F1F1F1F1F1F4F0F1					
000000	F1F1F1F1 F1F4F0F1 C2858899 8595A240	D7998995 A38599A2 40C9D5C3 4B404040	*11111401B..... P..... INC. *		
000020	40404040 40404040 40404040 40404040	F5F6F0F0 40C396A3 A3938540 D9968184	* 5600 C.... R...	*	
000040	40404040 40404040 40E28195 40D196A2	85404040 40404040 40404040 40404040	* S.. J...	*	
000060	4040C3C1 F9F5F1F9 F3000300 000F0000	07CA0008 0016000E 0032001E	* CA95193.....	*	
KEY OF RECORD - F1F1F1F1F1F5F0F1					
000000	F1F1F1F1 F1F5F0F1 D4994B40 D7859385	87A240D5 85898788 82969988 96968440	*11111501M.. P.... N..... *		
000020	C1838396 A495A340 40404040 40404040	F8F0F040 D54B40C6 99858485 99898392	*A..... 800 N. F..... *		
000040	40C1A585 95A48540 40C78189 A3888599	A282A499 87404040 40404040 40404040	* A..... G.....	*	
000060	4040D4C4 F2F0F8F7 F9000000 004F0000	07CA0008 0016000E 0032001E	* MD20879.... .....	*	
KEY OF RECORD - F1F1F1F1F1F6F9F9					
000000	F1F1F1F1 F1F6F9F9 D7968787 8940E385	83889589 83819340 E2A49797 9699A340	*11111699P.... T..... S..... *		
000020	C995834B 40404040 40404040 40404040	F5F6F0F0 40C396A3 A3938540 D9968184	*I... 5600 C.... R...	*	
000040	40404040 40404040 40E28195 40D196A2	85404040 40404040 40404040 40404040	* S.. J...	*	
000060	4040C3C1 F9F5F1F9 F3000000 100F0000	07CA0008 0016000E 0032001E	* CA95193.....	*	

## C.10 Customer Master File without Data Conversion

IDCAMS	SYSTEM SERVICES	TIME: 10:31:49	08/19/94	PAGE	1
KEY OF RECORD - 313131313131					
000000	31313131 31314E45 54544C45 53202020	20202020 20202020 2020CA07 08001300	*.....+...<.....*		
000020	0A000000 2100572E 20422E20 2842696C	6C292020 20202020 20203536 30302043	*.....%.....*		
000040	6F74746C 6520526F 61642020 20202020	20202053 616E204A 6F736520 20202020	*?...%...?/...../>..?.....*		
000060	20202020 20202020 20202020 43413935	31393331 31313131 313031	*.....*		
KEY OF RECORD - 313131313132					
000000	31313131 31325349 53544120 20202020	20202020 20202020 2020CA07 08001300	*.....*		
000020	0A000200 0D005372 696E7520 20202020	20202020 20202020 20203536 30302043	*.....>.....*		
000040	6F74746C 6520526F 61642020 20202020	20202053 616E204A 6F736520 20202020	*?...%...?/...../>..?.....*		
000060	20202020 20202020 20202020 43413935	31393331 31313131 323031	*.....*		
KEY OF RECORD - 313131313133					
000000	31313131 31334C41 52534F4E 20202020	20202020 20202020 2020CA07 08001300	*.....<... +.....*		
000020	0A000100 1D00522E 20432E20 2852616E	64792920 20202020 20203230 31205765	*...../>.....*		
000040	73742046 69727374 20537472 65657420	20202057 696E7374 6F6E2D53 616C656D	*.....>..?>./%_..*		
000060	20202020 20202020 20202020 4E433237	31303431 31313131 333031	*.....+.....*		
KEY OF RECORD - 313131313134					
000000	31313131 31344241 53534554 54202020	20202020 20202020 2020CA07 08001300	*.....*		
000020	0A000000 22004A2E 20542E20 284A6566	66292020 20202020 20203830 30204E2E	*.....+..*		
000040	20467265 64657269 636B2041 76656E75	65202047 61697468 65727362 75726720	*.....>...../.....*		
000060	20202020 20202020 20202020 4D443230	38373931 31313131 343031	*.....(.....*		
KEY OF RECORD - 313131313135					
000000	31313131 3135524F 47455253 20202020	20202020 20202020 2020CA07 08001300	*..... .....*		
000020	0A000000 2200522E 20452E20 28426F62	29202020 20202020 20203830 30204E2E	*.....?.....+..*		
000040	20467265 64657269 636B2041 76656E75	65202047 61697468 65727362 75726720	*.....>...../.....*		
000060	20202020 20202020 20202020 4D443230	38373931 31313131 353031	*.....(.....*		
KEY OF RECORD - 313131313136					
000000	31313131 3136504F 47474920 20202020	20202020 20202020 2020CA07 08001300	*.....& .....*		
000020	0A000000 2200562E 20532E20 2856616C	65726965 29202020 20203536 30302043	*...../.....*		
000040	6F74746C 6520526F 61642020 20202020	20202053 616E204A 6F736520 20202020	*?...%...?/...../>..?.....*		
000060	20202020 20202020 20202020 43413935	31393331 31313131 363939	*.....*		

## C.11 Account Details File without Data Conversion

IDCAMS	SYSTEM SERVICES	TIME: 11:30:14	08/19/94	PAGE	1
KEY OF RECORD - 3131313131313031					
000000	31313131 31313031 20A10700 CA070800	13000A00 00002200 54686520 4E657474	*.....+...*		
000020	6C657320 436F7270 6F726174 65204163	636F756E 74202020 20202020 20202020	*%....?..?./.....?>.....*		
000040	35363030 20436F74 746C6520 526F6164	20202020 20202020 2053616E 204A6F73	*.....?..%...?/...../>..?.....*		
000060	65202020 20202020 20202020 20202020	20204341 39353139 33	*.....*		
KEY OF RECORD - 3131313131323031					
000000	31313131 31323031 409C0000 CA070800	13000A00 02000900 53697374 61204D61	*...../.....(/*		
000020	726B6574 696E6720 436F6D70 616E7920	20202020 20202020 20202020 20202020	*.....>...?_./>.....*		
000040	35363030 20436F74 746C6520 526F6164	20202020 20202020 2053616E 204A6F73	*.....?..%...?/...../>..?.....*		
000060	65202020 20202020 20202020 20202020	20204341 39353139 33	*.....*		
KEY OF RECORD - 3131313131333031					
000000	31313131 31333031 68870100 CA070800	13000A00 01001400 54686520 4C617273	*.....</.....*		
000020	6F6E2044 6576656C 6F706D65 6E742043	6F72702E 20202020 20202020 20202020	*?>....%?_>...?.....*		
000040	33373730 20486972 6B6C6565 7320526F	61642020 20202020 2057696E 73746F6E	*.....%.....?/.....>..?>..*		
000060	2D53616C 656D2020 20202020 20202020	20204E43 32373130 34	*../%_.....+.....*		
KEY OF RECORD - 3131313131343031					
000000	31313131 31353031 E0930400 CA070800	13000A00 00002300 42617373 65747420	*...../.....*		
000020	5072696E 74657273 20494E43 2E202020	20202020 20202020 20202020 20202020	*&..>.....+.....*		
000040	38303020 4E2E2046 72656465 7269636B	20417665 6E756520 20476169 74686572	*.....+.....>...../.....*		
000060	73627572 67202020 20202020 20202020	20204D44 32303837 39	*.....(.....*		
KEY OF RECORD - 3131313131353031					
000000	31313131 31353031 04000000 CA070800	13000A00 00002300 4D722E20 526F6765	*.....(.....?..*		
000020	7273204E 65696768 626F7268 6F6F6420	4163636F 756E7420 20202020 20202020	*...+.....?..??.....?>.....*		
000040	38303020 4E2E2046 72656465 7269636B	20417665 6E756520 20476169 74686572	*.....+.....>...../.....*		
000060	73627572 67202020 20202020 20202020	20204D44 32303837 39	*.....(.....*		
KEY OF RECORD - 3131313131363939					
000000	31313131 31363939 64000000 CA070800	13000A00 00002300 506F6767 69205465	*.....&?.....*		
000020	63686E69 63616C20 53757070 6F727420	496E632E 20202020 20202020 20202020	*..>../%.....?.....>.....*		
000040	35363030 20436F74 746C6520 526F6164	20202020 20202020 2053616E 204A6F73	*.....?..%...?/...../>..?.....*		
000060	65202020 20202020 20202020 20202020	20204341 39353139 33	*.....*		



---

## Appendix D. PL/I Sample Application Listings

This appendix contains the following listings from our implementation of the PL/I sample application:

- PLILOAD source code
- PLIREAD source code
- PLIMAIN source code
- LISTCAT output of PL/I data set after running sample
- PL/I data set after running sample.

The programs are based on a set of PL/I samples written by Mansoor Dawoodbhoy, IBM Storage Systems Division.

---

### D.1 PLILOAD Source Code

```
/* **** */
/* NAME - PLILOAD.PLI */
/* */
/* DESCRIPTION */
/* Load a DDM KEYED INDEXED data set. */
/* */
/* USAGE */
/* The following OS/2 SET commands are required to establish */
/* the environment variables to run this program: */
/* */
/* SET DD:SYSIN=IBMBBOOKS.CAT,RECSIZE(81) */
/* SET DD:DIREC=drive:dsn,AMTHD(DDM) */
/* drive - one character drive ID, local or assigned */
/* with DFMDRIVE */
/* dsn - valid file name or data set name for the */
/* system assigned to drive */
/* **** */
*****
PROG01: proc options(main);

    dcl Direc file record direct output
        env(organization(indexed)
            recsize(80)
            keyloc(1)
            keylength(14)
        );

    dcl Eof bit(1) init('0'b),
        On52 bit(1) init('0'b),
        i fixed(15),
        j fixed(15),
        oncode builtin;

    dcl 1 IoArea,
        5 FormNo char(14),
        5 Date char(9),
        5 Title char(57);

    dcl Filler1 char(1);

    on endfile(sysin) Eof = '1'b;

    on key(Direc)
    begin;
        if oncode=52
            then On52 = '1'b;
    end;

    open file(Direc);

    get file(sysin) edit(FormNo,Date,Title,Filler1)
        (a(14),a(9),a(57),a(1));
```

```

i = 0; j = 1;

do while (-Eof);
  write file(Direc) from(IoArea) keyfrom(FormNo);
  if (-On52)
    then begin;
      i = i+1;
      put skip edit(IoArea)(a);
    end;
  else begin;
    put skip edit('FormNo [' ,FormNo,'] is a duplicate <KEY>')(a);
    put skip;
    On52 = '0'b;
  end;
  get file(sysin) edit(FormNo,Date,Title,filler1)
    (a(14),a(9),a(57),a(1));
  j = j+1;
end;

put skip;
put skip edit(i,' Total records processed ')(a);
put skip edit(j-1,' Total records read ')(a);

close file(Direc);

end PROG01;

```

---

## D.2 PLIREAD Source Code

```

/*****
/*
/* NAME - PLIREAD.PLI
/*
/* DESCRIPTION
/* Read a DDM KEYED INDEXED data set.
/*
/* USAGE
/* The following OS/2 SET commands are required to establish
/* the environment variables to run this program:
/*
/* SET DD:DIREC=drive:dsn,AMTHD(DDM)
/* drive - one character drive ID, local or assigned
/* with DFMDRIVE
/* dsn - valid file name or data set name for the
/* system assigned to drive
/*
*****/

PROG01: proc options(main);

  dcl Direc file record sequential input
    env(organization(indexed)
      recsize(80)
      keyloc(1)
      keylength(14)
    );

  dcl Eof bit(1) init('0'b),
    i fixed(15);

  dcl 1 IoArea,
    5 FormNo char(14),
    5 Date char(9),
    5 Title char(57);

  on endfile(Direc) Eof = '1'b;

  open file(Direc);

  read file(Direc) into(IoArea);
  i = 1;

  do while(-Eof);
    put file(sysprint) skip edit(IoArea) (a);
    read file(Direc) into(IoArea);
    i = i+1;
  end;

  close file(Direc);

  put skip;

```



```

        put skip edit(i-1,' Total records processed ')(a);
end PROG01;

```

### D.3 PLIMAINT Source Code

```

/*****
/*
/* NAME - PLIMAINT.PLI
/*
/* DESCRIPTION
/* Update a KEYED INDEXED data set by key.
/*
/* USAGE
/* The following OS/2 SET commands are required to establish
/* the environment variables to run this program:
/*
/* SET DD:SYSIN=MAINTCAT.CAT,RECSIZE(83)
/* SET DD:DIREC=drive:dsn,AMTHD(DDM)
/* drive - one character drive ID, local or assigned
/* with DFMDRIVE
/* dsn - valid file name or data set name for the
/* system assigned to drive
/*
*****/

PROG01: proc options(main);

    dcl Direc file record direct update
        env(organization(indexed)
            recsize(80)
            keyloc(1)
            keylength(14)
        );

    dcl 1 IoArea,
        5 FormNo char(14),
        5 Date char(9),
        5 Title char(57);

    dcl 1 Action char(3);
    dcl 1 WorkArea char(80);

    dcl oncode builtin;
    dcl Eof bit(1) init('0'b),
        On51 bit(1) init('0'b),
        On52 bit(1) init('0'b);

    on endfile(sysin) Eof = '1'b;

    on key(Direc)
    begin;
        if oncode=51 then
            On51='1'b;
        if oncode=52 then
            On52='1'b;
    end;

    open file(Direc) direct update;

    get file(sysin) edit(FormNo,Date,Title,Action)
        (a(14),a(9),a(57),a(3));

    do while (-Eof);

        select (Action);

            when('ADD')
            begin;
                write file(Direc) from(IoArea) keyfrom(FormNo);
                if (-On52) then
                begin;
                    put file(sysprint) skip edit('Adding ...')(a(10));
                    put file(sysprint) skip edit(FormNo,Date,Title)
                        (a(14),a(9),a(57));
                end;
            else
            begin;
                put skip edit('FormNo [',FormNo,'] is a duplicate <KEY>')(a);
                put file(sysprint) skip
                    edit('Record will not be added')(a);
            end;
        end;
    end;

```

```

        put file(sysprint) skip;
        0n52='0'b;
        end;
    end;
when('UPD')
begin;
    rewrite file(Direc) from(IoArea) key(FormNo);
    if (-0n51) then
    begin;
        put file(sysprint) skip edit('Updating .')(a(10));
        put file(sysprint) skip edit(FormNo,Date,Title)
            (a(14),a(9),a(57));
    end;
    else
    begin;
        put skip edit('FormNo [',FormNo,'] is not on file')(a);
        put file(sysprint) skip
            edit('Record will not be Updated')(a);
        put file(sysprint) skip;
        0n51='0'b;
        end;
    end;

when('DEL')
begin;
    read file(Direc) into(WorkArea) key(FormNo);
    if (-0n51) then
    begin;
        put file(sysprint) skip edit('Deleting .')(a(10));
        put file(sysprint) skip edit(WorkArea)(a);
        delete file(Direc) key(FormNo);
    end;
    else
    begin;
        put skip edit('FormNo [',FormNo,'] is not on file')(a);
        put file(sysprint) skip
            edit('Record will not be deleted')(a);
        put file(sysprint) skip;
        0n51='0'b;
        end;
    end;

otherwise
begin;
    put file(sysprint) skip(1)
        edit(FormNo,' invalid processing code .....')(a);
    put file(sysprint) skip
        edit('Record will not be processed')(a);
    put file(sysprint) skip;
end;

end;

get file(sysin) edit(FormNo,Date,Title,Action)
    (a(14),a(9),a(57),a(3));

end;

close file(Direc);

end PROG01;

```

---

## D.4 ADL for Host View of Books Catalog Record

```
/* Base View ADL for the Books Catalog for the PLI demo */
/* File name: DDMOUTB.ADL */
DECLARE
BEGIN;
  letters:
    SUBTYPE OF CHAR LENGTH(81) CCSID(500);
  PUBREC:
    SEQUENCE
    BEGIN;
      PUBNO: letters LENGTH(14);
      DATE:  letters LENGTH(9);
      TITLE: letters LENGTH(56);
      FILLER: letters LENGTH(1);
    END;
END;
```

---

## D.5 ADL for Application View of Books Catalog Record

```
/* Application View ADL for the Books Catalog for the PLI demo */
/* File name : DDMOUTV.ADL */
DECLARE
BEGIN;
  letters:
    SUBTYPE OF CHAR LENGTH(81) CCSID(437);
  PUBREC:
    SEQUENCE
    BEGIN;
      PUBNO: letters LENGTH(14);
      DATE:  letters LENGTH(9);
      TITLE: letters LENGTH(56);
      FILLER: letters LENGTH(1);
    END;
END;
```

## D.6 LISTCAT Output of PL/I Data Set after Running Sample

IDCAMS SYSTEM SERVICES

TIME: 11:33:12

08/24/94

PAGE

1

```
CLUSTER ----- PJRES1.PLISAMP.DATA
IN-CAT --- SYS1.ICFCAT.VSYS366
HISTORY
  DATASET-OWNER----- (NULL)      CREATION-----1994.235
  RELEASE-----2      EXPIRATION-----0000.000
SMSDATA
  STORAGECLASS -----NORMAL      MANAGEMENTCLASS--PRIMARY
  DATACLASS ----- (NULL)      LBACKUP ---1994.236.0125
  BWO STATUS-----00000000      BWO TIMESTAMP---00000 00:00:00.0
  PROTECTION-PSWD----- (NULL)    RACF----- (NO)
ASSOCIATIONS
  DATA-----PJRES1.PLISAMP.DATA.DATA
  INDEX-----PJRES1.PLISAMP.DATA.INDEX

DATA ----- PJRES1.PLISAMP.DATA.DATA
IN-CAT --- SYS1.ICFCAT.VSYS366
HISTORY
  DATASET-OWNER----- (NULL)      CREATION-----1994.235
  RELEASE-----2      EXPIRATION-----0000.000
  PROTECTION-PSWD----- (NULL)    RACF----- (NO)
ASSOCIATIONS
  CLUSTER--PJRES1.PLISAMP.DATA
ATTRIBUTES
  KEYLEN-----14      AVGLRECL-----80      BUFSIZE-----45568      CFSIZE-----22528
  RKP-----0      MAXLRECL-----80      EXCEXIT----- (NULL)    CI/CA-----2
  SHROPTNS (1,3)  RECOVERY  UNIQUE      NOERASE      INDEXED      NOWRITECHK      NOIMBED      NOREPLICAT
  UNORDERED      NOREUSE      NONSPANNED
  CCSID-----65535
STATISTICS
  REC-TOTAL-----10      SPLITS-CI-----0      EXCPS-----36
  REC-DELETED-----2      SPLITS-CA-----0      EXTENTS-----1
  REC-INSERTED-----10      FREESPACE-%CI-----0      SYSTEM-TIMESTAMP:
  REC-UPDATED-----2      FREESPACE-%CA-----0      X' A9C671CF13B73381'
  REC-RETRIEVED-----50      FREESPC-BYTES-----22528
ALLOCATION
  SPACE-TYPE-----TRACK      HI-ALLOC-RBA-----45056
  SPACE-PRI-----1      HI-USED-RBA-----45056
  SPACE-SEC-----1
VOLUME
  VOLSER-----SYS300      PHYREC-SIZE-----22528      HI-ALLOC-RBA-----45056      EXTENT-NUMBER-----1
  DEVTYPE-----X'3010200E'      PHYRECS/TRK-----2      HI-USED-RBA-----45056      EXTENT-TYPE-----X'00'
  VOLFLAG-----PRIME      TRACKS/CA-----1
  EXTENTS:
  LOW-CCHH-----X'0023000B'      LOW-RBA-----0      TRACKS-----1
  HIGH-CCHH-----X'0023000B'      HIGH-RBA-----45055

INDEX ----- PJRES1.PLISAMP.DATA.INDEX
IN-CAT --- SYS1.ICFCAT.VSYS366
HISTORY
  DATASET-OWNER----- (NULL)      CREATION-----1994.235
  RELEASE-----2      EXPIRATION-----0000.000
  PROTECTION-PSWD----- (NULL)    RACF----- (NO)
ASSOCIATIONS
  CLUSTER--PJRES1.PLISAMP.DATA
ATTRIBUTES
  KEYLEN-----14      AVGLRECL-----0      BUFSIZE-----0      CFSIZE-----512
  RKP-----0      MAXLRECL-----505      EXCEXIT----- (NULL)    CI/CA-----46
  SHROPTNS (1,3)  RECOVERY  UNIQUE      NOERASE      NOWRITECHK      NOIMBED      NOREPLICAT      UNORDERED
  NOREUSE
STATISTICS
  REC-TOTAL-----1      SPLITS-CI-----0      EXCPS-----7      INDEX:
  REC-DELETED-----0      SPLITS-CA-----0      EXTENTS-----1      LEVELS-----1
  REC-INSERTED-----0      FREESPACE-%CI-----0      SYSTEM-TIMESTAMP:      ENTRIES/SECT-----1
  REC-UPDATED-----0      FREESPACE-%CA-----0      X' A9C671CF13B73381'      SEQ-SET-RBA-----0
  REC-RETRIEVED-----0      FREESPC-BYTES-----23040      HI-LEVEL-RBA-----0
ALLOCATION
  SPACE-TYPE-----TRACK      HI-ALLOC-RBA-----23552
  SPACE-PRI-----1      HI-USED-RBA-----512
  SPACE-SEC-----1
VOLUME
  VOLSER-----SYS300      PHYREC-SIZE-----512      HI-ALLOC-RBA-----23552      EXTENT-NUMBER-----1
  DEVTYPE-----X'3010200E'      PHYRECS/TRK-----46      HI-USED-RBA-----512      EXTENT-TYPE-----X'00'
  VOLFLAG-----PRIME      TRACKS/CA-----1
  EXTENTS:
  LOW-CCHH-----X'00270003'      LOW-RBA-----0      TRACKS-----1
  HIGH-CCHH-----X'00270003'      HIGH-RBA-----23551
```

## D.7 PL/I Sample Data Set after Running Sample

IDCAMS	SYSTEM SERVICES	TIME: 14:42:14	08/23/94	PAGE	1
KEY OF RECORD - 5BC4C2F240404040F040404040					
000000 5BC4C2F2 40404040 40F04040 404040F9	F361F0F2 61F1F640 C4C261F2 40D78599	*\$DB2	0	93/02/16 DB/2 P..*	
000020 86969994 81958385 40D49695 89A39699	40D9F24B F140D389 82998199 A8404040	*..... M..... R2.1 L.....		*	
000040 40404040 40404040 40404040 40404040		*		*	
KEY OF RECORD - 5BD9C5D3F2C1C3C940F24BF04040					
000000 5BD9C5D3 F2C1C3C9 40F24BF0 404040F9	F361F0F2 61F1F640 C1E261F4 F0F040C3	*\$REL2ACI 2.0	93/02/16 AS/400 C*		
000020 969494A4 95898381 A3899695 A240D389	82998199 A840D985 938581A2 8540F24B	*..... L..... R..... 2.*		*	
000040 F0404040 40404040 40404040 40404040		*0		*	
KEY OF RECORD - C4D4E2C5F6C1E9E940F140404040					
000000 C4D4E2C5 F6C1E9E9 40F14040 404040F9	F361F0F2 61F1F740 E4E2C9D5 C740D6D5	*DMSE6AZZ 1	93/02/17 USING ON*		
000020 D3C9D5C5 40C4D6C3 E4D4C5D5 E3C1E3C9	D6D54040 40404040 40404040 40404040	*LINE DOCUMENTATION		*	
000040 40404040 40404040 40404040 40404040		*		*	
KEY OF RECORD - C7C7F2F4F3F5F4F940F140404040					
000000 C7C7F2F4 F3F5F4F9 40F14040 404040F9	F361F0F2 61F1F740 E4A28995 8740D585	*GG243549 1	93/02/17 U.... N.*		
000020 A3E58985 A640D9F3 40D7D7C9 40404040	40404040 40404040 40404040 40404040	*.V... R3 PPI		*	
000040 40404040 40404040 40404040 40404040		*		*	
KEY OF RECORD - C7C7F2F4F3F7F2F040F140404040					
000000 C7C7F2F4 F3F7F2F0 40F14040 404040F9	F361F0F2 61F1F740 C5E2C3D6 D540D481	*GG243720 1	93/02/17 ESCON M.*		
000020 95818785 9940D799 85A28595 A381A389	969540C7 99819788 8983A240 40404040	*..... P..... G.....		*	
000040 40404040 40404040 40404040 40404040		*		*	
KEY OF RECORD - C7C7F2F4F3F7F5F140F040404040					
000000 C7C7F2F4 F3F7F5F1 40F04040 404040F9	F361F0F2 61F1F740 D7D6E6C5 D940D585	*GG243751 0	93/02/17 POWER N.*		
000020 A3A69699 92899587 40C7A489 84854040	40404040 40404040 40404040 40404040	*..... G....		*	
000040 40404040 40404040 40404040 40404040		*		*	
KEY OF RECORD - C7C7F2F4F3F9F1F740F040404040					
000000 C7C7F2F4 F3F9F1F7 40F04040 404040F9	F361F0F2 61F1F740 D3C1D5C4 D740D585	*GG243917 0	93/02/17 LANDP N.*		
000020 A3A69699 92899587 40A689A3 8840C1D7	D7D54040 40404040 40404040 40404040	*..... APPN		*	
000040 40404040 40404040 40404040 40404040		*		*	
KEY OF RECORD - C7C7F2F4F4F0F8F740F040404040					
000000 C7C7F2F4 F4F0F8F7 40F04040 404040F9	F361F0F9 61F1F540 E4A28995 8740D5C6	*GG244087 0	93/09/15 U.... NF*		
000020 E2408995 408140D4 A493A389 A5859584	969940C5 95A58999 96959485 95A34040	*S .. . M..... E.....		*	
000040 40404040 40404040 40404040 40404040		*		*	
KEY OF RECORD - C7C7F6F6F3F2F3F940F14BF04040					
000000 C7C7F6F6 F3F2F3F9 40F14BF0 404040F9	F361F0F2 61F1F740 C5E2C3D6 D540D697	*GG663239 1.0	93/02/17 ESCON O.*		
000020 859981A3 969940D7 99968240 C485A385	99408995 40A38885 40D4E5E2 61C5E2C1	*..... P... D.... .. MVS/ESA*			
IDCAMS SYSTEM SERVICES	TIME: 14:42:14	08/23/94	PAGE	2	
000040 40C595A5 89999695 948595A3 40404040		* E.....		*	
KEY OF RECORD - C9C2D4C2D2F0F0F540F040404040					
000000 C9C2D4C2 D2F0F0F5 40F04040 404040F9	F361F1F0 61F0F540 D7E261F2 6B40D6E2	*IBMBK005 0	93/10/05 PS/2, OS*		
000020 61F26B40 D7C36B40 D7998995 A385996B	40D49684 85944081 958440E7 4BF2F540	* /2, PC, P....., M.... .. X.25		*	
000040 C381A381 93968740 40404040 40404040		*C.....		*	



## List of Abbreviations

<b>ACF/VTAM</b>	Advanced Communications Function/Virtual Telecommunications Access Method	<b>ISMF</b>	Interactive Storage Management Facility
<b>ACS</b>	automatic class selection	<b>ITSO</b>	International Technical Support Organization
<b>ADL</b>	A Data Language	<b>LAN</b>	local area network
<b>ADSM</b>	ADSTAR Distributed Storage Manager	<b>LDMI</b>	local data management interface
<b>API</b>	application program interface	<b>LU</b>	logical unit
<b>APPC</b>	Advanced Program-to-Program Communication	<b>MVS</b>	Multiple Virtual Storage
		<b>MVS/ESA</b>	Multiple Virtual Storage/Enterprise Systems Architecture
<b>ASCH</b>	APPC/MVS scheduler	<b>NDF</b>	network definition file
<b>ASCII</b>	American National Standard Code for Information Interchange	<b>OS/2</b>	Operating System/2
		<b>OS/400</b>	Operating System/400
<b>AS/400</b>	Application System/400	<b>PDSE</b>	partitioned data set extended
<b>CCSID</b>	coded character set identifier	<b>PM</b>	Presentation Manager
<b>CICS</b>	Customer Information Control System	<b>PS/2</b>	IBM Personal System/2
		<b>PU</b>	physical unit
<b>CM/2</b>	Communications Manager/2	<b>PWS</b>	programmable workstation
<b>DAS</b>	Data Access Services	<b>RACF</b>	Resource Access Control Facility
<b>DASD</b>	Direct Access Storage Device	<b>REXX</b>	Restructured Extended Executor
<b>DDF</b>	data description file	<b>RU</b>	request/response unit
<b>DDM</b>	Distributed Data Management	<b>SAA</b>	Systems Application Architecture
<b>DFM</b>	Distributed FileManager	<b>SMS</b>	Storage Management Subsystem
<b>DFM/MVS</b>	Distributed FileManager/MVS	<b>SNA</b>	Systems Network Architecture
<b>DFM/2</b>	Distributed FileManager/2	<b>TP</b>	transaction program/process
<b>DFSMS/MVS</b>	Data Facility Storage Management Subsystem/MVS	<b>TRLAN</b>	token ring local area network
<b>DLL</b>	data link library	<b>TSO</b>	Time Sharing Option
<b>EBCDIC</b>	extended binary-coded decimal interchange code	<b>VSAM</b>	Virtual Storage Access Method
<b>FAT</b>	file allocation table	<b>VTAM</b>	Virtual Telecommunications Access Method
<b>HPFS</b>	high performance file system		
<b>IBM</b>	International Business Machines Corporation		





---

## Index

### A

a data language  
    *See* ADL  
ACF/VTAM 3, 5, 8—10  
ACS 10  
ADL 33, 35, 61, 66, 75, 102, 103, 113  
ADSM 5  
ADSTAR Distributed Storage Manager  
    *See* ADSM  
Advanced Communications Function/Virtual  
    Telecommunications Access Method  
    *See* ACF/VTAM  
Advanced Program-to-Program Communication  
    *See* APPC  
American National Standard Code for Information  
    Interchange  
    *See* ASCII  
API 4, 13  
APPC 1, 3, 6, 7, 9, 11, 13, 16  
APPC/MVS 3, 5, 7, 8—10, 17  
APPC/MVS scheduler  
    *See* ASCH  
application program interface  
    *See* API  
Application System/400  
    *See* AS/400  
AS/400 30, 51—71  
ASCH 3, 7, 9  
ASCII 22, 26, 33, 38, 50  
automatic class selection  
    *See* ACS

### B

base LU 8

### C

C application  
    customization 33  
    description 30  
    execution scenarios 35  
    installation 32  
CCSID 26, 36, 45  
CICS/MVS 1  
CICS/VSE 1  
client/server 1—3  
CM/2 7, 10, 20, 22, 55—57  
coded character set identifier  
    *See* CCSID  
command  
    CRTFLR (AS/400) 54  
    DFMDRIVE 27, 35, 41, 46  
    DSPNETA (AS/400) 51

command (*continued*)

    STRTDFMC 26, 27, 34, 40, 66  
    STRTDFMR 27, 29, 34, 40  
    WRKLIND (AS/400) 52  
Communications Manager/2  
    *See* CM/2

### D

DAS 5, 39  
DASD 46  
Data Access Services  
    *See* DAS  
data conversion 26, 33, 35, 38, 40, 50, 61, 66, 67, 75,  
    106, 107  
data description file  
    *See* DDF  
Data Facility Storage Management Subsystem/MVS  
    *See* DFSMS/MVS  
data link library  
    *See* DLL  
DDF 33, 34  
DDM 1—3, 30, 35, 45, 51, 75  
DDM attributes 45  
definitions  
    ACF/VTAM 5  
    ACS routines 10  
    APPC/MVS 7  
    Communications Manager/2 11, 55  
    DFM/2 25, 60  
    DFM/MVS 9  
    LU 6  
DFSMS/MVS 1, 4, 5, 10, 30  
Direct Access Storage Device  
    *See* DASD  
Distributed Data Management  
    *See* DDM  
Distributed FileManager  
    communication 5  
    components 2  
    data flow 2  
    introduction 1  
    working environment 4  
DLL 4, 39  
DSESLIM parameter 6

### E

EBCDIC 26, 33, 38  
extended binary-coded decimal interchange code  
    *See* EBCDIC

## F

FAT 47  
file allocation table  
    *See* FAT  
Freelance Graphics 69, 70

## H

high performance file system  
    *See* HPFS  
HPFS 47

## I

IBM Personal System/2  
    *See* PS/2  
independent LU 6  
Interactive Storage Management Facility  
    *See* ISMF  
ISMF 36, 45

## K

keywords  
    DEFAULT\_CCSID 26  
    DEFAULT\_CONVERSION\_TABLE 26  
    DEFAULT\_DFM\_TARGET 26  
    DFM\_TARGET 25, 60  
    FILE\_DESCRIPTOR\_MAP 34—36, 40, 41, 61, 65  
    MODE\_NAME 26

## L

LDMI 3  
local data management interface  
    *See* LDMI  
logical unit  
    *See* LU  
logmode  
    QPCSUPP 7, 20, 26  
LU 3, 5—8, 15, 17, 18, 55, 56

## M

Multiple Virtual Storage/Enterprise Systems  
    Architecture  
        *See* MVS/ESA  
MVS/ESA 1, 5, 7, 17, 27, 33, 41, 61

## N

NDF 12, 22, 58  
network definition file  
    *See* NDF

## O

Operating System/2  
    *See* OS/2

Operating System/400  
    *See* OS/400  
OS/2 1, 3, 5, 10, 20, 22, 27, 30, 51—55, 62, 68, 69  
OS/400 1, 4, 51

## P

partitioned data set extended  
    *See* PDSE  
partner LU 17, 18, 25, 52, 55, 57  
PDSE 46—50  
physical unit  
    *See* PU  
PL/I application  
    customization 40  
    description 39  
    execution scenarios 41  
    installation 39  
PM 30  
Presentation Manager  
    *See* PM  
programmable workstation  
    *See* PWS  
PS/2 5  
PU 6  
PWS 5, 6, 10, 15, 18, 25, 26, 29, 33, 35, 39, 42, 46—56

## Q

QPCSUPP 7, 20, 26

## R

RACF 3  
request/response unit  
    *See* RU  
Resource Access Control Facility  
    *See* RACF  
Restructured Extended Executor  
    *See* REXX  
REXX 39—43  
RU 21

## S

SAA 1  
sample application  
    C 30  
    PL/I 39  
sample diskette 71  
SMS 10  
SNA 14, 22, 57  
source listing  
    C program 75  
    PL/I program 109  
source system 1—4, 25, 51  
Storage Management Subsystem  
    *See* SMS

stream I/O 46, 47, 67  
Systems Application Architecture  
    *See* SAA  
Systems Network Architecture  
    *See* SNA

## **T**

target system 1—4, 25, 27, 34, 36, 39, 41, 51, 53, 55  
Time Sharing Option  
    *See* TSO  
token ring local area network  
    *See* TRLAN  
TP 8  
transaction program  
    *See* TP  
TRLAN 5, 6, 51, 52  
TSO 49

## **V**

Virtual Storage Access Method  
    *See* VSAM  
VSAM 8, 30, 36, 41, 43  
VSAM/2 3, 5, 30, 39, 61



**International Technical Support Organization  
Easy Access to Host Data with Distributed FileManager  
November 1994**

**Publication No. GG24-4427-00**

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)**

<b>Overall Satisfaction</b>	_____	
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



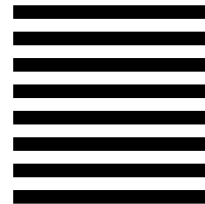
## BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Organization  
Department 471, Building 070B  
5600 COTTLE ROAD  
SAN JOSE CA  
USA 95193-0001

NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES



Fold and Tape

Please do not staple

Fold and Tape





Printed in U.S.A.

GG24-4427-00

