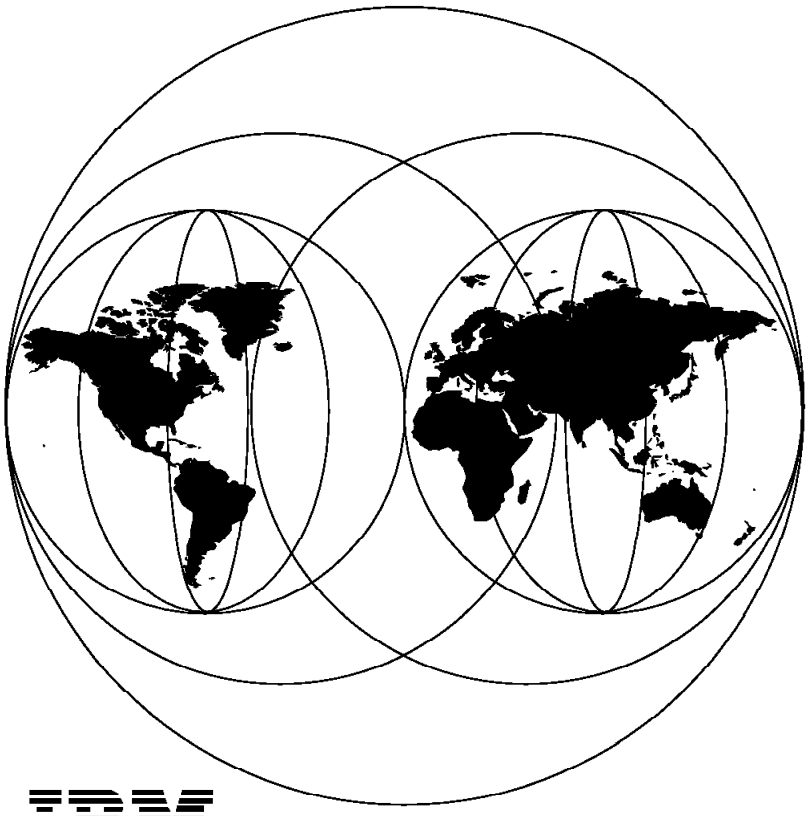
International Technical Support Organization

SG24-4529-00

MVS/ESA SP 5.2.2 OpenEdition MVS Installation and Customization Starter Kit

December 1995



IBM

International Technical Support Organization
Poughkeepsie Center

International Technical Support Organization

SG24-4529-00

MVS/ESA SP 5.2.2 OpenEdition MVS Installation and Customization Starter Kit

December 1995

Take Note!

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

First Edition (December 1995)

This edition applies to OpenEdition MVS for Version 5 Release 2 Modification 2 of MVS/ESA System Product (5655-068 and 5655-069).

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. 541 Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

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.

Abstract

This publication provides information to facilitate the installation and use of MVS/ESA SP 5.2.2 OpenEdition MVS. It provides step by step descriptions on how to install, customize and use the OpenEdition MVS product set. Step by step descriptions are also provided to help the reader customize DFSMS/MVS, APPC, RACF, TCP/IP and NFS to be used in an OpenEdition MVS environment. A diskette is included that contains the machine readable sample JCL and sample programs that are referenced in this book.

This document was written for MVS system programmers who will install and customize the OpenEdition MVS product set. Some knowledge of OpenEdition MVS is assumed.

Contents

Figures

Tables

Special Notices

This publication is intended to help the customer install, customize, operate, manage and maintain an MVS system with OpenEdition MVS. The information in this publication is not intended as the specification of any programming interfaces that are provided by OpenEdition MVS System Services, OpenEdition MVS Application Services, OpenEdition MVS Shell and Utilities and the OpenEdition MVS Debugger. See the PUBLICATIONS section of the IBM Programming Announcement for OpenEdition MVS for more information about what publications are considered to be product documentation.

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

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

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

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

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:

AD/Cycle
C/370
DFSMS
DFSMS/MVS
DFSMSdfp
DFSMSdss
IBM
Language Environment
MVS/ESA
OpenEdition
OS/2
RACF
RMF
VTAM

The following terms are trademarks of other companies:

Windows is a trademark of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

CA-ACF2 and CA-Top Secret are registered trademarks of Computer Associates International, Inc.

C-bus is a trademark of Corollary, Inc.

Other trademarks are trademarks of their respective companies.

Preface

This document is a guide to installing and customizing OpenEdition MVS.

This document is intended primarily for MVS systems programmers who are responsible for planning, installing, and customizing the OpenEdition MVS product set on MVS/ESA. Some familiarity with the MVS operating environment, OpenEdition MVS, DFSMS/MVS, RACF, TCP/IP and NFS is assumed.

How This Document Is Organized

The document is organized as follows:

- Chapter 1, “Introduction to OpenEdition MVS”
- Chapter 2, “Beginning the Installation”

  This chapter provides the information required to install the OpenEdition MVS System Services component and customize DFSMS/MVS, APPC and RACF for OpenEdition MVS.
- Chapter 3, “Installation of the OpenEdition MVS Components that Require the HFS”

  This chapter describes the installation of all the OpenEdition MVS components that are installed into the hierarchical file system (HFS).
- Chapter 4, “Installation Verification of OpenEdition MVS”

  This chapter describes the steps necessary to check the installation of OpenEdition MVS.
- Chapter 5, “Customizing OpenEdition MVS for General Users”

  This chapter describes the steps required when defining general users to use OpenEdition MVS services.
- Chapter 6, “Maintaining OpenEdition MVS”

  This chapter discusses what is required to maintain the OpenEdition MVS environment.
- Chapter 7, “Other Licensed Programs that Interact with OpenEdition MVS”

  This chapter discusses the setup and configuration of other licensed programs that interact with OpenEdition MVS.
- Chapter 8, “Using OpenEdition MVS”

  This chapter gives a brief overview, with examples, of how you would use some of the OpenEdition MVS services.
- Appendix A, “Refreshing RACF Profiles”

  This appendix shows the process of refreshing the RACF database after you add new RACF profiles.
- Appendix B, “Setting the TZ Environment Variable”

  This appendix explains how to set the TZ environment variable for OpenEdition MVS.
- Appendix C, “Troubleshooting”

This appendix outlines a few of the problems and solutions to those problems that we experienced while writing this book.

- Appendix D, “CA-ACF2 Support for OpenEdition MVS”

  This appendix outlines the steps necessary to customize CA-ACF2 to support OpenEdition MVS.

- Appendix E, “CA-Top Secret Support for OpenEdition MVS”

  This appendix outlines the steps necessary to customize CA-Top Secret to support OpenEdition MVS.

- Appendix F, “Uploading the Source Files from the Diskette”

  This appendix has the instructions on how to unload the sample diskette.

Related Publications

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

- *MVS/ESA OpenEdition MVS User's Guide*, SC23-3013
- *MVS/ESA OpenEdition MVS Command Reference*, SC23-3014
- *MVS/ESA Planning: OpenEdition MVS*, SC23-3015
- *MVS/ESA OpenEdition MVS Messages and Codes*, SC23-3780
- *MVS/ESA Using REXX to Access OpenEdition MVS Services*, SC23-3803
- *MVS/ESA OpenEdition MVS Communications Server Guide*, SC23-3883
- *MVS/ESA OpenEdition MVS Advanced Application Programming Tools*, SC23-3017
- *MVS/ESA Application Development Reference: Assembler Callable Services for OpenEdition MVS*, SC23-3020
- *MVS/ESA Planning: APPC Management*, GC28-1503
- *MVS/ESA Initialization and Tuning Reference*, SC28-1452
- *DFSMSdfp Storage Administration Reference*, SC26-4920
- *DFSMSdss Storage Administration Reference*, SC26-4929
- *TSO/E Procedures Language Reference for MVS/REXX*, SC28-1883-05
- *RMF Support for OpenEdition MVS User's Guide*, GC33-6475
- *RMF Support for OpenEdition MVS Analyzing RMF Reports*, LY33-9175
- *RACF System Programmer's Guide V2R1*, SC23-3725
- *RACF Security Administrator's Guide V2R1*, SC23-3726
- *RACF General User's Guide*, SC23-3728
- *RACF Command Language Reference V2R1*, SC23-3731
- *RACF Diagnosis Guide*, LY27-2635
- *DFSMS/MVS V1R2.0 Using ISMF*, SC26-4911
- *DFSMS/MVS V1R2.0 Planning for Installation*, SC26-4919
- *DFSMS/MVS: Network File System Performance Tuning Guide*, SC26-7019
- *DFSMS/MVS: Network File System User's Guide*, SC26-7028

- *DFSMS/MVS: Network File System Customization and Implementation*, SC26-7029
- *MVS/ESA SML Implementing System-Managed Storage*, SC26-3123
- *MVS/ESA SML Managing Data*, SC26-3124
- *MVS/ESA SML Managing Storage Groups*, SC26-3125
- *MVS/ESA SML Leading a Storage Administration Group*, SC26-3126
- *MVS/ESA SP V5 JES2 Initialization and Tuning Reference*, SC28-1454
- *MVS/ESA SP V5 JES3 Initialization and Tuning Reference*, SC28-1456
- *Device Support Facilities User's Guide and Reference*, GC35-0033
- *Language Environment for MVS & VM: Installation and Customization on MVS*, SC26-4817
- *Language Environment for MVS & VM: Debugging Guide and Run-Time Messages*, SC26-4829
- *C/C++ MVS User's Guide*, SC09-2061
- *C/C++ MVS Library Reference*, SC09-2063
- *C/C++ MVS Library Reference: OpenEdition MVS Sockets*, SC23-3875
- *AD/Cycle C/370 Library Reference*, SC09-1761
- *AD/Cycle C/370 Language Reference*, SC09-1762
- *AD/Cycle C/370 User's Guide*, SC09-1763
- *IBM TCP/IP for MVS: Messages and Codes*, SC31-7132-00
- *IBM TCP/IP for MVS: Customization and Administration Guide*, SC31-7134-00
- *IBM TCP/IP for MVS: Programmer's Reference*, SC31-7135
- *IBM TCP/IP for MVS: User's Guide*, SC31-7136
- *SMP/E R8.1 Reference*, SC28-1107
- *ADSM Version 2 Administrator's Guide*, SH26-4039
- *ADSM Version 2 Administrator's Reference*, SH26-4040
- *ADSM Version 2 MVS Installing the Server and Administrative Client*, SH26-4043
- *ADSM Version 2 Using the UNIX Backup-Archive Clients*, SH26-4052

International Technical Support Organization Publications

- *RACF Support for Open Systems*, GG26-2005
- *DFSMS/MVS Version 1 Release 3.0, Presentation Guide*, GG24-4391
- *Porting Applications to the OpenEdition MVS Platform*, GG24-4473
- *OpenEdition MVS Communication Server Implementation Guide*, SG24-4619

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

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

To get a catalog of ITSO redbooks, VNET users may type:

```
TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
```

A listing of all redbooks, sorted by category, may also be found on MKTTOOLS as ITSOCAT TXT. This package is updated monthly.

How to Order ITSO 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 1-800-445-9269. Most major credit cards are accepted. Outside the USA, customers should contact their local IBM office. Guidance may be obtained by sending a PROFS note to BOOKSHOP at DKIBMVM1 or E-mail to bookshop@dk.ibm.com.

Customers may order hardcopy ITSO books individually or in customized sets, called BOFs, which relate to specific functions of interest. IBM employees and customers may also order ITSO books in online format on CD-ROM collections, which contain redbooks on a variety of products.

ITSO Redbooks on the World Wide Web (WWW)

Internet users may find information about redbooks on the ITSO World Wide Web home page. To access the ITSO Web pages, point your Web browser to the following URL:

```
http://www.redbooks.ibm.com/redbooks
```

IBM employees may access LIST3820s of redbooks as well. The internal Redbooks home page may be found at the following URL: home page:

```
http://w3.itsc.pok.ibm.com/redbooks/redbooks.html
```

Other Publications

- *CA-ACF2 6.1 MVS Administration Guide*
- *CA-Top Secret 5.0 MVS Users Guide*

Acknowledgments

This project was designed and managed by:

Rich Conway
International Technical Support Organization, Poughkeepsie Center

The authors of this document are:

Eckhard Dorniak
IBM Germany

Jesus Guerrero
IBM Venezuela

Paul Lekkas
IBM Germany

Rainer Petri
IBM Germany

Rich Conway
International Technical Support Organization, Poughkeepsie Center

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

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

Alfred Christensen
International Technical Support Organization, Raleigh Center

Rainer Heyt
IBM Germany

Elizabeth Holland
OpenEdition DCE Development

Lori Kikuchi
IBM Software Migration Project Office

Hans Dieter Mertiens
IBM Germany

William Schoen
OpenEdition MVS Development

A special thanks to Dan Wilkinson, Scottie Snyder and the rest of the team at Computer Associates for providing us with material on CA-ACF2 and CA-Top Secret.

Chapter 1. Introduction to OpenEdition MVS

The MVS support for OpenEdition services component enables two open systems interfaces on the MVS operating system: an application programming interface (API) and, optionally, an interactive shell interface. With the API, the programs can run in any environment including batch, in jobs submitted by TSO/E interactive users, and in started tasks, or in any other MVS application task environment. The programs can request:

- Only MVS services
- Only OpenEdition MVS services
- Both MVS and OpenEdition MVS services

The optional shell interface is an execution environment analogous to TSO/E, with a programming language of shell commands analogous to the Restructured eXtended eXecutor (REXX) language. The shell work consists of:

- Programs run interactively by shell users.
- Shell commands and scripts run interactively by shell users.
- Shell commands and scripts run as batch jobs.

1.1 OpenEdition Support

OpenEdition support is provided by the following:

- The OpenEdition System Services component which provides OpenEdition services in answer to requests from programs and the shell.
- The OpenEdition Application Services component contains the code to support full-screen applications, remote login and automount support.
- The OpenEdition Shell and Utilities feature interprets commands from interactive users or from programs, called shell scripts, and requests MVS services in response to those commands.
- The OpenEdition Debugger feature helps an application programmer debug source programs written in C.

Before you can use OpenEdition services, the OpenEdition System Services component (the kernel) must be installed and started. The OpenEdition Shell and Utilities and the OpenEdition Debugger are optional features, however.

1.2 OpenEdition MVS Product Set

The following table, Table 1 on page 2 identifies the products needed in support of a full function OpenEdition MVS environment.

Table 1. Minimum Levels of Products in Support of the OpenEdition MVS Environment

Product Name	Program Number	Base FMID	VRM
Multiple Virtual Storage/Enterprise Systems Architecture (MVS/ESA)	5655-068 or 5655-069	JBB5522	5.2.2
OpenEdition System Services		HOM1130	
OpenEdition Application Services		HOT1130	
OpenEdition Shell and Utilities		HSU1130	
OpenEdition Debugger		HDX1130	
Data Facility System Managed Storage (DFSMS/MVS)	5695-DF1	HDZ11B0	1.2.0
ACF/VTAM	5685-085	HVT3401	3.4.1
TSO/E	5685-025	HTE24D0	2.4.0
ISPF	5665-402	HIF3502	3.5.0
RACF or equivalent	5695-039	HRF2210	2.1.0
C/C++ Language Support Feature of MVS/ESA SP 5.2.2 -OR- Language Environment for MVS and VM	5655-068, 069 or 5688-198	HMWL510	1.5.0
AD/Cycle C/370 Compiler -OR- C/C++ Compiler for MVS/ESA	5688-216 or 5655-121	HSQ4201 or HLB4301	1.2.0 or 3.1.0
Transmission Control Protocol/Internet Protocol (TCP/IP)	5655-HAL	HTCP310	3.1.0
DFSMS/MVS Network File System Server	5695-DF1	HDZ11NE or HDZ11NP	1.2.0
System Modified Program Extended (SMP/E)	5688-949	HMP1800	1.8.0
RMF	5685-029	HRM5520	5.2.2

Chapter 2. Beginning the Installation

This starter kit will help you with the installation and customization of OpenEdition MVS. Sample JCL, procedures, and source code that is referred to in this book is provided on a diskette and available on the internet. See Appendix F, “Uploading the Source Files from the Diskette” on page 225 for information on how to unload the files off the diskette and obtain a copy off the internet.

2.1 Installing the BCP and OpenEdition MVS System Services

You should start off by reading the MVS/ESA SP 5.2.2 program directory and following the steps to install the MVS SP 5.2.2 BCP and OpenEdition MVS System Services components as shown in Table 2. Use System Modification Program Extended (SMP/E) to RECEIVE, APPLY and optionally, ACCEPT the OpenEdition MVS System Services component FMIDs.

Table 2. List of BCP and OpenEdition MVS System Services Components

FMID	Component Name
JBB5522	Base Control Program
JBB55N2	BCP NLS US English
JBB55J2	BCP NLS Japanese
HOM1130	OpenEdition MVS System Services
JOM13N0	OpenEdition MVS System Services NLS US English
JOM13J0	OpenEdition MVS System Services NLS Japanese

2.2 Minimum Driving System Requirements to Install BCP and OpenEdition MVS System Services

This section describes the environment of the driving system required to install MVS/ESA SP 5.2.2. The term driving system, is the system that is IPLed and is used to run the SMP/E jobs against the target system. The term target system, is the system (or set of packs) where new software is to be installed.

The following products in Table 3 are the minimum levels required on the driving system to install the FMIDs listed in Table 2.

Table 3 (Page 1 of 2). Minimum Levels on the Driving System to Install JBB5522 and HOM1130

Product Name	Program Number	Minimum Version and Release
System Modification Program Extended (SMP/E)	5668-949	V1 R8 with PTF UR44005 (which is level 18.24) or SMP/E V1 R8.1 with PTF UR44006 (which is level 18.1.15)
Device Support Facilities (ICKDSF)	5655-257	V1 R15
High Level Assembler	5696-234	V1 R1 with PTFs UN39598, UN40215, UN42100, UN44370, UN69897, and UN71605

Table 3 (Page 2 of 2). Minimum Levels on the Driving System to Install JBB5522 and HOM1130		
Product Name	**Program Number**	**Minimum Version and Release**
Linkage editor or Program Binder	-	Packaged with one of the following: • DFSMS/MVS Version 1 (5695-DF1). • If using the Program Binder, PTFs: UW02204, UY88012, UY88444, UY94714, UY95204, UY94429, UY98043, UW00356, and UW02465 are required. • If using the Program Binder with DFSMS/MVS 1.2, PTFs UW02203, UW00747, and UW02464 are also required. • MVS/DFP Version 3 (5665-XA3). • MVS/XA DFP Version 2 (5665-XA2).

2.3 Minimum Driving System Requirements for the Rest of the OpenEdition MVS Components

The rest of the OpenEdition MVS components:

OpenEdition Application Services

OpenEdition Shell and Utilities

OpenEdition Debugger

install their elements into the OpenEdition MVS hierarchical file system (HFS). All access to the HFS is controlled by the OpenEdition MVS kernel address space. There are a number of customization steps that need to be completed before the OpenEdition MVS kernel can be started and the rest of the OpenEdition MVS components installed, such as the following:

- DFSMS/MVS 1.2 must be installed and a mini DFSMS environment activated to control the hierarchical file system.
- APPC/MVS must be customized and started since OpenEdition MVS uses address spaces provided by APPC/MVS.
- RACF 2.1 or an equivalent security product must be installed and a few profiles added to enable OpenEdition MVS services.

In addition to the products in Table 3 on page 3, the following products are needed on the driving system to install the rest of the OpenEdition MVS components.

Table 4. Minimum Driving System Requirements for the Rest of the OpenEdition MVS Components		
Product Name	**Program Number**	**Minimum Version and Release**
MVS/ESA with OpenEdition MVS System Services	5695-047 or 5695-048	V4.3
DFSMS/MVS	5695-DF1	V1.2
RACF or equivalent	5695-039	V2.1

Note: One way to get to this level of driving system software, is to first install the FMIDs in Table 2 on page 3, DFSMS/MVS 1.2 and RACF V2.1 or equivalent on a target system, then use this target system as the driving system for installing the rest of the OpenEdition MVS components.

2.4 Setup DFSMS for OpenEdition MVS

DFSMS/MVS manages the hierarchical file system (HFS) data sets that contain the file systems. If you do not have DFSMS customized on your system, you will need to customize it and get it running for OpenEdition MVS. The following steps will help you in preparing a mini System Managed Storage (SMS) environment in order to run OpenEdition MVS.

Before you define and activate an SMS configuration, you need to perform the following preparatory steps:

- Allocate control data sets to contain your SMS configuration and to permit the systems in your complex to communicate with each other.
- Modify SYS1.PARMLIB, which contains three members that direct the initialization and activation of SMS.
- Establish access to the ISMF Primary Option Menu for storage administrators.

After the preparatory steps are performed, you will need to define the following before activating SMS on your system:

- Define the base configuration attributes for the new source control data set (SCDS).
- Define a storage class to control the OpenEdition hierarchical file system (HFS).
- Define a storage group to contain the physical storage devices (DASD volumes) where the OpenEdition hierarchical file system (HFS) data sets will reside.
- Translate automatic class selection (ACS) routines that contain the rules for allocation of the OpenEdition hierarchical file system (HFS) data sets.

This section describes how to perform all these tasks with easy to read examples. For additional information regarding the planning and preparation for SMS, see *MVS/ESA SML: Implementing System-Managed Storage*.

2.4.1 Setting Up a Mini DFSMS Environment for OpenEdition MVS

Step 1 Using ICKDSF to prepare the DASD volume(s)

The first step to activating System Managed Storage (SMS) is the preparation of new DASD volume(s) for SMS storage groups. Use the Device Support Facility (ICKDSF) product to initialize DASD volumes to be SMS managed. The INIT command of ICKDSF is used to define a volume serial number and allocation of an indexed VTOC. The STORAGEGROUP (STGR) keyword of the INIT command is used to make a DASD volume available for the allocation of new system-managed data sets. A sample of this JCL is found in member ICKINIT in OMVS.JCL. For

additional information please read *Device Support Facilities User's Guide and Reference Rel16*.

Step 2 Allocating the SMS control data sets

Before you can activate an SMS configuration, you need to allocate SMS control data sets and define their contents. SMS uses three types of control data sets: a source control data set (SCDS), an active control data set (ACDS), and a communications data set (COMMDS). Control data sets are virtual storage access method (VSAM) linear data sets that contain base configuration information, SMS class, aggregate group, optical library, tape library, optical drive, storage group definitions, and ACS routines. You define and alter the contents of control data sets using Interactive Storage Management Facility (ISMF) panels. The necessary JCL to allocate the SMS control data sets is in member SMSALLOC in data set OMVS.JCL.

Step 3 Update SYS1.PARMLIB

Parmlib member IGDSMSxx contains the names of the active control data set (ACDS) and the communication data set (COMMDS) for SMS. Member IEFSSNxx contains the necessary entry for SMS and member IEASYSxx contains pointers to both the IGDSMSxx and IEFSSNxx members. Examples of the SMS relevant parameters are provided in data set OMVS.PARMLIB as IGDSMS00, IEFSSN00, and IEASYS00 respectively. They should only be used as references as to how you can set up your SMS parmlib members. For example, members IEASYS00 and IEFSSN00 contain many more parameters on your system that are not shown in the members of OMVS.PARMLIB. Make sure that the IEASYSxx member which is used to IPL your system points to the appropriate IGDSMSxx and IEFSSNxx members on your system.

Note: The SMS entry in member IEFSSNxx should be coded before the entry for the primary job entry subsystem. For more information see *MVS/ESA Initialization and Tuning Reference*.

The following figures will show examples of the members of OMVS.PARMLIB.

```
******************************** Top of Data ************************
SMS=00,         SMS SELECTION
SSN=00,         SUBSYSTEM INITIALIZATION NAMES
SYSNAME=SC60    SYSTEM NAME
******************************** Bottom of Data *********************
```

Figure 1. OMVS.PARMLIB(IEASYS00)

```
******************************** Top of Data ************************
SMS,IGDSSIIN,'ID=00,PROMPT=DISPLAY'  SMS
JES2,,,PRIMARY,NOSTART               JES2 IS THE PRIMARY SUBSYSTEM NAME
TNF,MVPTSSI                          TCP/IP
VMCF,MVPXSSI,WTSC60                  TCP/IP   /*change to your TCP/IP hostname */
******************************** Bottom of Data ***********************
```

Figure 2. OMVS.PARMLIB(IEFSSN00)

```
******************************** Top of Data ************************
SMS ACDS(SMS.ACDS1.ACDS) COMMDS(SMS.COMMDS1.COMMDS)
******************************** Bottom of Data ***********************
```

Figure 3. OMVS.PARMLIB(IGDSMS00)

Step 4 Activate ISMF panels

If SMS is not defined at an installation it is possible that the Interactive Storage Management Facility (ISMF) panels aren't active either. If this is the case, ISMF has to be installed prior to the activation of SMS because the necessary SMS definitions are done using ISMF panels. One part of this installation is the modification of the TSO logon procedure as shown in Figure 4.

Note: This example assumes that the ISP.V4R1M0.ISPLOAD data set is part of the LNKLST concatenation.

```
//ISPLLIB   DD DSN=SYS1.DFQLLIB,DISP=SHR
//          DD DSN=SYS1.DGTLLIB,DISP=SHR
//ISPPLIB   DD DSN=ISP.V4R1M0.ISPPENU,DISP=SHR
//          DD DSN=SYS1.DFQPLIB,DISP=SHR
//          DD DSN=SYS1.DGTPLIB,DISP=SHR
//ISPMLIB   DD DSN=ISP.V4R1M0.ISPMENU,DISP=SHR
//          DD DSN=SYS1.DFQMLIB,DISP=SHR
//          DD DSN=SYS1.DGTMLIB,DISP=SHR
//ISPSLIB   DD DSN=ISP.V4R1M0.ISPSENU,DISP=SHR
//          DD DSN=SYS1.DGTSLIB,DISP=SHR
//ISPTLIB   DD DSN=ISP.V4R1M0.ISPTLIB,DISP=SHR
//          DD DSN=SYS1.DGTTLIB,DISP=SHR
//SYSPROC   DD DSN=ISP.V4R1M0.ISPCLIB,DISP=SHR
//          DD DSN=SYS1.DGTCLIB,DISP=SHR
```

Figure 4. ISMF Panels Setup

After you have access to the ISMF data sets from your TSO session, you then can modify an existing ISPF panel, for example, ISR@PRIM, to provide entry into panel DGTSMMD1, the ISMF Primary Option menu. The additional entry can look like the following:

- `I,'PGM(DGTFMD01)'`

OR

if you don't want to modify an existing ISPF panel you can start the ISMF panels from TSO READY.

- From TSO READY type in: `ISPSTART PGM(DGTFMD01)`

For further information regarding the setup and use of the ISMF panels, see *DFSMS/MVS V1R2.0 Planning for Installation* and *DFSMS/MVS V1R2.0 Using ISMF*.

Step 5 Set your user ID up as storage administrator

To be able to perform the definitions described on the next pages the user must be setup as storage administrator for ISMF. Invoke the ISMF panel interface and give yourself storage administrator authority as shown in the next figures:

Note: Your initial panel presentation as shown in Figure 5 may be different.

```
  Menu  Utilities  Compilers  Options  Status  Help
-------------------------------------------------
ISR@PRIM                   ISPF Primary Option Menu
Option ===> I

0  Settings      Terminal and user parameters
1  View          Display source data or listings
2  Edit          Create or change source data
3  Utilities     Perform utility functions
4  Foreground    Interactive language processing
5  Batch         Submit job for language processing
6  Command       Enter TSO or Workstation commands
7  Dialog Test   Perform dialog testing
8  LM Facility   Library administrator functions
9  IBM Products  IBM program development products
10 SCLM          SW Configuration Library Manager
I  ISMF          Interactive Storage Management Facility
```

Figure 5. Setup the User As ISMF Storage Administrator (1 of 4)

```
                          ISMF PRIMARY OPTION MENU
ENTER SELECTION OR COMMAND ===> 0  TO CHANGE THE ISMF USER PROFILE

SELECT ONE OF THE FOLLOWING:

 0  ISMF PROFILE              - CHANGE ISMF USER PROFILE
 1  DATA SET                  - PERFORM FUNCTIONS AGAINST DATA SETS
 2  VOLUME                    - PERFORM FUNCTIONS AGAINST VOLUMES
 3  MANAGEMENT CLASS          - DISPLAY BACKUP AND MIGRATION CRITERIA
 4  DATA CLASS                - DISPLAY DATA SET ALLOCATION PARAMETERS
 5  STORAGE CLASS             - DISPLAY PERFORMANCE AND AVAILABILITY CRITERIA
 L  LIST                      - PERFORM FUNCTIONS AGAINST SAVED ISMF LISTS
 R  REMOVABLE MEDIA MANAGER   - PERFORM FUNCTIONS AGAINST REMOVABLE MEDIA
 X  EXIT                      - TERMINATE ISMF


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

Figure 6. Setup the User As ISMF Storage Administrator (2 of 4)

```
                          ISMF PROFILE OPTION MENU
ENTER SELECTION OR COMMAND ===> 0   FOR USER MODE SELECTION

SELECT ONE OF THE FOLLOWING OPTIONS AND PRESS ENTER:

  0  USER MODE SELECTION
  1  LOGGING AND ABEND CONTROL
  2  ISMF JOB STATEMENT
  3  DFSMSDSS EXECUTE STATEMENT
  4  ICKDSF EXECUTE STATEMENT
  5  DATA SET PRINT EXECUTE STATEMENT
  6  IDCAMS EXECUTE STATEMENT
  X  EXIT

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

Figure 7. Setup the User As ISMF Storage Administrator (3 of 4)

```
                          USER MODE ENTRY PANEL
COMMAND ===>

SPECIFY THE FOLLOWING:

  USER MODE  ===> 2    (TO SPECIFY YOUR CHOICE OF SESSION, TYPE IN A:

                         1 FOR AN END USER (EU)
                         2 FOR A STORAGE ADMINISTRATOR (SA)

                       IN THE USER MODE FIELD AND PRESS ENTER TO
                       VERIFY YOUR SELECTION.)

                       NOTE:  YOU MUST EXIT AND REENTER ISMF AFTER
                              CHANGING THE USER MODE FIELD IN ORDER
                              TO VIEW AND USE YOUR SELECTED SESSION.

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

Figure 8. Setup the User As ISMF Storage Administrator (4 of 4)

Note: You must exit ISMF now after changing your user mode to storage administrator and reenter ISMF as shown in the next step and continue with your customization of SMS.

Step 6 Select ISMF

In the figure below we will select ISMF now as storage administrator from the TSO/E main menu.

```
  Menu  Utilities  Compilers  Options  Status  Help
-------------------------------------------------
ISR@PRIM                  ISPF Primary Option Menu
Option ===> I

0  Settings      Terminal and user parameters
1  View          Display source data or listings
2  Edit          Create or change source data
3  Utilities     Perform utility functions
4  Foreground    Interactive language processing
5  Batch         Submit job for language processing
6  Command       Enter TSO or Workstation commands
7  Dialog Test   Perform dialog testing
8  LM Facility   Library administrator functions
9  IBM Products  IBM program development products
10 SCLM          SW Configuration Library Manager
I  ISMF          Interactive Storage Management Facility
```

Figure 9. Select ISMF

Step 7 Define a base configuration

The first thing that you need to define in an SCDS is the base configuration information. The base configuration contains installation defaults and identifies the systems to which the SMS configuration applies.

The next few screens will show how a base configuration is defined using the ISMF panels. In our example we only specified the minimum definitions needed to run OpenEdition MVS.

```
                          ISMF PRIMARY OPTION MENU
ENTER SELECTION OR COMMAND ===> 8

SELECT ONE OF THE FOLLOWING OPTIONS AND PRESS ENTER:

 0  ISMF PROFILE              - Specify ISMF User Profile
 1  DATA SET                  - Perform Functions Against Data Sets
 2  VOLUME                    - Perform Functions Against Volumes
 3  MANAGEMENT CLASS          - Specify Data Set Backup and Migration Criteria
 4  DATA CLASS                - Specify Data Set Allocation Parameters
 5  STORAGE CLASS             - Specify Data Set Performance and Availability
 6  STORAGE GROUP             - Specify Volume Names and Free Space Thresholds
 7  AUTOMATIC CLASS SELECTION - Specify ACS Routines and Test Criteria
 8  CONTROL DATA SET          - Specify System Names and Default Criteria
 9  AGGREGATE GROUP           - Specify Data Set Recovery Parameters
10  LIBRARY MANAGEMENT        - Specify Library and Drive Configurations
 C  DATA COLLECTION           - Process Data Collection Function
 L  LIST                      - Perform Functions Against Saved ISMF Lists
 R  REMOVABLE MEDIA MANAGER   - Perform Functions Against Removable Media
 X  EXIT                      - Terminate ISMF
```

Figure 10. Define Base Configuration (1 of 6)

```
                          CDS APPLICATION SELECTION
 COMMAND ===>

 TO PERFORM CONTROL DATA SET OPERATIONS, SPECIFY:

   CDS NAME ===> 'SMS.SCDS1.SCDS'
                         (1 to 44 character data set name or 'ACTIVE')

 SELECT ONE OF THE FOLLOWING OPTIONS ===> 2

  1  DISPLAY  - Display the Base Configuration
  2  DEFINE   - Define the Base Configuration
  3  ALTER    - Alter the Base Configuration
  4  VALIDATE - Validate the SCDS
  5  ACTIVATE - Activate the CDS


USE ENTER TO PERFORM SELECTION;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

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

Figure 11. Define Base Configuration (2 of 6)

The Help screen defines the input for the next few screens.

```
HELP------------------------------  DEFINE  --------------------------
COMMAND ===>

   Use the DEFINE option to define the base configuration attributes for a
   new Source Control Data Set (SCDS), as follows:

   DEFAULT MANAGEMENT CLASS  The default management class name.

   DEFAULT UNIT              The device name that should exist on the system
                             where the data set is to be allocated.
   DEFAULT DEVICE GEOMETRY   The default values that are used when converting
                             all requests for space into KB or MB.
       BYTES/TRACK           The default bytes per track that are used.
       TRACKS/CYLINDER       The default tracks per cylinder that are used.

   SYSTEM                    Indicates system names defined to the complex.
   SYSGRP                    Indicates system group names that consist in a
                             system or collection of systems that are defined
                             to the complex.
   Note: The system and system groups combined will not exceed 8 names.



   Use ENTER to continue, END to exit Help.
```

Figure 12. Define Base Configuration (3 of 6)

```
                              SCDS BASE DEFINE                    Page 1 of 2
COMMAND ===>

SCDS NAME:    SMS.SCDS1.SCDS
SCDS STATUS:  INVALID

TO ALTER SCDS BASE, SPECIFY:

 DESCRIPTION ===>
             ===>

 DEFAULT MANAGEMENT CLASS ===>           (1 to 8 characters)
 DEFAULT UNIT             ===>           (esoteric or generic device name)
 DEFAULT DEVICE GEOMETRY
   BYTES/TRACK            ===> 47476     (1-999999)
   TRACKS/CYLINDER        ===> 15        (1-999999)


USE ENTER TO PERFORM VERIFICATION; USE DOWN COMMAND TO VIEW NEXT PANEL;
USE HELP COMMAND FOR HELP; USE END COMMAND TO SAVE AND EXIT; CANCEL TO EXIT.

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

Figure 13. Define Base Configuration (4 of 6)

The system name specified on this panel is the same as the name specified on the SYSNAME parameter in the IEASYSxx member of SYS1.PARMLIB. The SYSNAME parameter of IEASYSxx specifies the name of the system being initialized. This system name is also used to define the system to DFSMS/MVS. See *MVS/ESA Initialization and Tuning Reference* for more information on the SYSNAME parameter of IEASYSxx.

```
                              SCDS BASE DEFINE                    Page 2 of 2
COMMAND ===>

SCDS NAME:    SMS.SCDS1.SCDS
SCDS STATUS:  INVALID

SPECIFY ONE OF THE FOLLOWING OPTIONS ===> 1     (1 Add, 2 Delete, 3 Rename)

  SPECIFY SYSTEM NAME       ===> SC60         or SYS GROUP NAME ===>

  NEW SYSTEM/SYS GROUP NAME ===>          (For option 3, Rename)

SYSTEM:



SYSGRP:



USE ENTER TO PERFORM OPTION; USE UP COMMAND TO VIEW PREVIOUS PANEL;
USE HELP COMMAND FOR HELP; USE END COMMAND TO SAVE AND EXIT; CANCEL TO EXIT.

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

Figure 14. Define Base Configuration (5 of 6)

After defining base configuration parameters, you can verify their completeness and correctness by pressing Enter. If your SCDS base configuration contains any errors, the cursor moves to the error and an error message appears in the short message area. Correct any errors and press Enter to verify the new contents of the SCDS base configuration.

After correcting all errors, use the End command to save the SCDS base configuration and to leave the SCDS Base Define/Alter Panel. A message indicates that SMS has saved your SCDS base configuration.

```
                          CDS APPLICATION SELECTION         CDS BASE SAVED
 COMMAND ===>

 TO PERFORM CONTROL DATA SET OPERATIONS, SPECIFY:

   CDS NAME ===> 'SMS.SCDS1.SCDS'
                           (1 to 44 character data set name or 'ACTIVE')

 SELECT ONE OF THE FOLLOWING OPTIONS ===>

  1  DISPLAY  - Display the Base Configuration
  2  DEFINE   - Define the Base Configuration
  3  ALTER    - Alter the Base Configuration
  4  VALIDATE - Validate the SCDS
  5  ACTIVATE - Activate the CDS


USE ENTER TO PERFORM SELECTION;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

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

Figure 15. Define Base Configuration (6 of 6)

Step 8 Defining a storage class

An SMS configuration can contain the following five types of constructs: storage group, management class, storage class, data class, and aggregat group. A configuration can contain multiple constructs of each type. Data sets managed by SMS are called system-managed. Each system-managed data set or object must reside in a storage group and have a storage class. The system-managed data set or object might also have a management class and a data class.

Because our goal is the definition of an SMS configuration which allows us to run OpenEdition MVS we define only a storage class and a storage group. This is the minimum requirement to run SMS. If you want to extend the SMS definitions to fully exploit SMS, which is highly recommended but beyond the scope of this book, please read the available manuals from the Storage Management Library:

- *MVS/ESA SML Implementing System-Managed Storage*
- *MVS/ESA SML Managing Data*

- *MVS/ESA SML Managing Storage Groups*
- *MVS/ESA SML Leading a Storage Administration Group*

A storage class is a list of performance objectives and availability requirements. Each storage class represents a list of services that are available to data sets and objects having similar access requirements. A storage class does not represent any physical storage, but rather, provides the criteria that SMS uses in determining an appropriate location to place a data set or object.

Every data set that SMS manages has a storage class. When a storage class is assigned to a data set, SMS places the storage class name in both the BCS and the VVDS catalog entries of the data set.

Our next step is the definition of a storage class. This is shown on the next four figures.

```
                          ISMF PRIMARY OPTION MENU
ENTER SELECTION OR COMMAND ===> 5

SELECT ONE OF THE FOLLOWING OPTIONS AND PRESS ENTER:

 0  ISMF PROFILE              - SPECIFY ISMF USER PROFILE
 1  DATA SET                  - PERFORM FUNCTIONS AGAINST DATA SETS
 2  VOLUME                    - PERFORM FUNCTIONS AGAINST VOLUMES
 3  MANAGEMENT CLASS          - SPECIFY DATA SET BACKUP AND MIGRATION CRITERIA
 4  DATA CLASS                - SPECIFY DATA SET ALLOCATION PARAMETERS
 5  STORAGE CLASS             - SPECIFY DATA SET PERFORMANCE AND AVAILABILITY
 6  STORAGE GROUP             - SPECIFY VOLUME NAMES AND FREE SPACE THRESHOLDS
 7  AUTOMATIC CLASS SELECTION - SPECIFY ACS ROUTINES AND TEST CRITERIA
 8  CONTROL DATA SET          - SPECIFY SYSTEM NAMES AND DEFAULT CRITERIA
 9  AGGREGATE GROUP           - SPECIFY DATA SET RECOVERY PARAMETERS
10  LIBRARY MANAGEMENT        - SPECIFY LIBRARY AND DRIVE CONFIGURATION
 C  DATA COLLECTION           - PROCESS DATA COLLECTION FUNCTION
 L  LIST                      - PERFORM FUNCTIONS AGAINST SAVED ISMF LISTS
 R  REMOVABLE MEDIA MANAGER   - PERFORM FUNCTIONS AGAINST REMOVABLE MEDIA
 X  EXIT                      - TERMINATE ISMF

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

Figure 16. Select Storage Class (1 of 4)

The storage class name of OPENMVS is used in all our examples, and is the name we use in our sample ACS routines. You can come up with your own name if you like, but make sure you also change the ACS routines. The ACS routines we used are on the diskette in data set, OMVS.JCL as members OMVSSC and OMVSSG.

```
                    STORAGE CLASS APPLICATION SELECTION
COMMAND ===>

TO PERFORM STORAGE CLASS OPERATIONS, SPECIFY:

  CDS NAME      ===> 'SMS.SCDS1.SCDS'
                        (1 TO 44 CHARACTER DATA SET NAME OR 'ACTIVE')

  STORAGE CLASS NAME ===> OPENMVS  (FOR STORAGE CLASS LIST, FULLY OR
                                    PARTIALLY SPECIFIED OR * FOR ALL)

SELECT ONE OF THE FOLLOWING OPTIONS  ===> 3

 1   LIST    - GENERATE A LIST OF STORAGE CLASSES
 2   DISPLAY - DISPLAY A STORAGE CLASS
 3   DEFINE  - DEFINE A STORAGE CLASS
 4   ALTER   - ALTER A STORAGE CLASS

IF OPTION 1 CHOSEN ABOVE,
   RESPECIFY VIEW CRITERIA           ===> N  (Y OR N)
   RESPECIFY SORT CRITERIA           ===> N  (Y OR N)
 F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=        F6=    F7=UP
 F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=CURSOR
```

Figure 17. Define a Storage Class (2 of 4)

Press Enter to verify, then End to save the new OPENMVS storage class.

```
                              STORAGE CLASS DEFINE
COMMAND ===>

SCDS NAME: 'SMS.SCDS1.SCDS'
STORAGE CLASS NAME: OPENMVS

TO DEFINE STORAGE CLASS, SPECIFY:
  DESCRIPTION ===>
              ===>

  PERFORMANCE OBJECTIVES
   DIRECT MILLISECOND RESPONSE     ===>             (1 TO 999 OR BLANK)
   DIRECT BIAS                     ===>             (R, W OR BLANK)
   SEQUENTIAL MILLISECOND RESPONSE ===>             (1 TO 999 OR BLANK)
   SEQUENTIAL BIAS                 ===>             (R, W OR BLANK)
   INITIAL ACCESS RESPONSE SECONDS ===>             (0 TO 9999 OR BLANK)
   SUSTAINED DATA RATE (MB/SEC)    ===>             (0 TO 999 OR BLANK)
  AVAILABILITY                     ===> N           (C, P, S or N)
  ACCESSIBILITY                    ===> N           (C, P, S or N)
  GUARANTEED SPACE                 ===> N           (Y or N)
  GUARANTEED SYNCHRONOUS WRITE     ===> N           (Y or N)
  F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=        F6=        F7=UP
  F8=DOWN    F9=SWAP   F10=LEFT   F11=RIGHT  F12=CURSOR
```

Figure 18. Define a Storage Class (3 of 4)

```
                    STORAGE CLASS APPLICATION SELECTION      OPENMVS SAVED
COMMAND ===>

TO PERFORM STORAGE CLASS OPERATIONS, SPECIFY:

  CDS NAME      ===> 'SMS.SCDS1.SCDS'
                        (1 TO 44 CHARACTER DATA SET NAME OR 'ACTIVE')

  STORAGE CLASS NAME ===> OPENMVS  (FOR STORAGE CLASS LIST, FULLY OR
                                    PARTIALLY SPECIFIED OR * FOR ALL)

SELECT ONE OF THE FOLLOWING OPTIONS  ===> 1

 1   LIST    - GENERATE A LIST OF STORAGE CLASSES
 2   DISPLAY - DISPLAY A STORAGE CLASS
 3   DEFINE  - DEFINE A STORAGE CLASS
 4   ALTER   - ALTER A STORAGE CLASS

IF OPTION 1 CHOSEN ABOVE,
   RESPECIFY VIEW CRITERIA           ===> N  (Y OR N)
   RESPECIFY SORT CRITERIA           ===> N  (Y OR N)
 F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=        F6=    F7=UP
 F8=DOWN    F9=SWAP    F10=LEFT   F11=RIGHT  F12=CURSOR
```

Figure 19. Define a Storage Class (4 of 4)

Step 9 Defining a storage group

Storage groups represent the physical storage managed by SMS. Each storage group is a collection of physical devices, virtual input/output (VIO) or dummy volumes defined by the storage administrator. A storage group, used with storage classes, separates the logical requirements of accessing data from the physical requirements to store data. A pool storage group is a group of one or more volumes that are accessible from all systems connected to those volumes. Storage groups contain attributes of the devices on which the data sets or objects reside.

The storage group name, OPENMVS, is used in our examples and is the name we use in our sample ACS routines. You can come up with your own name if you like, but make sure you also change the sample ACS routines to reflect this new name.

```
                          ISMF PRIMARY OPTION MENU
ENTER SELECTION OR COMMAND ===> 6

SELECT ONE OF THE FOLLOWING OPTIONS AND PRESS ENTER:

 0  ISMF PROFILE              - SPECIFY ISMF USER PROFILE
 1  DATA SET                  - PERFORM FUNCTIONS AGAINST DATA SETS
 2  VOLUME                    - PERFORM FUNCTIONS AGAINST VOLUMES
 3  MANAGEMENT CLASS          - SPECIFY DATA SET BACKUP AND MIGRATION CRITERIA
 4  DATA CLASS                - SPECIFY DATA SET ALLOCATION PARAMETERS
 5  STORAGE CLASS             - SPECIFY DATA SET PERFORMANCE AND AVAILABILITY
 6  STORAGE GROUP             - SPECIFY VOLUME NAMES AND FREE SPACE THRESHOLDS
 7  AUTOMATIC CLASS SELECTION - SPECIFY ACS ROUTINES AND TEST CRITERIA
 8  CONTROL DATA SET          - SPECIFY SYSTEM NAMES AND DEFAULT CRITERIA
 9  AGGREGATE GROUP           - SPECIFY DATA SET RECOVERY PARAMETERS
10  LIBRARY MANAGEMENT        - SPECIFY LIBRARY AND DRIVE CONFIGURATION
 C  DATA COLLECTION           - PROCESS DATA COLLECTION FUNCTION
 L  LIST                      - PERFORM FUNCTIONS AGAINST SAVED ISMF LISTS
 R  REMOVABLE MEDIA MANAGER   - PERFORM FUNCTIONS AGAINST REMOVABLE MEDIA
 X  EXIT                      - TERMINATE ISMF

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

Figure 20. Define a Storage Group (1 of 4)

```
                    STORAGE GROUP APPLICATION SELECTION
COMMAND ===>

TO PERFORM STORAGE GROUP OPERATIONS, SPECIFY:

  CDS NAME           ===> 'SMS.SCDS1.SCDS'
                          (1 to 44 character data set name or 'ACTIVE')
  STORAGE GROUP NAME ===> OPENMVS           (for Storage Group List, fully or
                                             partially specified or * for all)
  STORAGE GROUP TYPE ===> POOL              (VIO, POOL, DUMMY, OBJECT,
                                             OBJECT BACKUP, or TAPE)
SELECT ONE OF THE FOLLOWING OPTIONS ===> 2
   1  LIST    - Generate a list of Storage Groups
   2  DEFINE  - Define a Storage Group
   3  ALTER   - Alter a Storage Group
   4  VOLUME  - Display, Define, Alter or Delete Volume Information

 IF OPTION 1 CHOSEN ABOVE,
   RESPECIFY VIEW CRITERIA          ===> N   (Y or N)
   RESPECIFY SORT CRITERIA          ===> N   (Y or N)
  F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=        F6=        F7=U
  F8=DOWN    F9=SWAP    F10=LEFT   F11=RIGHT  F12=CURSOR
```

Figure 21. Define a Storage Group (2 of 4)

Fill in the required information and press F3 or select End to save the OPENMVS storage group.

```
                           POOL STORAGE GROUP DEFINE
 COMMAND ===>

 SCDS NAME:          SMS.SCDS1.SCDS
 STORAGE GROUP NAME: OPENMVS

 TO DEFINE STORAGE GROUP, SPECIFY:
  DESCRIPTION  ===> VOLUME POOL FOR OPENMVS STORAGE GROUP
               ===>
  AUTO MIGRATE ===> Y     (Y, N, I or P)  MIGRATE SYS/SYS GROUP NAME ===>
  AUTO BACKUP  ===> Y     (Y or N)        BACKUP SYS/SYS GROUP NAME  ===>
  AUTO DUMP    ===> N     (Y or N)        DUMP SYS/SYS GROUP NAME    ===>

  DUMP CLASS   ===>                    (1 to 8 characters)
  DUMP CLASS   ===>                    DUMP CLASS     ===>
  DUMP CLASS   ===>                    DUMP CLASS     ===>

  ALLOCATION/MIGRATION THRESHOLD: HIGH ===> 85  (1-99)    LOW ===> 50  (0-99)
  GUARANTEED BACKUP FREQUENCY          ===> 1          (1 to 9999 or NOLIMIT)

  DEFINE SMS STORAGE GROUP STATUS       ===> N   (Y or N)
  F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=        F6=        F7=U
  F8=DOWN    F9=SWAP    F10=LEFT   F11=RIGHT  F12=CURSOR
```

Figure 22. Define a Storage Group (3 of 4)

```
                    STORAGE GROUP APPLICATION SELECTION       OPENMVS SAVED
COMMAND ===>

TO PERFORM STORAGE GROUP OPERATIONS, SPECIFY:

  CDS NAME           ===> 'SMS.SCDS1.SCDS'
                          (1 to 44 character data set name or 'ACTIVE')
  STORAGE GROUP NAME ===> OPENMVS            (for Storage Group List, fully or
                                              partially specified or * for all)
  STORAGE GROUP TYPE ===> POOL               (VIO, POOL, DUMMY, OBJECT,
                                              OBJECT BACKUP, or TAPE)
SELECT ONE OF THE FOLLOWING OPTIONS ===> 2
   1  LIST    - Generate a list of Storage Groups
   2  DEFINE  - Define a Storage Group
   3  ALTER   - Alter a Storage Group
   4  VOLUME  - Display, Define, Alter or Delete Volume Information

 IF OPTION 1 CHOSEN ABOVE,
   RESPECIFY VIEW CRITERIA           ===> N   (Y or N)
   RESPECIFY SORT CRITERIA           ===> N   (Y or N)
  F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=        F6=        F7=U
  F8=DOWN    F9=SWAP    F10=LEFT   F11=RIGHT  F12=CURSOR
```

Figure 23. Define a Storage Group (4 of 4)

Step 10 Defining volumes in a storage group

After a storage group has been set up, volumes have to be defined for this storage group. This is shown on the next four screens.

```
                    STORAGE GROUP APPLICATION SELECTION
COMMAND ===>

TO PERFORM STORAGE GROUP OPERATIONS, SPECIFY:

  CDS NAME           ===> 'SMS.SCDS1.SCDS'
                          (1 to 44 character data set name or 'ACTIVE')
  STORAGE GROUP NAME ===> OPENMVS            (for Storage Group List, fully or
                                              partially specified or * for all)
  STORAGE GROUP TYPE ===> POOL               (VIO, POOL, DUMMY, OBJECT,
                                              OBJECT BACKUP, or TAPE)
SELECT ONE OF THE FOLLOWING OPTIONS ===> 4
   1  LIST    - Generate a list of Storage Groups
   2  DEFINE  - Define a Storage Group
   3  ALTER   - Alter a Storage Group
   4  VOLUME  - Display, Define, Alter or Delete Volume Information

 IF OPTION 1 CHOSEN ABOVE,
   RESPECIFY VIEW CRITERIA           ===> N:   (Y or N)
   RESPECIFY SORT CRITERIA           ===> N:   (Y or N)
  F1=HELP    F2=SPLIT   F3=END     F4=RETURN  F5=        F6=        F7=U
  F8=DOWN    F9=SWAP    F10=LEFT   F11=RIGHT  F12=CURSOR
```

Figure 24. Define a Volume for a Storage Group (1 of 4)

Enter the volser of your DFSMS volume below.

```
                        STORAGE GROUP VOLUME SELECTION
COMMAND ===>

CDS NAME:           SMS.SCDS1.SCDS
STORAGE GROUP NAME: OPENMVS
STORAGE GROUP TYPE: POOL

SELECT ONE OF THE FOLLOWING OPTIONS   ===> 2

  1  DISPLAY      - Display SMS Volume Statuses (Pool only)
  2  DEFINE       - Add Volumes to Volume Serial Number List
  3  ALTER        - Alter Volume Statuses   (Pool only)
  4  DELETE       - Delete Volumes from Volume Serial Number List

SPECIFY A SINGLE VOLUME (in PREFIX), OR RANGE OF VOLUMES:
       PREFIX   FROM     TO    SUFFIX  HEX
       ______  ______  ______  _____    _
 ===>  OP2HFS                            ('X' in HEX field allow
 ===>                                     FROM - TO range to include
 ===>                                     hex values A through F.)
 ===>

USE ENTER TO PERFORM SELECTION;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.



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

Figure 25. Define a Volume for a Storage Group (2 of 4)

Press Enter, then press F3 or select End to save and exit.

```
                          SMS VOLUME STATUS DEFINE
COMMAND ===>

SCDS NAME:              SMS.SCDS1.SCDS
STORAGE GROUP NAME:     OPENMVS
VOLUME SERIAL NUMBERS:  OP2HFS


TO ALTER SMS VOLUME STATUS, SPECIFY:

   SYSTEM/SYS GROUP     SMS VOL
   NAME                 STATUS
   ----------------     -------
   SC60            ===> ENABLE         ( Possible SMS VOL STATUS values
                   ===>                  for each system or system group:
                   ===>                  NOTCON, ENABLE, DISALL, DISNEW,
                   ===>                  QUIALL, or QUINEW )
                   ===>
                   ===>                  * SYS GROUP = sysplex minus systems
                   ===>                    in the sysplex explicitly defined
                   ===>                    in the SCDS

USE ENTER TO PERFORM VERIFICATION;
USE HELP COMMAND FOR HELP; USE END COMMAND TO SAVE AND EXIT; CANCEL TO EXIT.



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

Figure 26. Define a Volume for a Storage Group (3 of 4)

```
                        STORAGE GROUP VOLUME SELECTION    ALL VOLUMES DEFINED
COMMAND ===>

CDS NAME:           SMS.SCDS1.SCDS
STORAGE GROUP NAME: OPENMVS
STORAGE GROUP TYPE: POOL

SELECT ONE OF THE FOLLOWING OPTIONS   ===> 2

  1  DISPLAY      - Display SMS Volume Statuses (Pool only)
  2  DEFINE       - Add Volumes to Volume Serial Number List
  3  ALTER        - Alter Volume Statuses   (Pool only)
  4  DELETE       - Delete Volumes from Volume Serial Number List

SPECIFY A SINGLE VOLUME (in PREFIX), OR RANGE OF VOLUMES:
       PREFIX   FROM     TO    SUFFIX  HEX
       ______  ______  ______  _____    _
 ===>  OP2HFS                           ('X' in HEX field allow
 ===>                                    FROM - TO range to include
 ===>                                    hex values A through F.)
 ===>

USE ENTER TO PERFORM SELECTION;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.



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

Figure 27. Define a Volume for a Storage Group (4 of 4)

Step 11 Translate ACS routines

After storage class, and storage group have been defined and volumes have been defined to the storage group, the ACS routines have to be translated into an executable form. The translation process checks your source code for syntactic and semantic errors, generates an object form if no errors exist, and places the object form into the SCDS you specified on the translate panel. If the ACS routine that you are translating already exists in the SCDS, the new object form replaces the existing object form.

The sample ACS routines can be found in data set OMVS.JCL, member names OMVSSC and OMVSSG and are shown in Figure 28 on page 23 and Figure 29 on page 23.

```
******************************** Top of Data ******************
PROC STORCLAS
/*****************************************************************
/* STORAGECLASS DEFINITION FOR OMVS DATA SETS ONLY
/*****************************************************************
FILTLIST OMVS INCLUDE(OMVS*)
 SELECT
    WHEN (&HLQ= 'OMVS')
        SET &STORCLAS = 'OPENMVS'
    OTHERWISE
        SET &STORCLAS = ''
    END
END
************************** Bottom of Data **********************
```

Figure 28. Sample Storage Class ACS Routine

```
******************** Top of Data ******************
PROC STORGRP
SELECT
  WHEN (&STORCLAS = 'OPENMVS')
      SET &STORGRP = 'OPENMVS'
  OTHERWISE WRITE 'OPENMVS IS THE ONLY DEFINED STORCLAS'
  END
END
***************** Bottom of Data ******************
```

Figure 29. Sample Storage Group ACS Routine

ACS routines have to be translated before they can be used. How this translation is done is shown on the next screens. Go back to the ISMF PRIMARY OPTION MENU and choose option 7 to go to the automatic class selection panel as shown in Figure 30 on page 24.

```
                          ISMF PRIMARY OPTION MENU
ENTER SELECTION OR COMMAND ===> 7

SELECT ONE OF THE FOLLOWING OPTIONS AND PRESS ENTER:

 0  ISMF PROFILE              - SPECIFY ISMF USER PROFILE
 1  DATA SET                  - PERFORM FUNCTIONS AGAINST DATA SETS
 2  VOLUME                    - PERFORM FUNCTIONS AGAINST VOLUMES
 3  MANAGEMENT CLASS          - SPECIFY DATA SET BACKUP AND MIGRATION CRITERIA
 4  DATA CLASS                - SPECIFY DATA SET ALLOCATION PARAMETERS
 5  STORAGE CLASS             - SPECIFY DATA SET PERFORMANCE AND AVAILABILITY
 6  STORAGE GROUP             - SPECIFY VOLUME NAMES AND FREE SPACE THRESHOLDS
 7  AUTOMATIC CLASS SELECTION - SPECIFY ACS ROUTINES AND TEST CRITERIA
 8  CONTROL DATA SET          - SPECIFY SYSTEM NAMES AND DEFAULT CRITERIA
 9  AGGREGATE GROUP           - SPECIFY DATA SET RECOVERY PARAMETERS
10  LIBRARY MANAGEMENT        - SPECIFY LIBRARY AND DRIVE CONFIGURATION
 C  DATA COLLECTION           - PROCESS DATA COLLECTION FUNCTION
 L  LIST                      - PERFORM FUNCTIONS AGAINST SAVED ISMF LISTS
 R  REMOVABLE MEDIA MANAGER   - PERFORM FUNCTIONS AGAINST REMOVABLE MEDIA
 X  EXIT                      - TERMINATE ISMF

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

Figure 30. Setup Translation of ACS Routines (1 of 5)

To translate an ACS routine, select option 2, TRANSLATE, from the Automatic Class Selection Application Selection panel. ISMF displays the Translate ACS Routines panel shown in Figure 31.

```
                          ACS APPLICATION SELECTION
 COMMAND ===>

 SELECT ONE OF THE FOLLOWING OPTIONS ===> 2

   1  EDIT            - Edit ACS Routine source code
   2  TRANSLATE       - Translate ACS Routines to ACS Object Form
   3  VALIDATE        - Validate ACS Routines Against Storage Constructs
   4  TEST            - Define/Alter Test Cases and Test ACS Routines
   5  DISPLAY         - Display ACS Object Information
   6  DELETE          - Delete an ACS Object from a Source Control Data Set




 IF OPTION 5 CHOSEN ABOVE, SPECIFY:

   CDS NAME  ===> 'SMS.SCDS1.SCDS'
                              (1 to 44 character data set name or 'ACTIVE')



 USE ENTER TO PERFORM SELECTION;
 USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

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

Figure 31. Setup Translation of ACS Routines (2 of 5)

```
                          TRANSLATE ACS ROUTINES
COMMAND ===>

TO PERFORM ACS TRANSLATION, SPECIFY:

   SCDS NAME           ===> 'SMS.SCDS1.SCDS'
                                (1 to 44 character data set name)

   ACS SOURCE DATA SET ===> 'OMVS.JCL'
                                (1 to 44 character data set name)

   ACS SOURCE MEMBER   ===> OMVSSC   (1 to 8 characters)


   LISTING DATA SET    ===> LISTO
                                (1 to 44 character data set name)



USE ENTER TO PERFORM ACS TRANSLATION;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

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

Figure 32. Setup Translation of ACS Routines (3 of 5)

The following figure will show the output from a successful translation of an ACS routine; in this example, it is the ACS routine for the storage class.

```
ACS TRANSLATOR ***** TIME 18:46:11 DATE 08/13/1995 PAGE 0001 *****


SCDS NAME:               SMS.SCDS1.SCDS
ACS SOURCE DATA SET:     OMVS.JCL
ACS SOURCE MEMBER:       OMVSSC



0001      PROC STORCLAS
0002      /*************************************************
0003      /* STORAGECLASS DEFINITION FOR OMVS DATA SETS ONLY
0004      /*************************************************
0005      FILTLIST OMVS INCLUDE(OMVS*)
0006       SELECT
0007          WHEN (&HLQ= 'OMVS')
0008              SET &STORCLAS = 'OPENMVS'
0009          OTHERWISE
0010              SET &STORCLAS = ''
0011          END
0012       END



TRANSLATION RETURN CODE:  0000
```

Figure 33. Setup Translation of ACS Routines (4 of 5)

On the next screen a decision has to be made whether or not the output from the translation should be printed.

```
                          OUTPUT LISTING DISPOSITION   TRANSLATION SUCCESSFUL
COMMAND ===>

LISTING DATA SET: RPETRI.LISTO

SPECIFY OUTPUT LISTING DISPOSITION:


  PRINT OUTPUT LISTING            ===> N      (Y or N)


  DELETE OUTPUT LISTING           ===> N      (Y or N)









USE ENTER TO PERFORM SELECTION;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.


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

Figure 34. Setup Translation of ACS Routines (5 of 5)

Translate Storage Group

Go back to the beginning of this step and translate the OMVSSG ACS routine (storage group routine) the same way as the OMVSSC ACS routine (storage class routine) was done.

Step 12 Validate the configuration

After the storage class and storage group have been defined and the ACS routines have been translated the configuration can be validated. This is shown on the next screens.

```
                          ISMF PRIMARY OPTION MENU
ENTER SELECTION OR COMMAND ===> 8

SELECT ONE OF THE FOLLOWING OPTIONS AND PRESS ENTER:

 0  ISMF PROFILE              - Specify ISMF User Profile
 1  DATA SET                  - Perform Functions Against Data Sets
 2  VOLUME                    - Perform Functions Against Volumes
 3  MANAGEMENT CLASS          - Specify Data Set Backup and Migration Criteria
 4  DATA CLASS                - Specify Data Set Allocation Parameters
 5  STORAGE CLASS             - Specify Data Set Performance and Availability
 6  STORAGE GROUP             - Specify Volume Names and Free Space Thresholds
 7  AUTOMATIC CLASS SELECTION - Specify ACS Routines and Test Criteria
 8  CONTROL DATA SET          - Specify System Names and Default Criteria
 9  AGGREGATE GROUP           - Specify Data Set Recovery Parameters
10  LIBRARY MANAGEMENT        - Specify Library and Drive Configurations
 C  DATA COLLECTION           - Process Data Collection Function
 L  LIST                      - Perform Functions Against Saved ISMF Lists
 R  REMOVABLE MEDIA MANAGER   - Perform Functions Against Removable Media
 X  EXIT                      - Terminate ISMF

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

Figure 35. Validate the Configuration (1 of 2)

```
                         CDS APPLICATION SELECTION
COMMAND ===>

TO PERFORM CONTROL DATA SET OPERATIONS, SPECIFY:

  CDS NAME ===> 'SMS.SCDS1.SCDS'
                         (1 to 44 character data set name or 'ACTIVE')

SELECT ONE OF THE FOLLOWING OPTIONS ===> 4

 1  DISPLAY  - Display the Base Configuration
 2  DEFINE   - Define the Base Configuration
 3  ALTER    - Alter the Base Configuration
 4  VALIDATE - Validate the SCDS
 5  ACTIVATE - Activate the CDS




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

Figure 36. Validate the Configuration (2 of 2)

The output from the VALIDATE function should look like the following:

```
                              VALIDATION RESULTS

VALIDATION RESULT:  VALIDATION SUCCESSFUL
SCDS NAME:          SMS.SCDS1.SCDS
ACS ROUTINE TYPE:   *
DATE OF VALIDATION: 1995/08/13
TIME OF VALIDATION: 17:40
```

Figure 37. Output from the VALIDATE Function

Step 13 Activate the configuration

If the validation runs without any failure messages, the configuration can be activated.

```
                          CDS APPLICATION SELECTION
COMMAND ===>

TO PERFORM CONTROL DATA SET OPERATIONS, SPECIFY:

  CDS NAME ===> 'SMS.SCDS1.SCDS'
                          (1 to 44 character data set name or 'ACTIVE')

SELECT ONE OF THE FOLLOWING OPTIONS ===> 5

 1  DISPLAY  - Display the Base Configuration
 2  DEFINE   - Define the Base Configuration
 3  ALTER    - Alter the Base Configuration
 4  VALIDATE - Validate the SCDS
 5  ACTIVATE - Activate the CDS




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

Figure 38. Activate the Configuration

An outstanding reply will come out on the operators console to verify the activation of SMS:

```
IGD043D REPLY 'Y' TO ALLOW ACTIVATION OF A CONFIGURATION BY WELLIE2 ,
        'N' TO DENY THE REQUEST
```

With the ACTIVATE command the defined configuration is now stored into the SMS.ACDS1.ACDS data set.

To use SMS, an IPL has to be done because SYS1.PARMLIB members have been changed (IEFSSNxx, IGDSMSxx). If no changes to SYS1.PARMLIB have been done, SMS can be activated by the SET SMS=xx command where xx defines the IGDSMSxx Parmlib member.

Now DFSMS is active and ready to allocate HFS data sets with a high level qualifier of OMVS. If this qualifier is not acceptable at your installation, it can be easily changed by changing the Automatic Class Selection (ACS) routine for the storage class and going through the validation process again.

Note: The SMS configuration described above does not contain a data class and a management class definition. A data class simplifies the allocation of MVS data sets by allowing the storage administrator to define allocation defaults and to set up installation-wide standards to data set allocation.

A management class is a list of data set migration, backup and retention attribute values. Management class also includes object expiration criteria, object backup requirements, and class transition criteria for management of objects. DFSMShsm uses the attributes of the management class associated with a data set to manage storage.

Our goal was to only provide a mini DFSMS configuration for our OpenEdition MVS work. In this mini configuration, data class and management class are not needed.

2.5 APPC

OpenEdition MVS uses initiators provided by APPC/MVS. Whether you are currently using APPC or not, you will have to customize it for OpenEdition MVS.

2.5.1 Customizing APPC/MVS for OpenEdition MVS

Step 1 Create a VSAM KSDS data set for the TP profile

Before creating a TP profile for OpenEdition MVS, a virtual storage access method key-sequenced data set (VSAM KSDS) for a TP profile repository has to be defined. The JCL for this job is in member TPPROF in data set OMVS.JCL.

Step 2 Create TP profile for OpenEdition MVS

The job needed to create the TP profile for OpenEdition MVS is member BPXISTPA in data set OMVS.JCL.

Notes:

a. If this is the first time running this job expect a return code of 08 on the delete step.

b. OpenEdition MVS does not require an APPC sideinfo data set because it prepares the ALLOCATE request in storage. This avoids definitions being made in a side information data set.

Step 3 Customize the APPCPMxx parmlib member

The APPC ALLOCATE for OpenEdition MVS is formulated for a logical unit with the LU=OWN attribute. OpenEdition MVS requires that the base LU be associated with the APPC scheduler as shown in Figure 39 on page 31. If you have multiple LUADD statements in your configuration, make sure that there is only one defined with the parameters, SCHED(ASCH) and BASE together. You may have multiple LUADD statements with SCHED(ASCH) but only one LUADD statement can have the BASE parameter. If more than one LUADD statement is defined with the BASE parameter the system will use the last one in the list. For further information see *Planning: APPC Management*.

If APPC/MVS is already active at your system, add the following statement to member APPCPMxx, replacing the ACBNAME and TPDATA parameters with the those defined on your system.

Note: The ACBNAME in Figure 39 on page 31 does not have to correspond to a real VTAM APPL statement since OpenEdition MVS does not do any APPC outbound requests that would use VTAM services, but you must fill in this parameter with a dummy name anyway. We will use the name APPCOMVS.

```
LUADD ACBNAME(APPCOMVS) SCHED(ASCH) BASE
      TPDATA(SYS1.OMVS.APPCTP) TPLEVEL(SYSTEM)
```

Figure 39. LUADD Statement for Member APPCPMxx

If APPC/MVS is not active get member APPCPM00 from OMVS.PARMLIB.

Step 4 Customize RACF for APPC

Optional

This step is optional. RACF protecting the ACBNAME limits the opening of this ACB to APPC and other authorized programs. Programs like CICS opens their ACBs in an unauthorized state. If you are running CICS and if CICS needs to use this particular ACBNAME you will need to RACF permit the CICS user ID to this RACF profile protecting this resource.

The ACBNAME can be defined and protected through RACF with the following RACF commands:

```
RDEFINE VTAMAPPL APPCOMVS UACC(NONE)
SETROPTS CLASSACT(VTAMAPPL) RACLIST(VTAMAPPL)
```

Figure 40. RACF Commands on Behalf of APPC

Step 5 Customize the ASCHPMxx parmlib member

If APPC/MVS is already active at your system, add the following statement to member ASCHPMxx:

```
CLASSADD CLASSNAME(OPENMVS) MIN(15) MAX(300) RESPGOAL(1)
```

Figure 41. CLASSADD Statement for Member ASCHPMxx

If APPC/MVS is not active get member ASCHPM00 from OMVS.PARMLIB.

Step 6 Refresh or start APPC

If you are already running APPC on your system you will need to refresh it to pick up the changes for OpenEdition MVS. From the Operators console issue the following:

```
SET APPC=xx   (where xx is the suffix of your APPCPMxx member)
SET ASCH=xx   (where xx is the suffix of your ASCHPMxx member)
```

Figure 42. Refresh APPC with OE Changes

If you are not currently running APPC you will need to start it. Make sure you have an APPC and an ASCH proc in your procedure library, then start them:

```
START APPC,SUB=MSTR,APPC=xx (where xx is suffix of APPCPMxx member)
START ASCH,SUB=MSTR,ASCH=xx (where xx is suffix of ASCHPMxx member)
```

Figure 43. Starting APPC

The following message will appear:

```
ATB052E LOGICAL UNIT APPCOMVS FOR TRANSACTION SCHEDULER ASCH 169
NOT ACTIVATED IN THE APPC CONFIGURATION. REASON CODE = 5A.
```

A display command shows that the local LU now has status pending.

```
D APPC,LU,A
ATB101I  10.04.31  APPC DISPLAY 177
  ACTIVE LU'S    OUTBOUND LU'S    PENDING LU'S    TERMINATING LU'S
     00000           00000           00001             00000
  SIDEINFO=*NONE*
LLUN=APPCOMVS          SCHED=ASCH          BASE=YES
   STATUS=PENDING      PARTNERS=00000      TPLEVEL=SYSTEM
   TPDATA=SYS1.OMVS.APPCTP
```

A status of pending is normal because OpenEdition MVS doesn't do any APPC outbound requests.

2.6 RACF

This task will perform the steps necessary to get RACF customized in order to start OpenEdition MVS.

Note: If you are using CA-ACF2 or CA-Top Secret, see Appendix D, “CA-ACF2 Support for OpenEdition MVS” on page 209 and Appendix E, “CA-Top Secret Support for OpenEdition MVS” on page 217.

2.6.1 UNIX Security Basics

On UNIX systems each user needs an account that is made up of a user name, and a password. UNIX uses the /etc/passwd file to keep track of user name and an encrypted password for every user on the system. Internally, the UNIX operating system uses a numeric ID to refer to a user, so in addition to user name and password, the /etc/passwd file also contains a numeric ID called the user identifier or UID. UNIX operating systems differ, but generally these UIDs are unsigned 16 bit numbers, ranging from 0 to 65535. The UID is the actual number that the operating system uses to identify the user. User names are provided as a convenience, as an easy way for us to remember our sign on to the UNIX system. If two users are assigned the same UID, UNIX views them as the same user, even if they have different user names and passwords. Two users with the same UID can freely read and write over each other′s files and can kill each other′s processes. Assigning the same UID to multiple users is generally not recommended.

UNIX systems also have the concept of groups where you group together many users who need to access a set of common files, directories, or devices. Like user names and UIDs, groups have both group names and group identification numbers (GIDs). Each user belongs to a primary group that is stored in the /etc/passwd file on UNIX systems.

Along with user name, encrypted password, UID, and GID, the /etc/passwd file also contains the user′s full name, the user′s home directory (the directory where a user would generally store their files) and the file name of the shell program that is executed when the user initially logs in.

2.6.1.1 UNIX Security and Files

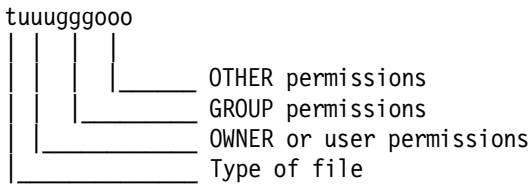
In UNIX systems, general information about each file is stored with the files in the file system. Included with this general information (such as, the file size and last modification date) is the security information for each file:

- The file′s owner (as represented by a UID)
- The file′s group (as represented by a GID)
- The file′s mode bits or file permissions bits

All UNIX files have three types of permissions; read, write and execute (displayed as r, w and x respectively). These permissions are stored with each file as file permission bits that are split into three groups:

- The file owner
- The file owners group
- All other users

The permission bits are set in the same way for each level of access using a simple binary 10-bit code.



The type of file can be:

- - regular file
- c character special file
- d directory
- l symbolic link
- p FIFO special file

The first group of three characters describes the owner permission, the second describes group permission, and the third describes other (or sometimes called world) permissions as shown in Table 5.

Table 5. File Access Types and Permission Bits

Pos.	Char.	Access Type	Permission for File	Permission for Directory
1	r	Read	Permission to read or print the contents.	Permission to read, but not search, the contents.
2	w	Write	Permission to change, add to, or delete from the contents.	Permission to change, add, or delete directory entries.
3	x	Execute or Search	Permission to run the file. This permission is used for executable files.	Permission to search the directory.
any	-	No access		

Permission bits are usually presented in a 3-digit octal number, where the first digit describes owner permissions, the second digit describes group permissions, and the third digit describes permissions for all others.

For each type of access, owner, group, and other, there is a corresponding octal number:

0 No access (---)

1 Execute-only access (--x)

2 Write-only access (-w-)

3 Write and execute (-wx)

4 Read-only access (r--)

5 Read and execute access (r-x)

6 Read and write access (rw-)

7 Read, write, and execute access (rwx)

Some typical 3-digit permissions are specified in octal in this way:

666 Owner (rw-) group(rw-) other(rw-)

700 Owner (rwx) group(---) other(---)

755 Owner (rwx) group(r-x) other(r-x)

777 Owner (rwx) group(rwx) other(rwx)

A user may set permission bits for any combination at any level of access. For example, if a user wishes to have, read, write and execute for their own files, but not allow access to any other member of their RACF defined default group, or any other OpenEdition MVS user, you would set the permission bits to 700, which is displayed as: *-rwx------*.

An example is shown in Figure 44 after the `ls -l` command was issued from the OpenEdition MVS shell:

```
$ ls -l
total 1600
-rw-r--r--   1 rich     system      2564 Nov 16 1994  phone.list
-rw-------   1 rich     itsc         978 Jun 14 13:14 machine.serials
drwxr-x---   2 rich     itsc         512 Sep 29 10:11 ocs
$
```

Figure 44. File Permission Bits

In the above example, the phone.list file can only be updated by rich, but everyone else on the system can read its contents. The machine.serials file can only be updated and read by rich.

The new OpenEdition MVS shell and utilities user may require their HFS to be protected from other uses, as would be usual for traditional MVS data sets.

2.6.2 OpenEdition MVS Security with RACF

With OpenEdition MVS the concept of user accounts are the same but the method of storing this account information is different. RACF, when used with OpenEdition MVS, integrates the OpenEdition MVS account information with the existing MVS account and system information to provide a central secure data base in which to store all your security information.

As with UNIX, each OpenEdition MVS user needs a UID and GID. These UIDs and GIDs are used by OpenEdition MVS to control the files and processes that a user may use. OpenEdition MVS security functions are implemented in RACF, partially as modifications to existing RACF functions and partially as new RACF functions. The security functions provided include user validation, file access checking, and privileged user checking.

OpenEdition MVS users are defined with RACF commands. When a job starts or a user logs on, the user ID and password are verified by existing MVS and RACF functions. When an address space requests an OpenEdition MVS function for the first time, RACF:

1. Verifies that the user is defined as an OpenEdition MVS user.
2. Verifies that the user's current connect group is defined as an OpenEdition MVS group.
3. Initializes the control blocks needed for subsequent security checks.

You can control access for the following users:

1. Owner: the owner of the file or directory, whose UID matches the UID for the file.
2. Group: a member of the group whose GID matches the GID for the file.
3. Others: anyone else

Therefore, it is recommended that you assign a unique UID to each user ID on your system and specify a unique GID for each group.

The RACF database contains, among other things, user profiles and group profiles. Within these user and group profiles is a new segment called the OMVS segment. The OMVS segment contains the UID and GID definitions for each user and group respectively. Your user profile may contain OpenEdition MVS information about you, in the OMVS segment.

The details RACF lists from the OMVS segment of the user profile are:

- The user identifier (UID)
- The initial directory path name (HOME)
- The program path name (PROGRAM)

Now, let's start customizing our RACF environment.

Warning: The following tasks will require that you have RACF SPECIAL attribute authority to make additions or changes to the RACF database.

On the next few pages you will find a set of tasks to define the RACF environment in order to install and run OpenEdition MVS. Included are examples using RACF panels and RACF JCL to define this environment. If you are going to use the RACF panels they will need to be customized. One way of gaining access to the RACF panels is to concatenate the RACF data sets in your TSO logon procedure as shown in Figure 45.

```
//ISPPLIB   DD DSN=ISP.V4R1M0.ISPPENU,DISP=SHR
//          DD DSN=SYS1.HRFPANL,DISP=SHR
//ISPMLIB   DD DSN=ISP.V4R1M0.ISPMENU,DISP=SHR
//          DD DSN=SYS1.HRFMSG,DISP=SHR
//ISPSLIB   DD DSN=ISP.V4R1M0.ISPSENU,DISP=SHR
//          DD DSN=SYS1.HRFSKEL,DISP=SHR
//SYSPROC   DD DSN=ISP.V4R1M0.ISPCLIB,DISP=SHR
//          DD DSN=SYS1.HRFCLST,DISP=SHR
```

Figure 45. RACF Panels Setup

After you have access to the RACF data sets from your TSO session, you then have to modify an existing ISPF panel, ISR@PRIM for example, to provide entry into panel ICHP00, the RACF Selection menu. The additional entry should look like the following:

- R,'PANEL(ICHP00)'

OR

If you don't want to modify an existing ISPF panel you can start the RACF panels from TSO READY.

- From TSO READY type in: ISPSTART PANEL(ICHP00)

For more information on using the RACF panels see *RACF General User's Guide*.

2.6.3 Defining RACF Groups for OpenEdition MVS

The first thing that you need to do is to add RACF profile definitions for new RACF groups or alter RACF profile definitions for existing RACF groups to define an OMVS segment containing a valid GID.

Note: RACF does not require the GID to be unique. The same numeric value can be assigned to multiple groups, but this is not recommended because individual group control would be lost.

2.6.3.1 Define the TTY Group

Certain shell programs, such as talk and write, require pseudoterminals to have a group name of TTY. When a user logs in, or issues the OMVS command from TSO/E, the group name associated with these terminals is changed to TTY. A group by that name must be defined to RACF. This group should be given a unique GID and no users should be connected to this group.

Run member RACF00 in the OMVS.JCL data set. It contains a JOB to define a group profile for TTY.

OR

Use the RACF interactive panels as shown in the next set of figures.

Note: The group name TTY is hardcoded, it cannot be changed or customized.

Select option 3 of the RACF main menu.

```
                        RACF - SERVICES OPTION MENU
OPTION ===>  3

SELECT ONE OF THE FOLLOWING:

   1  DATA SET PROFILES

   2  GENERAL RESOURCE PROFILES

   3  GROUP PROFILES AND USER-TO-GROUP CONNECTIONS

   4  USER PROFILES AND YOUR OWN PASSWORD

   5  SYSTEM OPTIONS

  99  EXIT
```

Figure 46. Add the TTY Group with an OMVS Segment (1 of 5)

Select option 1 in order to add this new group.

```
                    RACF - GROUP PROFILE SERVICES
OPTION ===>  1

SELECT ONE OF THE FOLLOWING.

   1  ADD           Add a group profile
   2  CHANGE        Change a group profile
   3  DELETE        Delete a group profile
   4  CONNECT       Add or change a user connection
   5  REMOVE        Remove users from the group


   8  DISPLAY       Display profile contents
   9  SEARCH        Search the RACF data base for profiles


ENTER THE FOLLOWING INFORMATION.

   GROUP NAME       ===>  TTY
```

Figure 47. Add the TTY Group with an OMVS Segment (2 of 5)

Fill all the specifications with your installation requirements, and select the OMVS parameters.

```
                         RACF - ADD GROUP TTY
COMMAND ===>

Enter the following information:

   OWNER                         ===>           Userid or group name
   SUPERIOR GROUP                ===>
   USE TERMINAL UACC             ===>           YES or NO

Identify a model profile for group data sets (optional):

   PROFILE NAME            ===>

To ADD the following optional information, enter any character:

  _  INSTALLATION DATA
  _  DFP PARAMETERS
  S  OMVS PARAMETERS
```

Figure 48. Add the TTY Group with an OMVS Segment (3 of 5)

Add a valid OMVS GID for the group you are adding. Remember to keep track of the OMVS GID that you assign for administrative purposes. The OMVS GID of 0 is unique for this group. You can select another unique OMVS GID based on your own installation criteria.

```
                          RACF - ADD GROUP TTY
                               OMVS PARAMETERS
 COMMAND ===>

 Enter OMVS segment information:

     GROUP IDENTIFIER   ===>  0            0 - 2147483647
```

Figure 49. Add the TTY Group with an OMVS Segment (4 of 5)

After the group has been added, you will receive the PROFILE ADDED message.

```
                     RACF - GROUP PROFILE SERVICES              PROFILE ADDED
OPTION ===>

SELECT ONE OF THE FOLLOWING.

   1  ADD           Add a group profile
   2  CHANGE        Change a group profile
   3  DELETE        Delete a group profile
   4  CONNECT       Add or change a user connection
   5  REMOVE        Remove users from the group


   8  DISPLAY       Display profile contents
   9  SEARCH        Search the RACF data base for profiles


ENTER THE FOLLOWING INFORMATION.

   GROUP NAME       ===> TTY
```

Figure 50. Add the TTY Group with an OMVS Segment (5 of 5)

2.6.3.2 Define a General Purpose Group for OpenEdition MVS

Optional

This step is optional. You can define a new RACF group for OpenEdition MVS users or you can can define a GID to an existing RACF group. See 2.6.3.3, “Change Existing Groups As OpenEdition MVS Groups” on page 43 for more information on how to add a valid GID to existing RACF groups.

To define a new RACF group:

Run member RACF01 in the OMVS.JCL data set. It contains a JOB to define a group profile.

OR

Use the RACF interactive panels, following the next set of figures.

This is the same task that you will perform for any other group that you want to add with a valid OMVS GID.

We have selected the name OMVSGRP for our OpenEdition MVS group. It is not mandatory to use this name. You can select any name you wish, to conform to your installations naming conventions.

Select option 3 of the RACF main menu.

```
                        RACF - SERVICES OPTION MENU
OPTION ===>  3

SELECT ONE OF THE FOLLOWING:

   1  DATA SET PROFILES

   2  GENERAL RESOURCE PROFILES

   3  GROUP PROFILES AND USER-TO-GROUP CONNECTIONS

   4  USER PROFILES AND YOUR OWN PASSWORD

   5  SYSTEM OPTIONS

  99  EXIT
```

Figure 51. Add a New Group with an OMVS Segment (1 of 5)

Select option 1 in order to add this new group.

```
                      RACF - GROUP PROFILE SERVICES
OPTION ===>  1

SELECT ONE OF THE FOLLOWING.

   1  ADD           Add a group profile
   2  CHANGE        Change a group profile
   3  DELETE        Delete a group profile
   4  CONNECT       Add or change a user connection
   5  REMOVE        Remove users from the group


   8  DISPLAY       Display profile contents
   9  SEARCH        Search the RACF data base for profiles


ENTER THE FOLLOWING INFORMATION.

   GROUP NAME       ===>  OMVSGRP
```

Figure 52. Add a New Group with an OMVS Segment (2 of 5)

Fill all the specifications with your installation requirements, and select the OMVS PARAMETERS.

```
                          RACF - ADD GROUP OMVSGRP
COMMAND ===>

Enter the following information:

   OWNER                          ===>            Userid or group name
   SUPERIOR GROUP                 ===>
   USE TERMINAL UACC              ===>            YES or NO

Identify a model profile for group data sets (optional):

   PROFILE NAME            ===>

To ADD the following optional information, enter any character:

  _  INSTALLATION DATA
  _  DFP PARAMETERS
  S  OMVS PARAMETERS
```

Figure 53. Add a New Group with an OMVS Segment (3 of 5)

Add a valid OMVS GID for the group you are adding. Remember to keep track of the OMVS GID given for administrative purposes. The OMVS GID of 1 selected is unique for this group. You can select a unique OMVS GID based on your own installation criteria.

```
                          RACF - ADD GROUP OMVSGRP
                               OMVS PARAMETERS
 COMMAND ===>

 Enter OMVS segment information:

     GROUP IDENTIFIER   ===>  1            0 - 2147483647
```

Figure 54. Add a New Group with an OMVS Segment (4 of 5)

After the group has been added, you will receive the PROFILE ADDED message.

```
                     RACF - GROUP PROFILE SERVICES              PROFILE ADDED
OPTION ===>

SELECT ONE OF THE FOLLOWING.

   1  ADD           Add a group profile
   2  CHANGE        Change a group profile
   3  DELETE        Delete a group profile
   4  CONNECT       Add or change a user connection
   5  REMOVE        Remove users from the group


   8  DISPLAY       Display profile contents
   9  SEARCH        Search the RACF data base for profiles


ENTER THE FOLLOWING INFORMATION.

   GROUP NAME       ===> OMVSGRP
```

Figure 55. Add a New Group with an OMVS Segment (5 of 5)

2.6.3.3 Change Existing Groups As OpenEdition MVS Groups

In addition to adding new RACF groups, as we did in the previous section, you might want to assign OMVS GIDs to existing RACF groups if you want existing TSO users to be able to access OpenEdition MVS.

Warning: When a user connects to the system, that is, logs on to a TSO/E session or uses `rlogin` to connect to the shell, a RACF group is selected as the user's current group. In most cases, if not overridden by the user, the RACF default group that is defined in the user's RACF profile will get control. This RACF default group is the user's connect group and *must* have a valid GID in order for the user to use OpenEdition MVS services. For more information see *MVS/ESA Planning: OpenEdition MVS*.

For example, you may want all users that have a default RACF group of SYS1 to have the ability to use OpenEdition MVS.

In order to do this you can run member RACF02 in the OMVS.JCL data set.

OR

Follow the interactive RACF panels as follows.

Select option 3 from the RACF main menu.

```
                        RACF - SERVICES OPTION MENU
OPTION ===>  3

SELECT ONE OF THE FOLLOWING:

   1  DATA SET PROFILES

   2  GENERAL RESOURCE PROFILES

   3  GROUP PROFILES AND USER-TO-GROUP CONNECTIONS

   4  USER PROFILES AND YOUR OWN PASSWORD

   5  SYSTEM OPTIONS

  99  EXIT
```

Figure 56. Add an OMVS Segment to an Existing RACF Group (1 of 5)

Since we are changing an existing group, select option 2 and specify the name of the group.

```
                      RACF - GROUP PROFILE SERVICES
OPTION ===>  2

SELECT ONE OF THE FOLLOWING.

   1  ADD           Add a group profile
   2  CHANGE        Change a group profile
   3  DELETE        Delete a group profile
   4  CONNECT       Add or change a user connection
   5  REMOVE        Remove users from the group


   8  DISPLAY       Display profile contents
   9  SEARCH        Search the RACF data base for profiles


ENTER THE FOLLOWING INFORMATION.

   GROUP NAME       ===>  SYS1
```

Figure 57. Add an OMVS Segment to an Existing RACF Group (2 of 5)

Select OMVS PARAMETERS, so you can enter the new OMVS definitions for the group selected.

```
                        RACF - CHANGE GROUP SYS1
COMMAND ===>

Enter the desired group profile changes.

   OWNER                    ===>            Userid or group name
   SUPERIOR GROUP           ===>
   USE TERMINAL ACCESS      ===>            YES or NO

Change the model profile used for group data sets (optional)

   MODEL PROFILE NAME      ===>
   STOP USING A MODEL      ===>            YES, if no model is to be
                                           used.

To CHANGE the following optional information, enter any character:

   _  INSTALLATION DATA
   _  DFP PARAMETERS
   S  OMVS PARAMETERS
```

Figure 58. Add an OMVS Segment to an Existing RACF Group (3 of 5)

Assign an OMVS GID to the group. It is recommended that you use a unique GID number.

```
                     RACF - CHANGE GROUP  SYS1
                     OMVS PARAMETERS

COMMAND ===>


Select only ONE option to CHANGE or DELETE, then press ENTER:


 Delete ALL OMVS information     (NOOMVS) ===>        ENTER YES to DELETE

     -- OR --

 Specify a NEW Group Identifier     (GID) ===>  2          0 - 2147483647

     -- OR --

 Delete ONLY the Group Identifier (NOGID) ===>        ENTER YES to DELETE
```

Figure 59. Add an OMVS Segment to an Existing RACF Group (4 of 5)

You will receive the message PROFILE CHANGED after the task is done.

```
                    RACF - GROUP PROFILE SERVICES              PROFILE CHANGED
OPTION ===>

SELECT ONE OF THE FOLLOWING.

   1  ADD           Add a group profile
   2  CHANGE        Change a group profile
   3  DELETE        Delete a group profile
   4  CONNECT       Add or change a user connection
   5  REMOVE        Remove users from the group


   8  DISPLAY       Display profile contents
   9  SEARCH        Search the RACF data base for profiles


ENTER THE FOLLOWING INFORMATION.

   GROUP NAME       ===> SYS1
```

Figure 60. Add an OMVS Segment to an Existing RACF Group (5 of 5)

Repeat this process to assign GIDs to as many RACF groups as needed.

2.6.4 Defining a Superuser to Run SMP/E Jobs

Now that you have your groups with a OMVS GID assigned to them, the next step will be to assign a user ID that has superuser authority. A superuser on UNIX systems (including OpenEdition MVS) is a user that has special rights and privileges. This user ID must have a UID of 0 and its current RACF connect group must have a valid GID in order to SMP/E install the rest of the OpenEdition MVS features.

There are several ways to perform this task.

Run member RACF03 in the OMVS.JCL data set.

OR

Use the interactive RACF panels as shown in the following figures.

If the user has already been defined as a TSO/E user, the next step is to alter the original RACF profile, so an OMVS segment can be defined. For example, in the next few screens we will define an OMVS segment for an existing user ID called SYSPROG.

From the RACF - Services Option Menu select option 4.

```
                          RACF - SERVICES OPTION MENU
OPTION ===> 4

SELECT ONE OF THE FOLLOWING:

   1  DATA SET PROFILES

   2  GENERAL RESOURCE PROFILES

   3  GROUP PROFILES AND USER-TO-GROUP CONNECTIONS

   4  USER PROFILES AND YOUR OWN PASSWORD

   5  SYSTEM OPTIONS

  99  EXIT
```

Figure 61. Add an OMVS Segment to an Existing TSO/E User (1 of 9)

The next step is to provide the user ID of the TSO/E user and select option 2 in order to change the RACF profile. In this case the user ID is SYSPROG.

```
                       RACF - USER PROFILE SERVICES
OPTION ===>  2

SELECT ONE OF THE FOLLOWING:

   1    ADD          Add a user profile
   2    CHANGE       Change a user profile
   3    DELETE       Delete a user profile
   4    PASSWORD     Change your own password or interval
   5    AUDIT        Monitor user activity (Auditors only)


   8    DISPLAY      Display profile contents
   9    SEARCH       Search the RACF data base for profiles


ENTER THE FOLLOWING INFORMATION:

   USER     ===>  SYSPROG    Userid
```

Figure 62. Add an OMVS Segment to an Existing TSO/E User (2 of 9)

If the user's current default group does not have an OMVS GID assigned to it, change the user's default group to a valid RACF group that has a GID assigned. In our case, SYSPROG's default group was already SYS1.

```
                        RACF - CHANGE USER SYSPROG
COMMAND ===>

ENTER THE DESIRED CHANGES:

   OWNER                    ===>            Userid or group name

   USER NAME                ===>

   DEFAULT GROUP            ===> SYS1       Group name

   PASSWORD                 ===>            User's initial password
                            ===>            Re-enter password to verify

   PASSWORD INTERVAL        ===>            1 - 254 days, NO, or blank

   REVOKE                   ===>            YES, mm/dd/yy (date) or blank

   RESUME                   ===>            YES, mm/dd/yy (date) or blank
```

Figure 63. Add an OMVS Segment to an Existing TSO/E User (3 of 9)

On the next screen, you need to specify YES for the optional information.

```
                        RACF - CHANGE USER SYSPROG
COMMAND ===>

TO ASSIGN A USER ATTRIBUTE, ENTER YES
TO CANCEL A USER ATTRIBUTE, ENTER NO

   GROUP ACCESS       ===>                SPECIAL         ===>
   ADSP               ===>                OPERATIONS      ===>
   OIDCARD            ===>                AUDITOR         ===>
   NO-PASSWORD        ===>

CHANGE OR DELETE THE MODEL PROFILE USED FOR USER DATA SETS (OPTIONAL):

   NEW MODEL          ===>
   DELETE             ===>          YES if no model is to be used


TO ADD OR CHANGE OPTIONAL INFORMATION, ENTER YES    ===> YES
```

Figure 64. Add an OMVS Segment to an Existing TSO/E User (4 of 9)

This screen will allow you to select the OMVS parameters, so all the necessary information about OpenEdition MVS can be added to the user's RACF profile.

```
                          RACF - CHANGE USER SYSPROG
COMMAND ===>


To add or change the following information, enter any character.

  _ CLASS AUTHORITY
  _ INSTALLATION DATA
  _ SECURITY LEVEL or CATEGORIES
  _ SECURITY LABEL
  _ LOGON RESTRICTIONS
  _ NATIONAL LANGUAGES

  _ DFP PARAMETERS
  _ TSO PARAMETERS
  _ OPERPARM PARAMETERS
  _ CICS PARAMETERS
  _ WORK ATTRIBUTES
  S OMVS PARAMETERS
```

Figure 65. Add an OMVS Segment to an Existing TSO/E User (5 of 9)

This screen presents different options for the OpenEdition MVS environment. The SYSPROG user is being assigned a UID of 0, so this user is being defined as a *superuser*. The HOME and PROGRAM parameters are both being selected.

```
                        RACF - CHANGE USER SYSPROG
                        OMVS PARAMETERS
COMMAND ===>


 Delete ALL OMVS information (NOOMVS)  ===>           Enter YES to DELETE

    -- OR --

Select only ONE option from each group to CHANGE or DELETE, then press ENTER.

 Specify new User Identifier    (UID)  ===> 0            0 - 2147483647
 Delete User Identifier       (NOUID)  ===>           Enter YES to DELETE


 Change Initial Path Name      (HOME)  ===> YES        Enter YES to CHANGE
 Delete Initial Path Name    (NOHOME)  ===>           Enter YES to DELETE


 Change Program Path Name   (PROGRAM)  ===> YES        Enter YES to CHANGE
 Delete Program Path Name (NOPROGRAM)  ===>           Enter YES to DELETE
```

Figure 66. Add an OMVS Segment to an Existing TSO/E User (6 of 9)

Set this parameter to reflect the initial directory path name for the user SYSPROG. Remember that an initial directory must be defined in the user's RACF OMVS segment to be able to enter the OpenEdition MVS environment. In this case, make the HOME directory the root (/).

```
                        RACF - CHANGE USER SYSPROG      2 of 3
                        OMVS PARAMETERS
COMMAND ===>


Enter Initial Directory Path Name (HOME) below, then press ENTER:

  => /
  =>
  =>
  =>
  =>
```

Figure 67. Add an OMVS Segment to an Existing TSO/E User (7 of 9)

Enter the name of the program that is executed when this user enters the shell. The OpenEdition MVS shell and utilities feature contains a POSIX compliant shell, in our case it is /bin/sh.

```
                        RACF - CHANGE USER SYSPROG      3 of 3
                        OMVS PARAMETERS
COMMAND ===>


Enter Program Path Name (PROGRAM), then press ENTER:

  => /bin/sh
  =>
  =>
  =>
```

Figure 68. Add an OMVS Segment to an Existing TSO/E User (8 of 9)

The message PROFILE CHANGED will be received.

```
                    RACF - USER PROFILE SERVICES              PROFILE CHANGED
OPTION ===>

SELECT ONE OF THE FOLLOWING:

   1    ADD          Add a user profile
   2    CHANGE       Change a user profile
   3    DELETE       Delete a user profile
   4    PASSWORD     Change your own password or interval
   5    AUDIT        Monitor user activity (Auditors only)


   8    DISPLAY      Display profile contents
   9    SEARCH       Search the RACF data base for profiles


ENTER THE FOLLOWING INFORMATION:

   USER     ===> SYSPROG     Userid
```

Figure 69. Add an OMVS Segment to an Existing TSO/E User (9 of 9)

2.6.5 Provide OpenEdition MVS Definitions for Other Resources

For any product that requires OpenEdition MVS services, that product must be associated with a RACF user ID that has a valid UID and a RACF group that has a valid GID.

For data gathering, RMF invokes Monitor III procedure RMFGAT to obtain OpenEdition MVS data. A RACF user ID, RMFGAT was created with a valid UID and having a default RACF group with a valid GID. The RMFGAT procedure was associated with a RACF user ID and RACF group as shown in the following:

PROCNAME	RACF user ID	UID	RACF group	GID
RMFGAT	RMFGAT	222	OMVSGRP	1

A TSO/E segment was not included for this profile since it will not be used to log on to TSO. A group called OMVSGRP with an unique OMVS GID was created before, and it was assigned as the default group for this user.

If you are defining a simple transport provider as explained in 7.1.1, “Setting Up a Simple TCP/IP Environment” on page 153, you need to define a RACF user ID with an OMVS segment as shown in the following:

PROCNAME	RACF user ID	UID	RACF group	GID
TCPIP	TCPIP	444	OMVSGRP	1

In our environment, we selected to define two transport providers, one for the MVS environment, and a second one for the OpenEdition MVS environment. For more detailed information refer to Figure 176 on page 161. Each Each procedure was associated with a different RACF user ID as shown in the following:

PROCNAME	RACF user ID	UID	RACF group	GID
TCPIPMVS	TCPIPMVS	445	OMVSGRP	1
TCPIPOE	TCPIPOE	446	OMVSGRP	1

In this way we can differentiate the two transport providers because many times error messages will display the RACF user ID the procedure is associated with, not the procedure name that is having a problem. A TSO/E segment was not included for these profiles since they will not be used to log on to TSO. A group called OMVSGRP with a unique OMVS GID was created before, and it was assigned as the default group for these user IDs.

2.6.6 RACF Requirements to Run the OpenEdition MVS Kernel

The OMVS cataloged procedure runs a program that initializes the kernel address space. This procedure will need to be associated with a RACF user ID that is a superuser (UID=0). The RACF user ID along with a RACF group, that has a valid GID, will need to be added to either the RACF Started Task Table or the RACF Started Class.

2.6.6.1 Defining the OMVSKERN User ID

Generally all started tasks need to be associated with a RACF user ID and RACF group when they are started. Additionally, this RACF user ID and RACF group that is associated with the OMVS started task also needs to have a valid UID and GID respectively. We will use the user ID of OMVSKERN for our installation, but this name is not mandatory, it can be changed to fit your installation naming conventions. The only requirement is that it has a UID of 0 because it will be the RACF user ID associated with starting the kernel address space. There is no need to define a TSO segment for this user ID.

To define user ID, OMVSKERN:

Run member RACF04 in the OMVS.JCL data set (change it to fit your requirements),

OR

Follow the next figures representing the RACF interactive panels to define the OMVSKERN user ID.

Select option 4 of the RACF - Services Option Menu.

```
                        RACF - SERVICES OPTION MENU
OPTION ===> 4

SELECT ONE OF THE FOLLOWING:

   1  DATA SET PROFILES

   2  GENERAL RESOURCE PROFILES

   3  GROUP PROFILES AND USER-TO-GROUP CONNECTIONS

   4  USER PROFILES AND YOUR OWN PASSWORD

   5  SYSTEM OPTIONS

  99  EXIT
```

Figure 70. Add the OMVSKERN User ID (1 of 9)

Next step is to select option 1 of the following menu and to provide the user ID of the new user.

```
                        RACF - USER PROFILE SERVICES
OPTION ===>  1

SELECT ONE OF THE FOLLOWING:

   1    ADD          Add a user profile
   2    CHANGE       Change a user profile
   3    DELETE       Delete a user profile
   4    PASSWORD     Change your own password or interval
   5    AUDIT        Monitor user activity (Auditors only)


   8    DISPLAY      Display profile contents
   9    SEARCH       Search the RACF data base for profiles


ENTER THE FOLLOWING INFORMATION:

   USER     ===> OMVSKERN       Userid
```

Figure 71. Add the OMVSKERN User ID (2 of 9)

Fill the blanks with all the information necessary. Remember to provide the user with a default group with a valid OMVS GID assigned.

```
                           RACF - ADD USER OMVSKERN
COMMAND ===>

ENTER THE FOLLOWING INFORMATION:

   OWNER                     ===>            Userid or group name

   USER NAME                 ===>

   DEFAULT GROUP             ===> OMVSGRP      Group name

   PASSWORD                  ===>            User's initial password
                             ===>            Re-enter password to verify

   PASSWORD INTERVAL         ===>            1 - 254 (days), NO, or blank


                         Press ENTER to continue.
```

Figure 72. Add the OMVSKERN User ID (3 of 9)

Specify YES for the optional information to add the OMVS segment information.

```
                          RACF - ADD USER OMVSKERN
COMMAND ===>

TO ASSIGN USER ATTRIBUTES, ENTER YES:

   GROUP ACCESS       ===> NO               SPECIAL         ===> NO
   ADSP               ===> NO               OPERATIONS      ===> NO
   OIDCARD            ===> NO               AUDITOR         ===> NO
   NO-PASSWORD        ===> NO

IDENTIFY THE MODEL PROFILE FOR USER DATA SETS (OPTIONAL):

   MODEL PROFILE      ===>

TO CREATE THE FOLLOWING, ENTER YES (OPTIONAL):

   A GENERIC DATA SET PROFILE          ===> NO
   A MINIDISK PROFILE                  ===> NO


TO ADD OPTIONAL INFORMATION, ENTER YES     ===> YES
```

Figure 73. Add the OMVSKERN User ID (4 of 9)

Select the parameters required for this user. In this case, OMVSKERN only requires an OMVS RACF segment.

```
                          RACF - ADD USER OMVSKERN
 COMMAND ===>


 To ADD the following information, enter any character:

   _ CLASS AUTHORITY
   _ INSTALLATION DATA
   _ GROUP AUTHORITY
   _ SECURITY LEVEL or CATEGORIES
   _ SECURITY LABEL
   _ LOGON RESTRICTIONS
   _ NATIONAL LANGUAGES

   _ DFP PARAMETERS
   _ TSO PARAMETERS
   _ OPERPARM PARAMETERS
   _ CICS PARAMETERS
   _ WORK ATTRIBUTES
   S OMVS PARAMETERS
```

Figure 74. Add the OMVSKERN User ID (5 of 9)

This user ID will be given a UID of 0 since the OMVS kernel needs to run as a superuser.

```
                          RACF - ADD USER OMVSKERN      1 of 3
                          OMVS PARAMETERS
 COMMAND ===>

 Enter User Identifier (UID) below, then press ENTER:

    USER IDENTIFIER    ===>  0           0 - 2147483647
```

Figure 75. Add the OMVSKERN User ID (6 of 9)

Set this parameter to reflect the initial directory path name for the user OMVSKERN. We are defining the root directory as the home directory for this user ID.

```
                          RACF - ADD USER OMVSKERN    2 of 3
                          OMVS PARAMETERS
COMMAND ===>

Enter Initial Directory Path Name (HOME), then press ENTER:

  => /
  =>
  =>
  =>
```

Figure 76. Add the OMVSKERN User ID (7 of 9)

Define the program path name that defines the shell program.

```
                          RACF - ADD USER OMVSKERN    3 of 3
                          OMVS PARAMETERS
COMMAND ===>

Enter Program Path Name (PROGRAM), then press ENTER:

  => /bin/sh
  =>
  =>
  =>
  =>
  =>
```

Figure 77. Add the OMVSKERN User ID (8 of 9)

Finally, you will receive the message PROFILE ADDED.

```
                    RACF - USER PROFILE SERVICES              PROFILE ADDED
OPTION ===>

SELECT ONE OF THE FOLLOWING:

   1    ADD          Add a user profile
   2    CHANGE       Change a user profile
   3    DELETE       Delete a user profile
   4    PASSWORD     Change your own password or interval
   5    AUDIT        Monitor user activity (Auditors only)


   8    DISPLAY      Display profile contents
   9    SEARCH       Search the RACF data base for profiles


ENTER THE FOLLOWING INFORMATION:

   USER     ===> OMVSKERN    Userid
```

Figure 78. Add the OMVSKERN User ID (9 of 9)

2.6.6.2 Define the OMVS Cataloged Procedure to RACF

With RACF 2.1 there are two ways to define a cataloged procedure to RACF:

1. The RACF started procedures table (module ICHRIN03). The entry in the RACF started procedures table will assign a user ID and a group ID to the OMVS address space.
2. The RACF STARTED class. This allows the installation to assign RACF identities to started procedures and jobs dynamically. In this way, there is no need for modifying code or re-IPL the system in order to add or modify RACF entries for started procedures.

For our examples, we have used the STARTED class instead of modifying the started procedures table (module ICHRIN03). Activate the RACF STARTED class, with the SETROPTS CLASSACT(STARTED) command if this is the first time you will use it.

Run member RACF05 in the OMVS.JCL data set.

OR

Use the interactive RACF panels as follows.

From the RACF main menu, select option 2, in order to define a profile for the STARTED class.

```
                          RACF - SERVICES OPTION MENU
OPTION ===> 2

SELECT ONE OF THE FOLLOWING:

   1  DATA SET PROFILES

   2  GENERAL RESOURCE PROFILES

   3  GROUP PROFILES AND USER-TO-GROUP CONNECTIONS

   4  USER PROFILES AND YOUR OWN PASSWORD

   5  SYSTEM OPTIONS

  99  EXIT
```

Figure 79. Add a STARTED Class Profile for the OMVS Cataloged Procedure (1 of 7)

Now select option 1 to add a profile.

```
              RACF - GENERAL RESOURCE PROFILE SERVICES
OPTION ===> 1

SELECT ONE OF THE FOLLOWING:

   1   ADD           Add a profile
   2   CHANGE        Change a profile
   3   DELETE        Delete a profile
   4   ACCESS        Maintain access list
   5   AUDIT         Monitor access attempts (Auditors only)


   8   DISPLAY       Display profile contents
   9   SEARCH        Search the RACF data base for profiles
```

Figure 80. Add a STARTED Class Profile for the OMVS Cataloged Procedure (2 of 7)

On this screen, you must specify the class you want to add profiles to. In our case, the class is the STARTED class. Create a profile with the name of the cataloged procedure.

```
                        RACF - GENERAL RESOURCE SERVICES -  ADD
OPTION ===>

ENTER THE FOLLOWING PROFILE INFORMATION:

   CLASS     ===> STARTED

   PROFILE   ===> OMVS.*


                         <==end of data

   USE A MODEL        ===>        YES or NO


      NOTE: Embedded Blanks are NOT ALLOWED in class or profile names.
```

Figure 81. Add a STARTED Class Profile for the OMVS Cataloged Procedure (3 of 7)

Change the information according to your installation requirements, and be sure to specify YES to add optional information related to the started class.

```
                    RACF - ADD GENERAL RESOURCE PROFILE
 COMMAND ===>

   CLASS:         STARTED
   PROFILE      _ OMVS.*


ENTER OR CHANGE THE FOLLOWING INFORMATION:

   OWNER                    ===>            Userid or group name
   LEVEL                    ===> 0          0-99
   FAILED ACCESSES          ===> FAIL       FAIL or WARN
   UACC                     ===> NONE       NONE, READ, UPDATE,
                                            CONTROL, ALTER or EXECUTE
   AUDIT SUCCESSES          ===> NOAUDIT    READ, UPDATE, CONTROL,
                                            ALTER or NOAUDIT
   AUDIT FAILURES           ===> READ       READ, UPDATE, CONTROL,
                                            ALTER or NOAUDIT
   NOTIFY                   ===>            Userid

 TO ADD OPTIONAL INFORMATION, ENTER YES     ===> YES
```

Figure 82. Add a STARTED Class Profile for the OMVS Cataloged Procedure (4 of 7)

Specify YES to get the STDATA information. This field is only for the STARTED class.

```
                    RACF - ADD GENERAL RESOURCE PROFILE
 COMMAND ===>

   CLASS:         STARTED
   PROFILE      _ OMVS.*


 To add optional information, enter YES:

    INSTALLATION DATA               ===>
    APPLICATION DATA                ===>
    ACCESS LIST                     ===>
    SECURITY LEVEL or CATEGORIES    ===>
    SECURITY LABEL                  ===>
    MEMBERS                         ===>
    TAPE VOLUME INFORMATION         ===>        TAPEVOL   class only
    TERMINAL INFORMATION            ===>        TERMINAL  class only
    VTAM SESSION                    ===>        APPCLU    class only
    DLF DATA                        ===>        DLFCLASS  class only
    STDATA INFORMATION              ===> YES    STARTED   class only
```

Figure 83. Add a STARTED Class Profile for the OMVS Cataloged Procedure (5 of 7)

Enter the information related to your OMVS cataloged procedure. In our case, we are giving the OMVS cataloged procedure the user ID of OMVSKERN for RACF association. For the group, we defined the OMVSGRP group.

The OMVS cataloged procedure is defined as trusted, so auditing can be requested. Trace is set to NO to avoid messages to the operator.

```
                    RACF - ADD GENERAL RESOURCE
                         STDATA INFORMATION
COMMAND ===>


 Enter the desired STDATA information, then press ENTER to continue


 Specify started task user     (USER) ===> OMVSKERN   Userid
 Use the procname as userid (=MEMBER) ===>            YES to use procname

 Specify started task group   (GROUP) ===> OMVSGRP    Group name
 Use the procname as group  (=MEMBER) ===>            YES to use procname

 Started task should run    (TRUSTED) ===> YES        YES or NO

 Started task should run (PRIVILEGED) ===> NO         YES or NO

 Issue message to operator    (TRACE) ===> NO         YES or NO
```

Figure 84. Add a STARTED Class Profile for the OMVS Cataloged Procedure (6 of 7)

At the end you will receive the PROFILE ADDED message.

```
             RACF - GENERAL RESOURCE PROFILE SERVICES          PROFILE ADDED
 OPTION ===>

 SELECT ONE OF THE FOLLOWING:

    1   ADD           Add a profile
    2   CHANGE        Change a profile
    3   DELETE        Delete a profile
    4   ACCESS        Maintain access list
    5   AUDIT         Monitor access attempts (Auditors only)


    8   DISPLAY       Display profile contents
    9   SEARCH        Search the RACF data base for profiles
```

Figure 85. Add a STARTED Class Profile for the OMVS Cataloged Procedure (7 of 7)

In our environment, we included several started procedures in the STARTED class.

The following started procedures were added to the STARTED class with a started task user ID of TCPIPMVS, and a started task group name of OMVSGRP

and run as TRUSTED. All these procedures will run with the same authority as the TCPIPMVS user ID:

- TCPIPMVS
- PORTMAPM
- ROUTEDMV
- NFSMVS
- EZAFTSRV

The following started procedures were added to the STARTED class with a started task user ID of TCPIPOE, and a started task group name of OMVSGRP and run as TRUSTED. All these procedures will run with the same authority as the TCPIPOE user ID:

- TCPIPOE
- PORTMAPO
- ROUTEDOE
- NFSOE

The definitions were made this way because we have defined two TCP/IP environments. For more detailed information, see the topic 7.1.2, “Setting Up the Multiple Transport Driver Support” on page 158.

The following started procedures would be added to the STARTED class if you were running a single TCP/IP environment. They would be added with a started task user ID of TCPIP, and a started task group name of OMVSGRP and run as TRUSTED. All these procedures will run with the same authority as the TCPIP user ID:

- TCPIP
- PORTMAP
- NFS
- EZAFTSRV

See 7.1.1, “Setting Up a Simple TCP/IP Environment” on page 153 for more detailed information.

The following started procedures were also added to the STARTED class with a started task user ID of OMVSKERN, and a started task group name of OMVSGRP and run as TRUSTED. This procedure will run with the same authority as the OMVSKERN user ID:

- INETD

The following started procedure was also added to the STARTED class with a started task user ID of RMFGAT, and a started task group name of OMVSGRP and run as TRUSTED. This procedure will run with the same authority as the RMFGAT user ID:

- RMFGAT

Notes:

1. Remember to activate the STARTED class, if this is your first cataloged procedure defined.
2. Even if you have used the STARTED class to define all your started procedures, you must have at least a dummy started procedures table (module ICHRIN03). RACF can not be initialized if the module ICHRIN03 is not present.

3. If you will use the STARTED class, it is recommended to have a basic ICHRIN03 in place, just in case the STARTED class is inactivated accidentally.
4. Repeat the task for all the started procedures that you want to add to the STARTED class.
5. After adding or changing profiles a RACF refresh is required, in order for the new information to be available. Refer to Appendix A, “Refreshing RACF Profiles” on page 195 on how to do a refresh.
6. For more information on the STARTED class and the RACF started procedures table (module ICHRIN03) refer to the *RACF System's Programmer Guide*.
7. A copy of the cataloged started procedure for OpenEdition MVS can be found in topic 2.10, “Starting OMVS” on page 93.

2.6.7 Define Regular OpenEdition MVS Users

Regular OpenEdition MVS users are those users that have a UID set to a value other than zero.

You could use different criteria to map UIDs. For example, if you have a group with a GID of 200, give all the users belonging to this group a UID between 200 and 299.

To add new users that can access OpenEdition MVS services via TSO and rlogin:

Run member RACF06 in the OMVS.JCL data set (change it to fit your requirements),

OR

Follow the next figures representing the RACF interactive panels to define new users.

Select option 4 of the RACF - Services Option Menu.

```
                        RACF - SERVICES OPTION MENU
OPTION ===> 4

SELECT ONE OF THE FOLLOWING:

   1  DATA SET PROFILES

   2  GENERAL RESOURCE PROFILES

   3  GROUP PROFILES AND USER-TO-GROUP CONNECTIONS

   4  USER PROFILES AND YOUR OWN PASSWORD

   5  SYSTEM OPTIONS

  99  EXIT
```

Figure 86. Add a New User with TSO/E and OMVS Segments (1 of 10)

Next step is to select option 1 of the following menu and to provide the user ID of the new user.

```
                      RACF - USER PROFILE SERVICES
OPTION ===>  1

SELECT ONE OF THE FOLLOWING:

   1    ADD          Add a user profile
   2    CHANGE       Change a user profile
   3    DELETE       Delete a user profile
   4    PASSWORD     Change your own password or interval
   5    AUDIT        Monitor user activity (Auditors only)


   8    DISPLAY      Display profile contents
   9    SEARCH       Search the RACF data base for profiles


ENTER THE FOLLOWING INFORMATION:

   USER     ===> USER1       Userid
```

Figure 87. Add a New User with TSO/E and OMVS Segments (2 of 10)

Fill in the blanks with all the information necessary. Remember to provide the user with a default group with a OMVS GID assigned to it.

```
                          RACF - ADD USER USER1
COMMAND ===>

ENTER THE FOLLOWING INFORMATION:

   OWNER                     ===>            Userid or group name

   USER NAME                 ===>

   DEFAULT GROUP             ===> OMVSGRP      Group name

   PASSWORD                  ===>            User's initial password
                             ===>            Re-enter password to verify

   PASSWORD INTERVAL         ===>            1 - 254 (days), NO, or blank


                        Press ENTER to continue.
```

Figure 88. Add a New User with TSO/E and OMVS Segments (3 of 10)

Specify YES for the optional information to add the TSO/E and OMVS segment information.

```
                          RACF - ADD USER USER1
COMMAND ===>

TO ASSIGN USER ATTRIBUTES, ENTER YES:

   GROUP ACCESS       ===> NO               SPECIAL         ===> NO
   ADSP               ===> NO               OPERATIONS      ===> NO
   OIDCARD            ===> NO               AUDITOR         ===> NO
   NO-PASSWORD        ===> NO

IDENTIFY THE MODEL PROFILE FOR USER DATA SETS (OPTIONAL):

   MODEL PROFILE      ===>

TO CREATE THE FOLLOWING, ENTER YES (OPTIONAL):

   A GENERIC DATA SET PROFILE          ===> NO
   A MINIDISK PROFILE                  ===> NO


TO ADD OPTIONAL INFORMATION, ENTER YES     ===> YES
```

Figure 89. Add a New User with TSO/E and OMVS Segments (4 of 10)

Select the parameters required for this user. In this case, USER1 requires both TSO/E and OMVS segments.

```
                          RACF - ADD USER USER1
 COMMAND ===>


 To ADD the following information, enter any character:

   _ CLASS AUTHORITY
   _ INSTALLATION DATA
   _ GROUP AUTHORITY
   _ SECURITY LEVEL or CATEGORIES
   _ SECURITY LABEL
   _ LOGON RESTRICTIONS
   _ NATIONAL LANGUAGES

   _ DFP PARAMETERS
   S TSO PARAMETERS
   _ OPERPARM PARAMETERS
   _ CICS PARAMETERS
   _ WORK ATTRIBUTES
   S OMVS PARAMETERS
```

Figure 90. Add a New User with TSO/E and OMVS Segments (5 of 10)

Enter all the information necessary according to the requirements of your installation. If the user does not need a TSO/E segment, do not enter any information. Since this user is a TSO user and a OpenEdition MVS user, we have entered TSO related information. The values contained in the next figure are not recommendations, these values are for our own installation. TSO related data is installation dependant.

```
                          RACF - ADD USER USER1
                          TSO-RELATED INFORMATION
COMMAND ===>


ENTER THE FOLLOWING TSO-RELATED INFORMATION:

   JOB CLASS                      ===>
   MESSAGE CLASS                  ===>
   HOLD CLASS                     ===>
   SYSOUT CLASS                   ===>
   ACCOUNT NUMBER                 ===> #ACCNT
   LOGON PROCEDURE NAME           ===> IKJACCNT
   REGION SIZE                    ===> 6072
   UNIT                           ===>
   DESTINATION ID                 ===>
   MAXIMUM REGION SIZE            ===>
   USER DATA                      ===>
   LOGON SECURITY LABEL           ===>
```

Figure 91. Add a New User with TSO/E and OMVS Segments (6 of 10)

This user will be given a UID of 200. The number selected will depend on your installation.

```
                           RACF - ADD USER USER1         1 of 3
                           OMVS PARAMETERS
 COMMAND ===>


 Enter User Identifier (UID) below, then press ENTER:

    USER IDENTIFIER    ===>  200         0 - 2147483647
```

Figure 92. Add a New User with TSO/E and OMVS Segments (7 of 10)

Set this parameter to reflect the initial directory path name for the user, USER1. Remember that the directory must exist for the user to be able to enter the OpenEdition MVS environment. See 5.1.1, “Creating the User File Systems” on page 117 for information on adding user HFS data sets.

```
                          RACF - ADD USER USER1       2 of 3
                          OMVS PARAMETERS
COMMAND ===>


Enter Initial Directory Path Name (HOME), then press ENTER:

  => /u/user1
  =>
  =>
  =>
```

Figure 93. Add a New User with TSO/E and OMVS Segments (8 of 10)

Define the program path name if a shell is installed.

```
                          RACF - ADD USER USER1       3 of 3
                          OMVS PARAMETERS
COMMAND ===>


Enter Program Path Name (PROGRAM), then press ENTER:

  => /bin/sh
  =>
  =>
  =>
  =>
  =>
```

Figure 94. Add a New User with TSO/E and OMVS Segments (9 of 10)

Finally, you will receive the message PROFILE ADDED.

```
                    RACF - USER PROFILE SERVICES              PROFILE ADDED
OPTION ===>

SELECT ONE OF THE FOLLOWING:

   1    ADD          Add a user profile
   2    CHANGE       Change a user profile
   3    DELETE       Delete a user profile
   4    PASSWORD     Change your own password or interval
   5    AUDIT        Monitor user activity (Auditors only)


   8    DISPLAY      Display profile contents
   9    SEARCH       Search the RACF data base for profiles


ENTER THE FOLLOWING INFORMATION:

   USER     ===> USER1       Userid
```

Figure 95. Add a New User with TSO/E and OMVS Segments (10 of 10)

2.6.8 Defining the BPX.SUPERUSER RACF Profile

Optional

This step is optional. Perform this step if you want to allow selected users to use the su (switch user) command to temporarily switch to have privileged user ID authority.

A superuser on UNIX systems (including OpenEdition MVS) is a user that has special rights and privileges. In a previous section, user IDs OMVSKERN and SYSPROG were defined as superusers (UID=0). OMVSKERN was defined as the privileged user ID to run the OMVS kernel address space and SYSPROG was defined as a privileged user ID to run OpenEdition MVS SMP/E tasks.

There are times when a user such as a systems administrator needs to temporarily have superuser authority to perform a special task in OpenEdition MVS. For example, to access another user′s files for recovery purposes. A special RACF facility class profile called BPX.SUPERUSER if defined, allows those users that are permitted to this profile to use the su (switch user) command to temporarily gain superuser authority. This access can be monitored by turning on RACF auditing.

Note: Remember that a superuser can do anything in the system, so like users with the RACF SPECIAL attribute, keep the number of superusers as low as your installation requires.

To define the BPX.SUPERUSER facility class profile:

Run member RACF07 in the OMVS.JCL data set.

OR

Follow the RACF interactive panels for adding a superuser with the BPX.SUPERUSER RACF profile, as shown in the next 10 figures.

In order to create the BPX.SUPERUSER profile, select option 2 of the RACF - Services Option Menu.

```
                        RACF - SERVICES OPTION MENU
OPTION ===>  2

SELECT ONE OF THE FOLLOWING:

   1  DATA SET PROFILES

   2  GENERAL RESOURCE PROFILES

   3  GROUP PROFILES AND USER-TO-GROUP CONNECTIONS

   4  USER PROFILES AND YOUR OWN PASSWORD

   5  SYSTEM OPTIONS

  99  EXIT
```

Figure 96. Add Superusers with the BPX.SUPERUSER Profile (1 of 10)

The next step is to select option 1 of the RACF - General Resource Profile Services Menu.

```
              RACF - GENERAL RESOURCE PROFILE SERVICES
OPTION ===> 1

SELECT ONE OF THE FOLLOWING:

   1   ADD           Add a profile
   2   CHANGE        Change a profile
   3   DELETE        Delete a profile
   4   ACCESS        Maintain access list
   5   AUDIT         Monitor access attempts (Auditors only)


   8   DISPLAY       Display profile contents
   9   SEARCH        Search the RACF data base for profiles
```

Figure 97. Add Superusers with the BPX.SUPERUSER Profile (2 of 10)

Select the class FACILITY and create the RACF profile BPX.SUPERUSER. This is a new RACF profile with the sole purpose of defining OpenEdition MVS superusers.

```
                      RACF - GENERAL RESOURCE SERVICES -  ADD
OPTION ===>

ENTER THE FOLLOWING PROFILE INFORMATION:

   CLASS     ===> FACILITY

   PROFILE   ===> BPX.SUPERUSER


                        <==end of data

   USE A MODEL        ===>        YES or NO


      NOTE: Embedded Blanks are NOT ALLOWED in class or profile names.
```

Figure 98. Add Superusers with the BPX.SUPERUSER Profile (3 of 10)

Change the fields to meet your installation requirements. Specify YES to the optional information. The information that you see on this screen corresponds to our installation.

```
                    RACF - ADD GENERAL RESOURCE PROFILE
COMMAND ===>

  CLASS:         FACILITY
  PROFILE      _ BPX.SUPERUSER


ENTER OR CHANGE THE FOLLOWING INFORMATION:

  OWNER                   ===> SYS1       Userid or group name
  LEVEL                   ===> 0          0-99
  FAILED ACCESSES         ===> FAIL       FAIL or WARN
  UACC                    ===> NONE       NONE, READ, UPDATE,
                                          CONTROL, ALTER or EXECUTE
  AUDIT SUCCESSES         ===> READ       READ, UPDATE, CONTROL,
                                          ALTER or NOAUDIT
  AUDIT FAILURES          ===> READ       READ, UPDATE, CONTROL,
                                          ALTER or NOAUDIT
  NOTIFY                  ===>            Userid

TO ADD OPTIONAL INFORMATION, ENTER YES    ===> YES
```

Figure 99. Add Superusers with the BPX.SUPERUSER Profile (4 of 10)

Specify YES to the Access List to include the OpenEdition MVS user IDs that you want to be able to switch to gain superuser authority.

```
                    RACF - ADD GENERAL RESOURCE PROFILE
COMMAND ===>

  CLASS:         FACILITY
  PROFILE      _ BPX.SUPERUSER


To add optional information, enter YES:

   INSTALLATION DATA               ===>
   APPLICATION DATA                ===>
   ACCESS LIST                     ===> YES
   SECURITY LEVEL or CATEGORIES    ===>
   SECURITY LABEL                  ===>
   MEMBERS                         ===>
   TAPE VOLUME INFORMATION         ===>        TAPEVOL   class only
   TERMINAL INFORMATION            ===>        TERMINAL  class only
   VTAM SESSION                    ===>        APPCLU    class only
   DLF DATA                        ===>        DLFCLASS  class only
   STDATA INFORMATION              ===>        STARTED   class only
```

Figure 100. Add Superusers with the BPX.SUPERUSER Profile (5 of 10)

Select option 1 to add the user ID of the user(s).

```
                 RACF - MAINTAIN GENERAL RESOURCE ACCESS LIST
OPTION ===>  1

  CLASS:         FACILITY
  PROFILE      _ BPX.SUPERUSER


SELECT ONE OF THE FOLLOWING:

   1  ADD      Add users or groups.
               Copy the access list from an existing profile.


   2  REMOVE   Remove specified users and groups from the access list.


   3  RESET    Remove all users and groups from the access list.
```

Figure 101. Add Superusers with the BPX.SUPERUSER Profile (6 of 10)

Enter YES for the SPECIFY option.

```
           RACF - MAINTAIN GENERAL RESOURCE ACCESS LIST - ADD
COMMAND ===>

  CLASS:         FACILITY
  PROFILE      _ BPX.SUPERUSER


ENTER YES FOR EITHER OR BOTH OF THE FOLLOWING:


   COPY         ===>        YES to copy the access list from another
                            profile.


   SPECIFY      ===>  YES   YES to specify the users and groups to be
                            added to the access list.
```

Figure 102. Add Superusers with the BPX.SUPERUSER Profile (7 of 10)

Once the installation has the user IDs of the users that will be defined as superusers, those user IDs will be added to the access list of the BPX.SUPERUSER RACF profile of the FACILITY class with an authority of READ. Remember that the users selected must have a RACF profile defined with an OMVS UID assigned to it and a valid OMVS GID. See 2.6.7, “Define Regular OpenEdition MVS Users” on page 63.

You could add the name of a group, but remember that all the members of the group will have the ability to become superusers. The following shows permitting two existing OpenEdition MVS users, SYSADM1 and SYSADM2 to the BPX.SUPERUSER FACILITY class profile. You can substitute SYSADM1 and SYSADM2 with other existing OpenEdition MVS user IDs.

```
              RACF - MAINTAIN GENERAL RESOURCE ACCESS LIST - ADD
COMMAND ===>

  CLASS:         FACILITY
  PROFILE      _ BPX.SUPERUSER


Enter the access authority to be granted:

   AUTHORITY      ===> READ      NONE, READ, UPDATE,
                                 CONTROL, ALTER or EXECUTE

Enter the users or groups for which entries are to be added:

   ===> SYSADM1  ===> SYSADM2  ===>          ===>
   ===>          ===>          ===>          ===>          ===>
   ===>          ===>          ===>          ===>          ===>
   ===>          ===>          ===>          ===>          ===>
   ===>          ===>          ===>          ===>          ===>

To add these entries to a conditional access list,
   enter YES     ===>
```

Figure 103. Add Superusers with the BPX.SUPERUSER Profile (8 of 10)

After these steps have been done, RACF will send a message as follows, indicating that a refresh of the FACILITY class needs to be performed. Refer to Appendix A, “Refreshing RACF Profiles” on page 195 on how to do a refresh with the RACF interactive panels.

```
ICH10006I RACLISTED PROFILES FOR FACILITY WILL NOT REFLECT THE
          ADDITION(S) UNTIL A SETROPTS REFRESH IS ISSUED.
ICH06011I RACLISTED PROFILES FOR FACILITY WILL NOT REFLECT THE
          UPDATE(S) UNTIL A SETROPTS REFRESH IS ISSUED.
```

Figure 104. Add Superusers with the BPX.SUPERUSER Profile (9 of 10)

After you have finished with the definitions, you will receive the message PROFILE ADDED.

```
              RACF - GENERAL RESOURCE PROFILE SERVICES          PROFILE ADDED
OPTION ===>

SELECT ONE OF THE FOLLOWING:

   1   ADD           Add a profile
   2   CHANGE        Change a profile
   3   DELETE        Delete a profile
   4   ACCESS        Maintain access list
   5   AUDIT         Monitor access attempts (Auditors only)


   8   DISPLAY       Display profile contents
   9   SEARCH        Search the RACF data base for profiles
```

Figure 105. Add Superusers with the BPX.SUPERUSER Profile (10 of 10)

2.6.9 Security Levels for Daemons

A daemon is a long-lived process that runs unattended to perform continuous or periodic system functions. In native MVS, the TCAS or VTAM started tasks could be thought of as daemons. In OpenEdition MVS, as in UNIX systems, daemons run authorized (superuser authority) and can issue authorized functions like setuid and seteuid to change the identity of a users process. In OpenEdition MVS you have the choice of running daemons with regular UNIX security or using RACF controls.

Optional

This step is optional. If you decided to run with UNIX security you don't need to do anything here.

In our environment, we have established an extra level of security with MVS by using the special RACF profile of BPX.DAEMON. This special RACF protection checks to make sure that the daemon program is being executed out of a specific library and only that specific library residing on a particular DASD volume on the system. All this information is stored in a RACF profile covering the daemon program. This protection will prevent an unscrupulous user who might gain access to a superuser's files from adding their own copy of a daemon program in the superuser's files. If this new daemon program is executed unknowingly by a person with superuser authority, it could, if coded correctly, give the unscrupulous user, superuser authority. For more information on the BPX.DAEMON profile, see *MVS/ESA Planning: OpenEdition MVS*.

We have provided a job to define the BPX.DAEMON profile.

Run job RACF08 in the OMVS.JCL data set.

OR

Use the interactive RACF panels as follows:

2.6.9.1 Defining the BPX.DAEMON Profile

In order to create the BPX.DAEMON profile, select option 2 of the RACF - Services Option Menu.

```
                        RACF - SERVICES OPTION MENU
OPTION ===>  2

SELECT ONE OF THE FOLLOWING:

   1  DATA SET PROFILES

   2  GENERAL RESOURCE PROFILES

   3  GROUP PROFILES AND USER-TO-GROUP CONNECTIONS

   4  USER PROFILES AND YOUR OWN PASSWORD

   5  SYSTEM OPTIONS

  99  EXIT
```

Figure 106. Define the BPX.DAEMON Profile (1 of 10)

The next step is to select option 1 of the RACF - General Resource Profile Services Menu.

```
              RACF - GENERAL RESOURCE PROFILE SERVICES
OPTION ===> 1

SELECT ONE OF THE FOLLOWING:

   1   ADD           Add a profile
   2   CHANGE        Change a profile
   3   DELETE        Delete a profile
   4   ACCESS        Maintain access list
   5   AUDIT         Monitor access attempts (Auditors only)


   8   DISPLAY       Display profile contents
   9   SEARCH        Search the RACF data base for profiles
```

Figure 107. Define the BPX.DAEMON Profile (2 of 10)

Select the class, FACILITY and create the RACF profile BPX.DAEMON. This is a new RACF profile.

```
                        RACF - GENERAL RESOURCE SERVICES -  ADD
OPTION ===>

ENTER THE FOLLOWING PROFILE INFORMATION:

   CLASS     ===> FACILITY

   PROFILE   ===> BPX.DAEMON


                         <==end of data

   USE A MODEL        ===>        YES or NO

      NOTE: Embedded Blanks are NOT ALLOWED in class or profile names.
```

Figure 108. Define the BPX.DAEMON Profile (3 of 10)

Change the fields to meet your installation requirements. Specify YES to the optional information. The information that you see on this screen corresponds to our installation.

```
                    RACF - ADD GENERAL RESOURCE PROFILE
 COMMAND ===>

   CLASS:         FACILITY
   PROFILE      _ BPX.DAEMON


ENTER OR CHANGE THE FOLLOWING INFORMATION:

   OWNER                    ===> SYS1       Userid or group name
   LEVEL                    ===> 0          0-99
   FAILED ACCESSES          ===> FAIL       FAIL or WARN
   UACC                     ===> NONE       NONE, READ, UPDATE,
                                            CONTROL, ALTER or EXECUTE
   AUDIT SUCCESSES          ===> NOAUDIT    READ, UPDATE, CONTROL,
                                            ALTER or NOAUDIT
   AUDIT FAILURES           ===> READ       READ, UPDATE, CONTROL,
                                            ALTER or NOAUDIT
   NOTIFY                   ===>            Userid

 TO ADD OPTIONAL INFORMATION, ENTER YES     ===> YES
```

Figure 109. Define the BPX.DAEMON Profile (4 of 10)

Specify YES to the Access List in order to include the user ID of the users the installation would like to be able to start daemon processes.

```
                    RACF - ADD GENERAL RESOURCE PROFILE
 COMMAND ===>

   CLASS:         FACILITY
   PROFILE      _ BPX.DAEMON


 To add optional information, enter YES:

    INSTALLATION DATA               ===>
    APPLICATION DATA                ===>
    ACCESS LIST                     ===> YES
    SECURITY LEVEL or CATEGORIES    ===>
    SECURITY LABEL                  ===>
    MEMBERS                         ===>
    TAPE VOLUME INFORMATION         ===>        TAPEVOL   class only
    TERMINAL INFORMATION            ===>        TERMINAL  class only
    VTAM SESSION                    ===>        APPCLU    class only
    DLF DATA                        ===>        DLFCLASS  class only
    STDATA INFORMATION              ===>        STARTED   class only
```

Figure 110. Define the BPX.DAEMON Profile (5 of 10)

Select option 1 to add the user ID of the user(s).

```
                RACF - MAINTAIN GENERAL RESOURCE ACCESS LIST
OPTION ===>  1

  CLASS:         FACILITY
  PROFILE      _ BPX.DAEMON


SELECT ONE OF THE FOLLOWING:

   1  ADD      Add users or groups.
               Copy the access list from an existing profile.


   2  REMOVE   Remove specified users and groups from the access list.


   3  RESET    Remove all users and groups from the access list.
```

Figure 111. Define the BPX.DAEMON Profile (6 of 10)

Enter YES for the Specify option.

```
           RACF - MAINTAIN GENERAL RESOURCE ACCESS LIST - ADD
COMMAND ===>

  CLASS:         FACILITY
  PROFILE      _ BPX.DAEMON


ENTER YES FOR EITHER OR BOTH OF THE FOLLOWING:


   COPY         ===>        YES to copy the access list from another
                            profile.


   SPECIFY      ===>  YES   YES to specify the users and groups to be
                            added to the access list.
```

Figure 112. Define the BPX.DAEMON Profile (7 of 10)

Once the installation has the user ID of the users that will be defined to the BPX.DAEMON profile of the FACILITY class, those user IDs will be added to the access list with an authority of READ. The users that you are adding to the BPX.DAEMON are those RACF user IDs that are used to start daemons. Remember that the users selected must have a RACF profile defined with an OMVS UID assigned to it and a valid OMVS group.

You could add the name of a group, but remember that all the members of the group will have the ability to use the advantages of the BPX.DAEMON profile.

```
              RACF - MAINTAIN GENERAL RESOURCE ACCESS LIST - ADD
COMMAND ===>

  CLASS:         FACILITY
  PROFILE      _ BPX.DAEMON


Enter the access authority to be granted:

   AUTHORITY      ===> READ      NONE, READ, UPDATE,
                                 CONTROL, ALTER or EXECUTE

Enter the users or groups for which entries are to be added:

   ===> OMVSKERN ===>          ===>          ===>          ===>
   ===>          ===>          ===>          ===>          ===>
   ===>          ===>          ===>          ===>          ===>
   ===>          ===>          ===>          ===>          ===>
   ===>          ===>          ===>          ===>          ===>

To add these entries to a conditional access list,
   enter YES     ===>
```

Figure 113. Define the BPX.DAEMON Profile (8 of 10)

After these steps have been done, RACF will send a message as follows, indicating that a refresh of the FACILITY class needs to be performed.

```
ICH10006I RACLISTED PROFILES FOR FACILITY WILL NOT REFLECT THE
          ADDITION(S) UNTIL A SETROPTS REFRESH IS ISSUED.
ICH06011I RACLISTED PROFILES FOR FACILITY WILL NOT REFLECT THE
          UPDATE(S) UNTIL A SETROPTS REFRESH IS ISSUED.
```

Figure 114. Define the BPX.DAEMON Profile (9 of 10)

After you have finished with the definitions, you will receive the message PROFILE ADDED.

```
              RACF - GENERAL RESOURCE PROFILE SERVICES          PROFILE ADDED
OPTION ===>

SELECT ONE OF THE FOLLOWING:

   1   ADD           Add a profile
   2   CHANGE        Change a profile
   3   DELETE        Delete a profile
   4   ACCESS        Maintain access list
   5   AUDIT         Monitor access attempts (Auditors only)


   8   DISPLAY       Display profile contents
   9   SEARCH        Search the RACF data base for profiles
```

Figure 115. Define the BPX.DAEMON Profile (10 of 10)

Now that the BPX.DAEMON profile is defined, the next figures will show the interactive RACF panels for creating program profiles for daemon programs.

2.6.9.2 Generic RACF Program Protection for IBM Supplied Daemons

Once you have defined the BPX.DAEMON profile, you need to mark all programs that are loaded into an address space that require daemon authority, as RACF program controlled. This can be done by adding a generic program profile of *****, or adding discrete program profiles for programs that the daemon loads into the address space (see 2.6.9.3, “Discrete RACF Program Protection for rlogin and inetd Daemons” on page 86). For IBM supplied daemons, programs will be loaded from the LINKLIB and SCEERUN data sets. Adding a program profile of * will RACF protect all programs that reside in data sets LINKLIB and SCEERUN.

Mandatory

If you defined the BPX.DAEMON profile in 2.6.9, “Security Levels for Daemons” on page 75, you either have to define generic program protection which will cover all IBM supplied daemons or if you are only interested in running the inetd and rlogin daemons, you can define discrete program protection (see 2.6.9.3, “Discrete RACF Program Protection for rlogin and inetd Daemons” on page 86) to cover only those programs that inetd and rlogin call.

To add a RACF program profile of * to cover all IBM supplied daemons:

Run job RACF09 in the OMVS.JCL data set.

OR

Use the interactive RACF panels as follows:

In order to create the profile for load modules as controlled programs, select option 2 of the RACF - Services Option Menu.

```
                          RACF - SERVICES OPTION MENU
OPTION ===>  2

SELECT ONE OF THE FOLLOWING:

   1  DATA SET PROFILES

   2  GENERAL RESOURCE PROFILES

   3  GROUP PROFILES AND USER-TO-GROUP CONNECTIONS

   4  USER PROFILES AND YOUR OWN PASSWORD

   5  SYSTEM OPTIONS

  99  EXIT
```

Figure 116. Create Profile for Controlled Libraries (1 of 7)

The next step is to select option 1 of the RACF - General Resource Profile Services Menu.

```
              RACF - GENERAL RESOURCE PROFILE SERVICES
OPTION ===> 1

SELECT ONE OF THE FOLLOWING:

   1   ADD           Add a profile
   2   CHANGE        Change a profile
   3   DELETE        Delete a profile
   4   ACCESS        Maintain access list
   5   AUDIT         Monitor access attempts (Auditors only)


   8   DISPLAY       Display profile contents
   9   SEARCH        Search the RACF data base for profiles
```

Figure 117. Create Profile for Controlled Libraries (2 of 7)

Select the class PROGRAM and create the RACF profile ***** to cover all programs in the libraries selected later.

```
                      RACF - GENERAL RESOURCE SERVICES -  ADD
OPTION ===>

ENTER THE FOLLOWING PROFILE INFORMATION:

   CLASS     ===> PROGRAM

   PROFILE   ===> *

                        <==end of data

   USE A MODEL        ===>        YES or NO


      NOTE: Embedded Blanks are NOT ALLOWED in class or profile names.
```

Figure 118. Create Profile for Controlled Libraries (3 of 7)

Change the fields to meet your installation requirements. Specify YES to the optional information. The information that you see on this screen corresponds to our installation.

Warning

The following UACC must be READ. Be careful, the default on the panel is UACC(NONE). Do not specify UACC(NONE). A UACC(NONE) profile will lock out certain MVS subsystems from the SYS1.LINKLIB and CEE.V1R5M0.SCEERUN data sets. This level of protection will, in most cases, cause your system to fail during IPL.

```
                    RACF - ADD GENERAL RESOURCE PROFILE
 COMMAND ===>

   CLASS:         FACILITY
   PROFILE      _ *


ENTER OR CHANGE THE FOLLOWING INFORMATION:

   OWNER                   ===> SYS1       Userid or group name
   LEVEL                   ===> 0          0-99
   FAILED ACCESSES         ===> FAIL       FAIL or WARN
   UACC                    ===> READ       NONE, READ, UPDATE,
                                           CONTROL, ALTER or EXECUTE
   AUDIT SUCCESSES         ===> NOAUDIT    READ, UPDATE, CONTROL,
                                           ALTER or NOAUDIT
   AUDIT FAILURES          ===> READ       READ, UPDATE, CONTROL,
                                           ALTER or NOAUDIT
   NOTIFY                  ===>            Userid

 TO ADD OPTIONAL INFORMATION, ENTER YES    ===> YES
```

Figure 119. Create Profile for Controlled Libraries (4 of 7)

Specify YES to the Members option in order to include the libraries with the programs that need to be marked as controlled.

```
                    RACF - ADD GENERAL RESOURCE PROFILE
COMMAND ===>

  CLASS:         FACILITY
  PROFILE      _ *


To add optional information, enter YES:

   INSTALLATION DATA               ===>
   APPLICATION DATA                ===>
   ACCESS LIST                     ===>
   SECURITY LEVEL or CATEGORIES    ===>
   SECURITY LABEL                  ===>
   MEMBERS                         ===> YES
   TAPE VOLUME INFORMATION         ===>        TAPEVOL   class only
   TERMINAL INFORMATION            ===>        TERMINAL  class only
   VTAM SESSION                    ===>        APPCLU    class only
   DLF DATA                        ===>        DLFCLASS  class only
   STDATA INFORMATION              ===>        STARTED   class only
```

Figure 120. Create Profile for Controlled Libraries (5 of 7)

Select option 1 to add the user ID of the user(s).

```
                  RACF - ADD GENERAL RESOURCE GROUP MEMBERS
COMMAND ===>

  CLASS:         PROGRAM
  PROFILE      _ *


ENTER THE MEMBERS TO BE ADDED:
===> 'SYS1.LINKLIB'/OP2RS1/NOPADCHK
===>                      1
===> 'CEE.V1R5M0.SCEERUN'/OP2RS1/NOPADCHK
===>                           2
===>
===>
```

Figure 121. Create Profile for Controlled Libraries (6 of 7)

1 Is the volume where the SYS1.LINKLIB data set resides.

2 Is the volume where the CEE.V1R5M0.SCEERUN data set resides.

After you have finished with the definitions, you will receive the message PROFILE ADDED.

```
              RACF - GENERAL RESOURCE PROFILE SERVICES          PROFILE ADDED
OPTION ===>

SELECT ONE OF THE FOLLOWING:

   1   ADD           Add a profile
   2   CHANGE        Change a profile
   3   DELETE        Delete a profile
   4   ACCESS        Maintain access list
   5   AUDIT         Monitor access attempts (Auditors only)


   8   DISPLAY       Display profile contents
   9   SEARCH        Search the RACF data base for profiles
```

Figure 122. Create Profile for Controlled Libraries (7 of 7)

2.6.9.3 Discrete RACF Program Protection for rlogin and inetd Daemons

Optional

This step is optional only if you want to set up discrete program profiles for the rlogin and inetd daemons instead of setting up generic program profiles as in 2.6.9.2, "Generic RACF Program Protection for IBM Supplied Daemons" on page 81.

Adding a program profile of * in 2.6.9.2, "Generic RACF Program Protection for IBM Supplied Daemons" on page 81, protects all programs that reside in data sets LINKLIB and SCEERUN. This may not be desirable since you are in fact, program protecting all programs in LINKLIB and SCEERUN regardless of use by a daemon program. If you are only running the rlogin and inetd daemons, the specific programs that the inetd and rlogin daemons call have been identified by running the trace described in RACF APAR OW10524. For more complete details on how to identify which programs need to be defined under RACF Program Control, see *RACF Diagnosis Guide*.

To add discrete program profiles for the inetd and rlogin daemons:

Run job RACF10 in the OMVS.JCL data set.

OR

Use the interactive RACF panels as follows:

Select the same panels as in 2.6.9.2, "Generic RACF Program Protection for IBM Supplied Daemons" on page 81 but replace the * with each specific program name below.

```
                        RACF - GENERAL RESOURCE SERVICES -  ADD
OPTION ===>

ENTER THE FOLLOWING PROFILE INFORMATION:

   CLASS     ===> PROGRAM

   PROFILE   ===> BPXBINIT
```

```
                        RACF - ADD GENERAL RESOURCE GROUP MEMBERS
OPTION ===>
  CLASS:          PROGRAM
  PROFILE         BPXBINIT


ENTER THE MEMBERS TO BE ADDED:
===> 'SYS1.LINKLIB'/OP2RS1/NOPADCHK
===>                      1
===>
```

1 Is the volume where the SYS1.LINKLIB data set resides.

Add the following specific program profiles the same way as you added BPXBINIT above:

Program Name	Library
BPXEV003	SYS1.LINKLIB
BPXOLVD	SYS1.LINKLIB
BPXOV	SYS1.LINKLIB
BPXPLPKA	SYS1.LINKLIB
BPXUCSNM	SYS1.LINKLIB
BPXUEYI1	SYS1.LINKLIB
BPXUI1EY	SYS1.LINKLIB
BPXZ24	SYS1.LINKLIB
RLOGIND	SYS1.LINKLIB

Now that the BPX.DAEMON profile is defined, and the programs are marked as controlled, all users will be able to rlogin from another UNIX system directly to the OpenEdition MVS environment, once all the definitions for the network communication are defined. See 7.1.3, “Customizing for inetd and rlogind Daemons” on page 165 for more information on further customization requirements for the inetd and rlogin daemons.

2.7 Allocate the HFS Root File System

OpenEdition MVS provides a POSIX 1003.1-defined file system called the hierarchical file system (HFS) shown in Figure 123 on page 88.

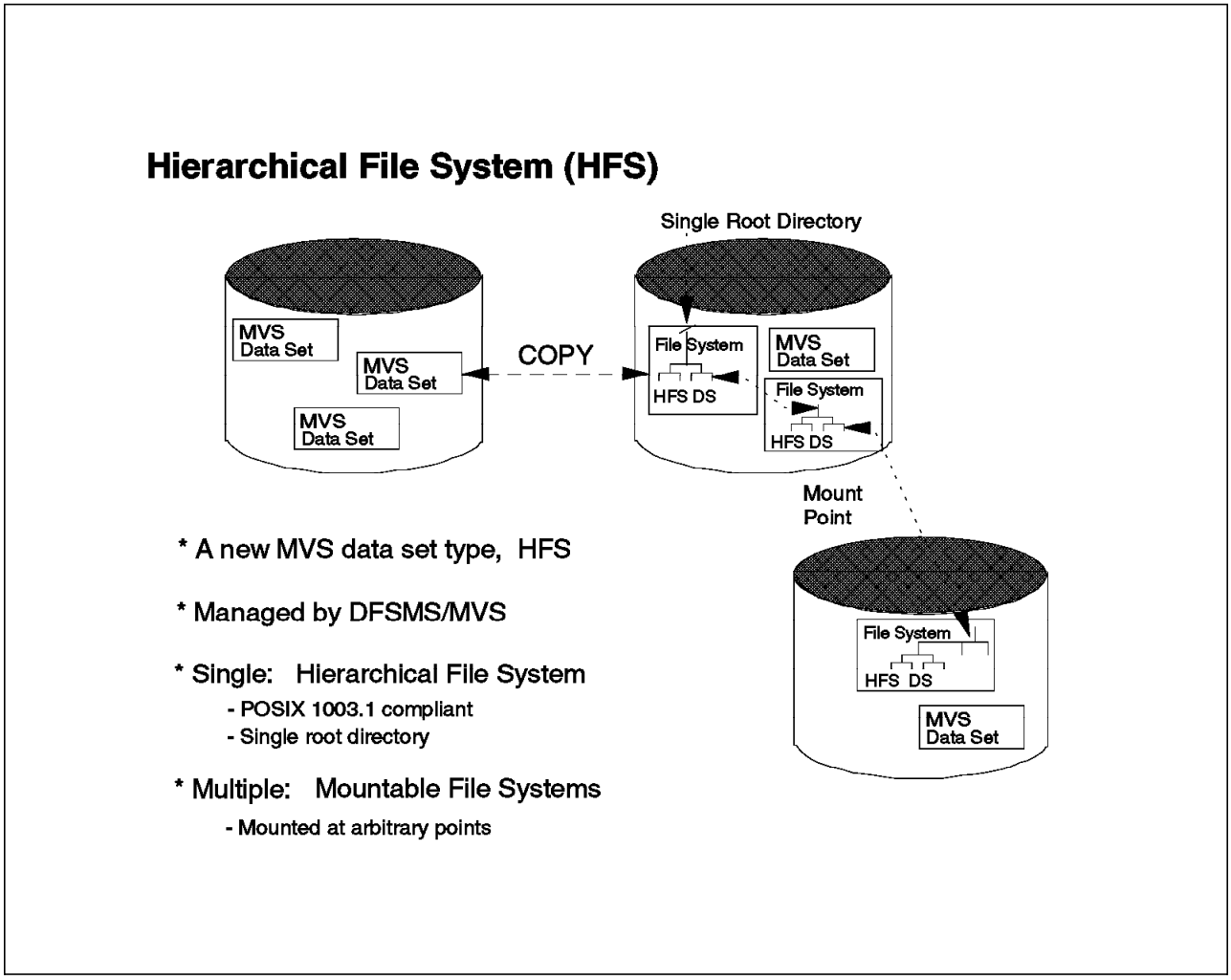


Figure 123. Hierarchical File System (HFS)

The HFS is a set of files contained within a hierarchy of directories, much like the DOS and OS/2 file system. A file within a hierarchical file system directory is called an HFS file. An HFS file is an identifiable, named unit of text or binary information. HFS files are described in directories. A directory is a special file, consisting of the names of files and the characteristics of the files that are described in that directory. A directory can include a file that is itself a directory. Each directory belongs to another directory above it in the file system hierarchy. This facilitates any number of levels in the hierarchical or tree-like structure. The highest level of the hierarchy is the root directory. HFS files can belong directly to the root directory.

HFS file data is byte-oriented, unlike most MVS data sets, which are record-oriented. Input and output (I/O) for HFS files is typically performed through the use of a data stream. Despite the differences between them, you can copy HFS files into MVS data sets and MVS data sets into HFS files using special TSO/E copy commands like OCOPY, OGET and OPUT. You can access HFS files and manipulate file data from application programs using defined C library functions.

MVS treats the file system as a new type of data set, a hierarchical file system data set (HFS data set). Each physical HFS data set is limited to a single DASD volume. DFSMS/MVS facilities are used to manage HFS data sets. MVS views

an entire hierarchical file system as a collection of HFS data sets. Each HFS data set is a mountable file system. The root file system is the first file system mounted when OpenEdition MVS is started. There can be only one root file system. Subsequent file systems can be mounted on a directory within the root file system or on a directory within any mounted file system. HFS data sets can reside on the same physical volume as traditional MVS data sets as long as the volume is SMS managed.

You will need to allocate the HFS root file system at this time.

Sample JCL to allocate the root file system is shown in Figure 124 and the ROOTALLC member in the OMVS.JCL data set.

```
//STEP1    EXEC PGM=IEFBR14
//MKFS     DD DSNAME=OMVS.ROOT,
//            SPACE=(CYL,(100,1,1)),DCB=(DSORG=PO),
//            DSNTYPE=HFS,
//            DISP=(NEW,CATLG,DELETE),
//            STORCLAS=OPENMVS
```

Figure 124. JCL To Allocate the Root File System

Note: If you made your own changes to the sample ACS routines that were shipped with this book (like changing the storage class name), you may need to modify the JCL above.

2.8 Customize the OpenEdition MVS Parmlib Members

Create BPXPRMxx and, if component tracing is required, create CTnBPXxx parmlib members.

The BPXPRMxx parmlib member contains the parameters that control OpenEdition MVS processing and the file system. The system uses these values when initializing OpenEdition MVS. If component tracing is required the CTnBPXxx member has to be added to SYS1.PARMLIB as well.

Step 1 Copy sample member BPXPRM00 from OMVS.PARMLIB into your SYS1.PARMLIB.

This sample will have to be changed to define your ROOT file system data set name if needed. The figure Figure 125 on page 90, shows the changes we have applied to the BPXPRMxx member on our system.

```
MAXFILEPROC(256)                /* Allow at most 256 open files */

MAXPTYS(256)                    /* Allow at most 256 pseudo-TTY
                                   sessions active at once      */


CTRACE(CTIBPX00)                /* Parmlib member 'CTIBPX00' will
                                   contain the initial tracing
                                   options to be used            */

STEPLIBLIST('/system/steplib')  /* HFS file /system/steplib will
                                   contain the list of sanctioned
                                   step libraries for set-user-ID
                                   and set-group-ID executables. */

FILESYSTYPE TYPE(HFS)
            ENTRYPOINT(GFUAINIT)

ROOT     FILESYSTEM('OMVS.ROOT')  /* The ROOT filesystem */
         TYPE(HFS)
         MODE(RDWR)


FILESYSTYPE TYPE(AUTOMNT)          1
            ENTRYPOINT(BPXTAMD)

FILESYSTYPE TYPE(UDS) ENTRYPOINT(BPXTUINT)        2
  NETWORK DOMAINNAME(AF_UNIX)
          DOMAINNUMBER(1)
          MAXSOCKETS(2000)
          TYPE(UDS)
```

Figure 125. Used BPXPRMxx Parmlib Member

1 This controls the automount feature which is explained in 5.1.2.2, “Customizing the Automount Facility” on page 125.

2 These statements control UNIX domain socket support see 7.1, “Customizing and Starting TCP/IP” on page 153 for more information.

Note: The above coded statements in BPXPRMxx are enough for you to start the OpenEdition MVS kernel, and continue with the installation. There are many more parameters that control OpenEdition MVS, see *MVS/ESA Planning: OpenEdition MVS* for details.

Step 2 Copy sample member CTIBPX00 from OMVS.PARMLIB into your SYS1.PARMLIB.

```
 TRACEOPTS
/* ------------------------------------------------------------- */
/*   ON OR OFF: PICK 1                                           */
/* ------------------------------------------------------------- */
/*        ON                                                     */
     OFF
/* ------------------------------------------------------------- */
/*   ASID: 1 TO 16, 2-HEXBYTE VALUES                             */
/* ------------------------------------------------------------- */
/*        ASID(0042,0043,0044) */
/* ------------------------------------------------------------- */
/*   BUFSIZE: A VALUE IN RANGE 30K TO 400K; OR 4M                */
/* ------------------------------------------------------------- */
          BUFSIZE(4M)
/* ------------------------------------------------------------- */
/*   OPTIONS: NAMES OF FUNCTIONS TO BE TRACED, OR "ALL", OR "NONE"  */
/* ------------------------------------------------------------- */
           OPTIONS(
/*                'ALL     '   */
/*               ,'CHARS   '   */
/*               ,'DEVPTY  '   */
/*               ,'FILE    '   */
/*               ,'LOCK    '   */
/*               ,'PIPE    '   */
/*               ,'PROCESS '   */
/*               ,'PTRACE  '   */
/*               ,'SIGNAL  '   */
/*               ,'STORAGE '   */
/*               ,'SYSCALL '   */
```

Figure 126. CTIBPX00 Parmlib Member

Performance will be affected if component tracing is set on during normal operations. Usually, component tracing is only turned on when there is a problem and you need to trace selected components of OpenEdition MVS. It is best to leave it turned off unless instructed to turn it on by IBM Support.

Step 3 Activating VLF

To improve the performance of the OpenEdition MVS RACF support you should add the Virtual Lookaside Facility (VLF) classes that control caching of the OpenEdition MVS UID and GID information. The mapping of OpenEdition MVS group identifiers (GIDs) to RACF group names and the mapping of OpenEdition MVS user identifiers (UIDs) to RACF user IDs is implemented using VLF services to gain performance improvements. It is strongly recommended that VLF be active when running OpenEdition MVS. The new VLF classes IRRUMAP and IRRGMAP must be defined in the COFVLFxx parmlib member before VLF is started. See *RACF System Programmer's Guide* and *MVS/ESA Planning: OpenEdition MVS* for further information on how to use VLF for OpenEdition MVS.

Copy or merge the sample member COFVLF00 from OMVS.PARMLIB into your SYS1.PARMLIB. Figure 127 on page 92 shows the COFVLF00 member:

```
/*Start of specifications *******************************************/
/*                                                                   */
/*  Name: COFVLF00                                                   */
/*                                                                   */
/*  Descriptive Name: Virtual Lookaside Facility (VLF)               */
/*                    default SYS1.PARMLIB member                    */
/*                                                                   */
/*  Copyright =                                                      */
/*        5685-001                                                   */
/*        This macro is "RESTRICTED MATERIALS OF IBM"                */
/*        (C) Copyright IBM Corporation 1988                         */
/*        Licensed materials - Property of IBM                       */
/*                                                                   */
/*  Status: JBB3311                                                  */
/*                                                                   */
/*  Function:                                                        */
/*    COFVLF00 specifies the VLF CLASS and Major Name used for       */
/*    objects stored by Library Lookaside (LLA). The class       @P2C*/
/*    of objects is named "CSVLLA" with a single major name of   @P1C*/
/*    "LLA".                                                         */
/*                                                                   */
/*  Change activity:                                                 */
/*    $L0=VLF     HBB3310 871023 PDAM: Virtual Lookaside Facility    */
/*    $P1=VLF     HBB3310 880209 PDAM: Change LLA class name         */
/*    $P2=PCG00085 JBB3311 880419 PDAM: Library Lookaside name       */
/*                                                                   */
/*   UID and GID information is cached                               */
/*   Changes made to improve MVS OpenEdition performance             */
/*                                                                   */
/*End of Specifications *******************************************/
CLASS NAME(CSVLLA)           /* Class name for Library Lookaside @P2C*/
      EMAJ(LLA)              /* Major name for Library Lookaside @P2C*/
CLASS NAME(IRRUMAP)          /* OpenMVS-RACF UMAP Table     */
      EMAJ(UMAP)             /* Enable caching of OE UIDs   */
CLASS NAME(IRRGMAP)          /* OpenMVS-RACF GMAP Table     */
      EMAJ(GMAP)             /* Enable caching of OE GIDs   */
CLASS NAME(IRRGTS)           /* RACF GTS Table              */
      EMAJ(GTS)              /* Enable caching of RACF GTS  */
CLASS NAME(IRRACEE)          /* RACF saved ACEEs            */
      EMAJ(ACEE)             /* Enable caching of RACF ACEE */
```

Figure 127. Used COFVLF00 Parmlib Member

The VLF member can be activated using the operator command:

```
START VLF,SUB=MSTR,NN=xx
```

where xx is the suffix of the COFVLF member of parmlib.

The start of VLF can be added to the COMMNDxx member of parmlib to be started at each system IPL.

2.9 IPL the System to Activate OpenEdition MVS System Services

Warning: The OpenEdition MVS System Services component installs elements in SYS1.LPALIB, so you will need to IPL this new system with a CLPA before continuing. After the IPL, follow the steps in the next section to customize the OMVS procedure.

2.10 Starting OMVS

Build a procedure to run OpenEdition MVS.

Step 1 Copy sample OMVS procedure into a proclib data set.

Figure 128 is a copy of the OMVS cataloged procedure that is used to start the OpenEdition MVS kernel address space. You can get a copy of this procedure in member OMVS in the OMVS.PROCLIB data set found on the diskette. Change the highlighted parameter to point to the appropriate BPXPRMxx member in your SYS1.PARMLIB.

```
//*************************************************************/
//*                                                           */
//*     $MAC(OMVS) COMP(SCPX1) PROD(BPX):                     */
//*                                                           */
//* OPENMVS STARTUP PROC                                      */
//*                                                           */
//* TO START THE OPENMVS SERVER ADDRESS SPACE,                */
//* ISSUE CMD                                                 */
//*                                                           */
//*  S OMVS,OMVS=XX                                           */
//*                                                           */
//* WHERE XX IS THE SUFFIX ON A SYS1.PARMLIB MEMBER NAMED     */
//* BPXPRMXX.                                                 */
//*                                                           */
//* SEE SYS1.SAMPLIB(BPXPRMXX) FOR A SAMPLE OF OPTIONS        */
//*                                                           */
//* CHANGE-ACTIVITY:                                          */
//*                                                           */
//*  $00=OW06506  HOM1130 940912 PDJI: FREE IEFPARM           */
//*************************************************************/
//OMVS PROC     OMVS=xx                 <===CHANGE
//OMVS EXEC     PGM=BPXINIT,PARM='&OMVS',REGION=0K
//IEFPARM DD    DSN=SYS1.PARMLIB,FREE=CLOSE,DISP=SHR
```

Figure 128. OMVS Cataloged Procedure

Step 2 Start OpenEdition MVS

If you do not specify OMVS=xx, the system will use member, BPXPRM00 as a default or whichever suffix you specified in the OMVS cataloged procedure. In order to start the OpenEdition MVS environment issue the following operator command:

```
START OMVS,OMVS=xx
```

where xx is the suffix of the BPXPRM member of parmlib.

You should get messages similar to the following:

```
IEF403I OMVS - STARTED - TIME=12.17.44
BPXF013I FILE SYSTEM OMVS.ROOT 344
WAS SUCCESSFULLY MOUNTED.
BPXI004I OMVS INITIALIZATION COMPLETE
```

2.11 Building the Directory Structure in the New ROOT HFS

Run the REXX exec, BPXISMKD from SYS1.SAMPLIB to build the directory structure. You will need to change these values in the exec to match the settings of these parameters in your BPXPRMxx member of SYS1.PARMLIB. Based on the sample BPXPRM00 member that is shipped with this book they should reflect:

```
$root = "/"
MAXFILEPROC = '256'             /* MAXFILEPROC                    */
MAXPTYS = '256'
```

Any errors will be printed to the screen as well as to an error log data set called:

```
youruserID.BPXISMKD.FAERROR
```

If you use a production system, you need to make a copy of the root file system and mount the copy at a new mountpoint. See 6.2.1, “Installing Service to Products in the HFS” on page 148 and *MVS/ESA Planning: OpenEdition MVS* for more information on installing service and cloning procedures. You will need to run the BPXISMKD exec against this new copy.

2.12 Using the OpenMVS ISPF Shell

The OpenMVS ISPF Shell or ISHELL is an ISPF panel interface that you can use instead of TSO/E commands or shell commands to perform tasks against OpenEdition user IDs and the HFS file system. If you are more comfortable using the ISPF editor and ISPF pull-down menus, the ISHELL is the tool for you.

Note: You must have an OMVS segment in your RACF profile to use the ISHELL.

With the ISHELL an OpenEdition MVS user with superuser authority and/or with the RACF SPECIAL attribute can:

- Mount a file system
- Unmount a file system
- Change attributes for an OpenEdition MVS user
- Display a list of users and sort them by name, UID, and GID
- Print a list of users
- Setup OpenEdition MVS users
- Setup OpenEdition MVS groups
- Create character special files

With the ISHELL a general OpenEdition MVS user (a user without superuser authority and without any special RACF attributes) can:

- Create a directory
- List directory trees
- Create a file
- Copy files
- Edit and browse files using the ISPF editor
- Create symbolic links or external links
- Create a hard link to a file

Step 1 Add the following OpenEdition MVS data sets to the data sets allocated to your TSO/E ID:

```
//ISPPLIB   DD DSN=ISP.V4R1MO.ISPPENU,DISP=SHR
//          DD DSN=SYS1.SBPXPENU,DISP=SHR
//ISPMLIB   DD DSN=ISP.V4R1MO.ISPMENU,DISP=SHR
//          DD DSN=SYS1.SBPXMENU,DISP=SHR
//ISPTLIB   DD DSN=ISP.V4R1MO.ISPTENU,DISP=SHR
//          DD DSN=SYS1.SBPXTENU,DISP=SHR
//SYSPROC   DD DSN=ISP.V4R1MO.ISPCLIB,DISP=SHR
//          DD DSN=SYS1.SBPXEXEC,DISP=SHR
```

Figure 129. ISHELL Panels Setup

Step 2 From ISPF option 6, type in: ISHELL

```
  File  Directory  Special_file  Tools  File_systems  Options  Setup  Help
------------------------------------------------------------------------
                           OpenMVS ISPF Shell

Enter a pathname and do one of these:

    - Press Enter.
    - Select an action bar choice.
    - Specify an action code or command on the command line.

Return to this panel to work with a different pathname.
                                                                More: +
   /
   ___________________________________________________________________
   ___________________________________________________________________
   ___________________________________________________________________


(C) Copyright IBM Corp., 1993.  All rights reserved.
Command ===> _____________________________________________________________
 F1=Help      F3=Exit      F5=Retrieve  F6=Keyshelp  F7=Backward  F8=Forward
F10=Actions  F11=Command  F12=Cancel
```

Figure 130. ISHELL Main Panel

Step 3 If you press Enter, a list of the files that exist in the ROOT directory will be displayed.

```
                              Directory List

/
Select one or more files with / or action codes.

  Type  Filename                                                Row 1 of 10
_ Dir   .
_ Dir   ..
_ Dir   bin
_ Dir   dev
_ Dir   etc
_ Dir   lib
_ Dir   samples
_ Dir   tmp
_ Dir   u
_ Dir   usr

Command ===> ____________________________________________________________
 F1=Help      F3=Exit      F4=Name      F5=Retrieve  F6=Keyshelp  F7=Backward
 F8=Forward  F11=Command  F12=Cancel
```

Figure 131. ISHELL Directory List

From here you can list, browse, edit, copy and delete files and directories in the HFS file system. The ISHELL will become more valuable as an administration tool after you install the Shell and Utilities and as more and more people start to use OpenEdition MVS.

2.13 Controlling Access to the Root Directory

When the root directory is initially allocated, the default permission bits are set to 700, which will give read, write and execute access to only the system programmer who created the HFS data set mounted at the root. You will need to change the permission bits to allow other users on the system to read and execute files from the root.

Step 1 From ISPF option 6, type in: ISHELL. Bring the cursor up to the Directory pull-down and press Enter.

```
File Directory Special_file  Tools  File_systems Options Setup  Help
-------------------------------------------------------------------------
                           OpenMVS ISPF Shell

Enter a pathname and do one of these:

    - Press Enter.
    - Select an action bar choice.
    - Specify an action code or command on the command line.

Return to this panel to work with a different pathname.
                                                                More: +
   /
   ___________________________________________________________________
   ___________________________________________________________________
   ___________________________________________________________________


(C) Copyright IBM Corp., 1993.  All rights reserved.
Command ===> ____________________________________________________________
 F1=Help      F3=Exit      F5=Retrieve  F6=Keyshelp  F7=Backward  F8=Forward
F10=Actions  F11=Command  F12=Cancel
```

Figure 132. ISHELL Main Panel

Step 2 Select Attributes, option 3, and press Enter.

```
  File  Directory  Special_file  Tools  File_systems  Options  Setup  Help
-------------------------------------------------------------------------

     ____________________________
     | 3  1. List directory(L)...  |  Shell
     |    2. New(N)...             |
Enter|    3. Attributes(A)...      |
     |    4. Delete(D)...          |
    -|    5. Rename(R)...          |
    -|    6. Copy to PDS(C)...     |
    -|    7. Copy from PDS(I)...   | on the command line.
     |    8. Print(P)              |
Retur|    9. Compare(M)...         | erent pathname.
     |   10. Find strings(F)...    |                          More: +
   / |   11. Set working directory(W) |
   __-------------------------------- ___________________________________
   ___________________________________________________________________
   ___________________________________________________________________


(C) Copyright IBM Corp., 1993.  All rights reserved.
Command ===> ____________________________________________________________
 F1=Help      F3=Exit      F5=Retrieve  F6=Keyshelp  F7=Backward  F8=Forward
F10=Actions  F11=Command  F12=Cancel
```

Figure 133. ISHELL Directory List

Step 3 Select the Edit pull-down and press Enter.

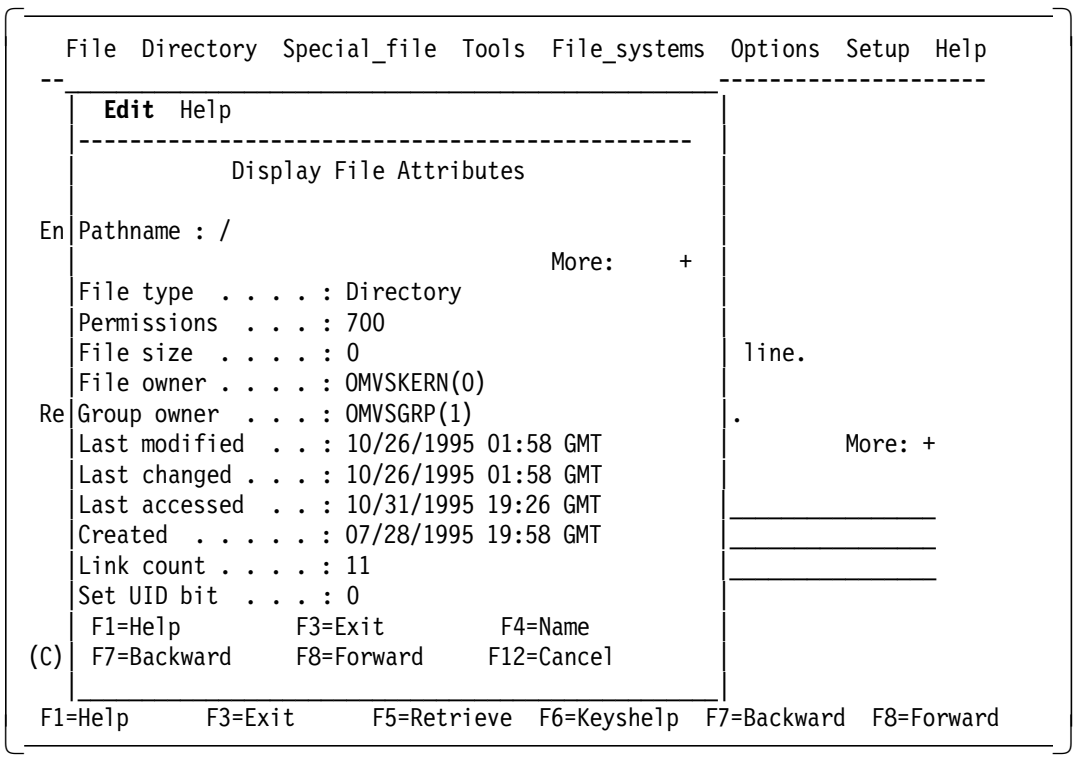


Figure 134. Edit File Attributes

Step 4 Select option 1.

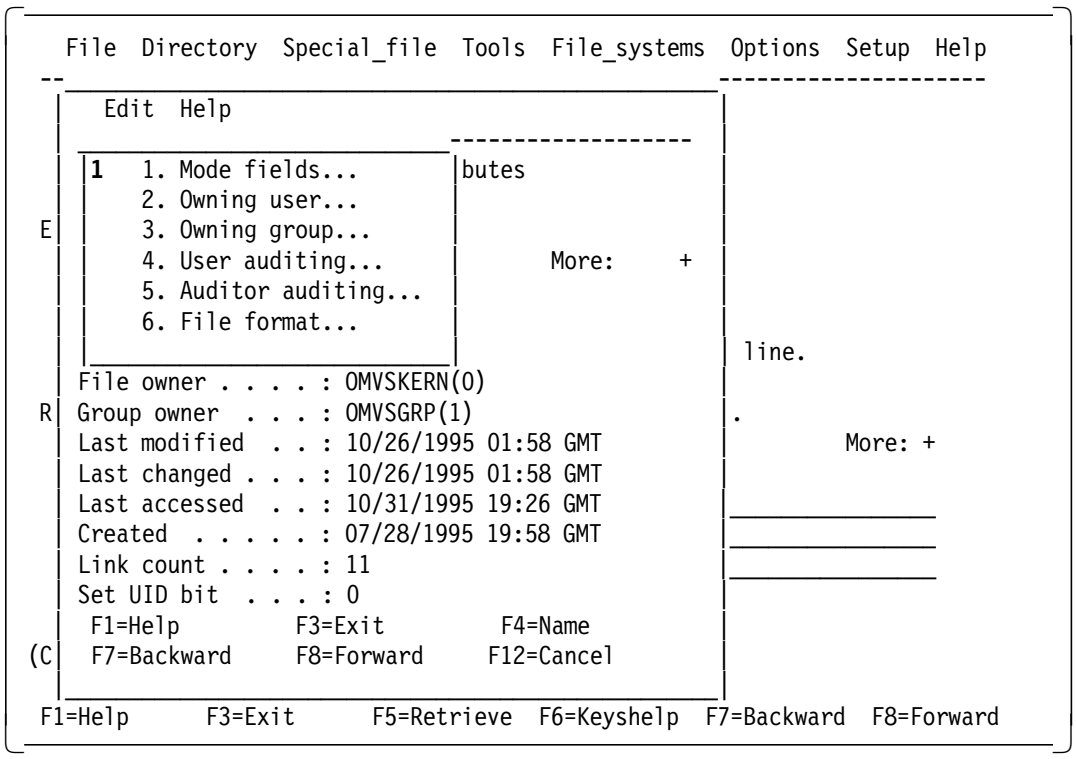


Figure 135. Change File Mode

Step 5 Change the permissions to 755.

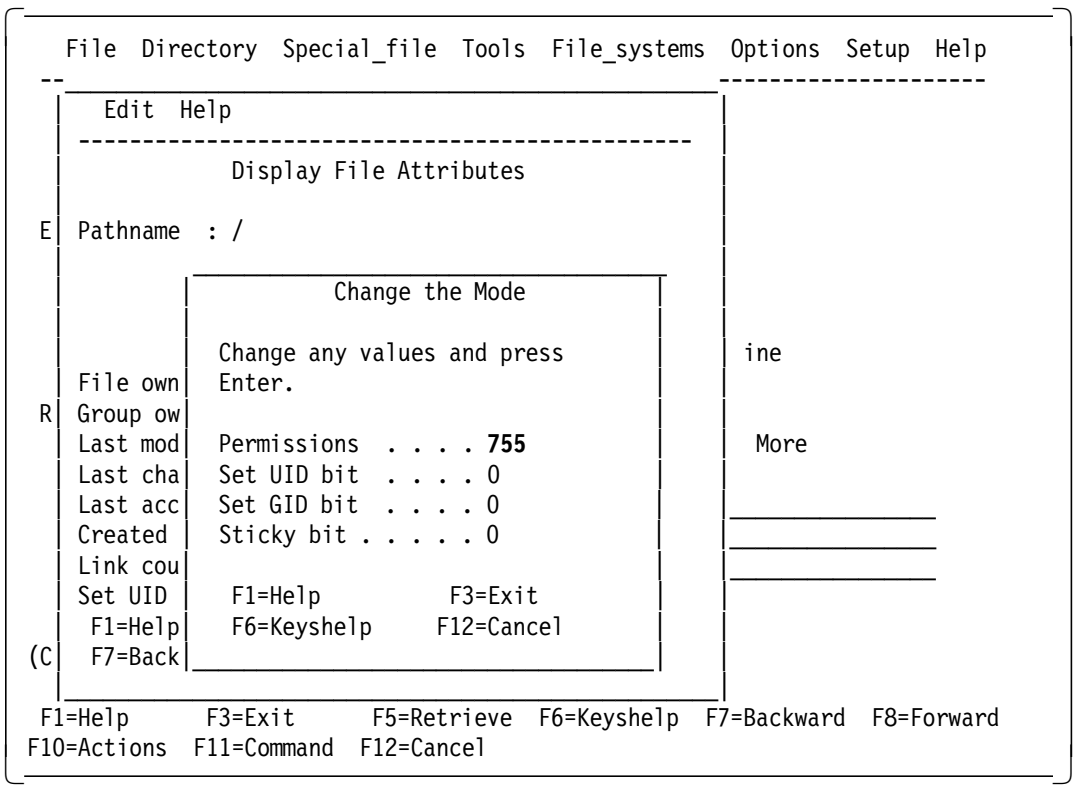


Figure 136. Change Permission Bits

The root file system now has its permission bits set to 755. This is the correct setting to enable general users to access the root file system with read and execute modes.

2.14 SMP/E Install the Language Environment Component

OpenEdition MVS services can be used from C/C++ functions and assembler calls. Applications written in COBOL or PL/I can use OpenEdition MVS services indirectly through C/C++ functions or assembler calls.

The new functions introduced with OpenEdition MVS require one of the following run time environments:

- Language Environment for MVS and VM, Program Number 5688-198 (formerly LE/370).

  OR

- C/C++ Language Support Feature of MVS/ESA SP Version 5, Program Number 5655-068 (JES2) or 5655-069 (JES3).

The C/C++ Language Support Feature of MVS/ESA SP Version 5 is a no-price feature of MVS/ESA SP V5. It only contains the Language Environment base, C/C++ components in mixed-case U.S. English and Japanese, and program publications. If you are just interested in running C programs then this no-price feature is all you need.

Language Environment for MVS and VM contains everything that is in C/C++ Language Support Feature of MVS/ESA SP Version 5 plus additional COBOL, PL/I and Fortran components.

The C/C++ OpenEdition HFS component (FMID JMWL55H) for both C/C++ Language Support Feature of MVS/ESA SP Version 5 and Language Environment for MVS and VM contains elements that install into the HFS. For example, the C/C++ header files usually installed in CEE.V1R5M0.SCEEH.H are also installed into the HFS directory /usr/include/IBM with hardlinks to /usr/include alias names. The OpenEdition MVS support can only be installed, if the OMVS address space is currently active. Installation, service and customization of the OpenEdition MVS support must be performed from either a OpenEdition MVS superuser or the owner of the target HFS directories. The user ID must also have a home directory of ('/') and a program name of ('/bin/sh') assigned.

To install the Language Environment component:

Step 1 Refer to *IBM Language Environment for MVS and VM Installation and Customization on MVS* when you are ready to install Language Environment for MVS and VM or the C/C++ Language Support Feature of MVS/ESA SP Version 5. Install all the FMIDs necessary, according to your installation's requirements.

Once you have finished the installation, it is recommended to place the runtime library in the LNKLSTxx member of SYS1.PARMLIB.

Step 2 Run the Installation Verification Procedures (IVP) for the components you installed.

Note: You cannot run IVP job CEEWIVP8 at this time because it requires the OpenEdition MVS Shell and Utilities feature to be installed first. Come back and run this job later after you have installed the OpenEdition MVS Shell and Utilities feature.

2.15 SMP/E Install a C Compiler

Step 1 If you plan to write applications in the C/C++ high-level language (HLL) you will need to install and customize IBM C/C++ for MVS/ESA Version 3 Release 1, Program Number 5655-121. See *C/C++ MVS User's Guide* for more information. If you are only interested in C HLL support, it is provided by IBM AD/Cycle C/370 Version 1 Release 2, Program Number 5688-216. See *C/370 User's Guide* for more information. Install the FMIDs necessary according to these books.

Once you have finished the installation, it is recommended to place the run-time library in the LNKLSTxx member of SYS1.PARMLIB.

Step 2 Run the Installation Verification Procedures (IVP) for the components you installed.

Chapter 3. Installation of the OpenEdition MVS Components that Require the HFS

The OpenEdition MVS components must be installed in a particular order. The first component, OpenEdition MVS System Services contains the kernel code. This component was installed in Chapter 2, "Beginning the Installation" on page 3. The OpenEdition MVS Application Services component contains the code to support full-screen applications, remote login and automount support. The last two components, the OpenEdition MVS Shell and Utilities and OpenEdition MVS Debugger are optional components and should be installed last. The order of installation is important. You need to have the OMVS kernel started and running before you can SMP/E install the OpenEdition MVS Application Services, OpenEdition MVS Shell and Utilities and OpenEdition MVS Debugger features. These three components install their elements into the root HFS data set. All access to the root file system goes through the OpenEdition MVS kernel address space. Table 6 contains all the OpenEdition MVS FMIDs that need the OpenEdition MVS kernel to be started before they can be installed.

Table 6. List of OpenEdition MVS Components That Need the OpenEdition MVS Kernel For Installation

FMID	Component Name
HOT1130	OpenEdition MVS Application Services
JOT13N0	OpenEdition MVS Application Services NLS US English
JOT13J0	OpenEdition MVS Application Services NLS Japanese
HSU1130	OpenEdition MVS Shell and Utilities
JSU13N0	OpenEdition MVS Shell and Utilities NLS US English
JSU13J0	OpenEdition MVS Shell and Utilities NLS Japanese
HDX1130	OpenEdition MVS Debugger

The instructions on the next few pages will assist you in installing these OpenEdition MVS components.

3.1 SMP/E Install the OpenEdition MVS Features

Use System Modification Program Extended (SMP/E) to install the rest of the OpenEdition MVS features. SMP/E requires that the OpenEdition MVS kernel be started in order to access directories in the HFS.

Step 1 The *Program Directory for MVS/ESA SP 5.2.2* has specific information on the SMP/E tasks required to install OpenEdition MVS Application Services, FMIDs HOT1130, JOT13N0, and optionally, the Japanese feature FMID, JOT13J0.

Run the FOMISCPY job in SYS1.SAMPLIB if you didn't point to the same SCEELKED and SCEERUN libraries for C/C++ Language Support Feature of MVS/ESA SP Version 5 and Language Environment for MVS and VM.

The note in the *Program Directory for MVS/ESA SP 5.2.2* about cleaning up old HFS directories if you are installing on top of an older release of OpenEdition MVS in incorrect. Do not delete

the following dddefs and directories until after the SMP/E APPLY of the Shell and Utilities feature, HSU1130:

```
SFSUMETC          '/etc/IBM/'
SFSUMSAM          '/etc/samples/IBM/'
```

SMP/E needs those directories to be there for it to delete old FMIDs, HSU1120 and HSU1110.

See the note in the program directory about running job FOMISCHD in SYS1.SAMPLIB to change the permission bits of those files that were installed with OpenEdition MVS Application Services if you are using SMP/E 1.8. People using SMP/E 1.8.1 will not have to run this job because the permission bits are changed during SMP/E APPLY processing.

Step 2 The *Program Directory for OpenEdition MVS Shell and Utilities* has specific information on the SMP/E tasks required to install OpenEdition MVS Shell and Utilities, FMIDs, HSU1130, JSU13N0, and optionally, the Japanese feature FMID, JSU13J0.

See the note in the program directory about running job FSUMUSCH in SYS1.SAMPLIB to change the permission bits of those files that were installed with OpenEdition MVS Shell and Utilities if you are using SMP/E 1.8. People using SMP/E 1.8.1 will not have to run this job because the permission bits are changed during SMP/E APPLY processing.

Step 3 The *Program Directory for the OpenEdition MVS Debugger* has specific information on the SMP/E tasks required to install the OpenEdition MVS Debugger, FMID, HDX1130.

See the note in the program directory about running job FDBXUSDD in SYS1.SAMPLIB to change the permission bits of those files that were installed with the OpenEdition MVS Debugger if you are using SMP/E 1.8. People using SMP/E 1.8.1 will not have to run this job because the permission bits are changed during SMP/E APPLY processing.

3.2 Customize the Shell and Utilities

If you installed the Shell and Utilities feature of OpenEdition MVS it will need to be customized before using it. The Shell and Utilities feature ships a default initialization program in /usr/sbin/init. This program is run every time the OMVS kernel is started. Before /usr/sbin/init invokes the shell it reads a file called /etc/init.options which contains options such as what shell program to execute at startup and the name of the initial shell script that is executed. The init.options file, an initial shell script file called /etc/rc and a file called /etc/profile all need to be customized at this time.

Note: If you are installing OpenEdition MVS on top of an older release of OpenEdition MVS you may have old copies of the /etc/init.options, /etc/rc and /etc/profile files. You will need to check to make sure you will not be overlaying any of your own changes in these files before you replace them from the /samples directory.

Step 1 In MVS/ESA V5.1 of OpenEdition MVS the initialization program was shipped in /etc/init. Now in MVS/ESA V5.2.2 it is shipped in /usr/sbin/init. If you are installing OpenEdition MVS on top of a MVS/ESA V5.1 system you need to go in and delete the /etc/init program by hand (using the ISHELL). The order of search for the shell when looking for the initialization program is /etc/init first then /usr/sbin/init. If you don't delete the old /etc/init program by hand, then it will be incorrectly executed. This procedure will be documented in the PSP bucket for MVS/ESA V5.2.2 in the near future.

Step 2 Copy the init.options file from the /samples directory to the /etc directory. We will use the ISHELL to perform the copy and customization. The ISHELL is discussed in 2.12, "Using the OpenMVS ISPF Shell" on page 94. Invoke the ISHELL and type in the /samples directory and press Enter.

```
  File  Directory  Special_file  Tools  File_systems  Options  Setup  Help
-------------------------------------------------------------------------
                           OpenMVS ISPF Shell

Enter a pathname and do one of these:

    - Press Enter.
    - Select an action bar choice.
    - Specify an action code or command on the command line.

Return to this panel to work with a different pathname.
                                                              More: +
   /samples
   ___________________________________________________________________
   ___________________________________________________________________
   ___________________________________________________________________


(C) Copyright IBM Corp., 1993.  All rights reserved.
Command ===> ___________________________________________________________
 F1=Help      F3=Exit      F5=Retrieve  F6=Keyshelp  F7=Backward  F8=Forward
F10=Actions  F11=Command  F12=Cancel
```

Figure 137. Build the init.options File (1 of 5)

Step 3 A list of the files that exist in the /samples directory will be displayed. Type in the c (copy) action code next to the init.options file and press Enter.

```
                              Directory List

/samples
Select one or more files with / or action codes.

  Type  Filename                                                Row 15 of 34
_ File  hfsinfo.c
_ File  hfsinfo.o
_ File  hobbies
_ Dir   IBM
_ File  inetd.conf
c File  init.options
_ File  magic
_ File  mailx.rc
_ File  mountx
_ File  ohelp.ENU
_ File  profile
_ File  rc
_ File  startup.mk
_ File  system
_ File  unmountx
Command ===> ____________________________________________________________
 F1=Help      F3=Exit      F4=Name      F5=Retrieve  F6=Keyshelp  F7=Backward
 F8=Forward  F11=Command  F12=Cancel
```

Figure 138. Build the init.options File (2 of 5)

Step 4 Select option 1 as the destination of the copy operation and press Enter.

```
                              Directory List

/ ┌──────────────────────────── Copy from a File ──────────────────────┐
S │                                                                    │
  │ Copying from file:                                                 │
  │ /samples/init.options                                              │      of 34
_ │                                                                    │
_ │ Destination for copy:                                              │
_ │ 1_  1.  File...                                                    │
_ │     2.  Data set...                                                │
_ │                                                                    │
c │ Select additional options for data set copy:                       │
_ │ _  Binary copy                                                     │
_ │ _  Conversion...                                                   │
_ │                                                                    │
_ │                                                                    │
_ │                                                                    │
_ │  F1=Help      F3=Exit      F6=Keyshelp F12=Cancel                  │
_ └────────────────────────────────────────────────────────────────────┘
_ File  system
_ File  unmountx
Command ===> ____________________________________________________________
 F1=Help      F3=Exit      F4=Name      F5=Retrieve  F6=Keyshelp  F7=Backward
 F8=Forward  F11=Command  F12=Cancel
```

Figure 139. Build the init.options File (3 of 5)

Step 5 We will be copying /samples/init.options to /etc/init.options. Press Enter after typing in the target file name.

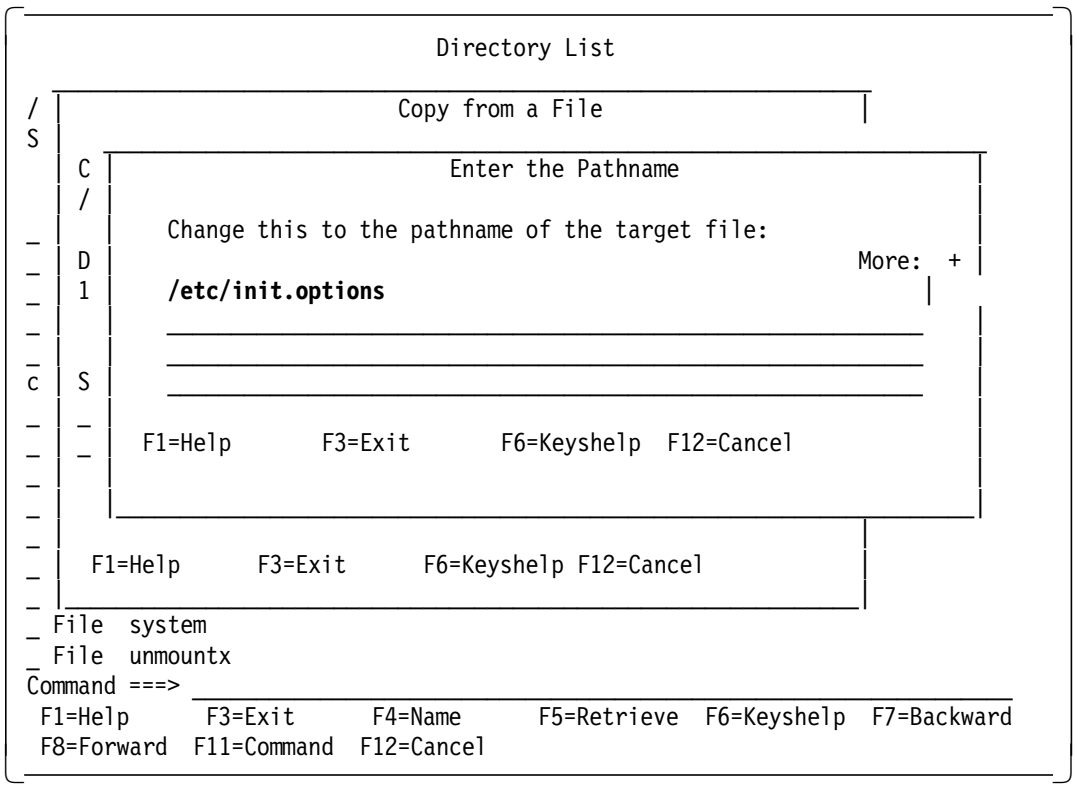


Figure 140. Build the init.options File (4 of 5)

Step 6 /samples/init.options will be copied to /etc/init.options.

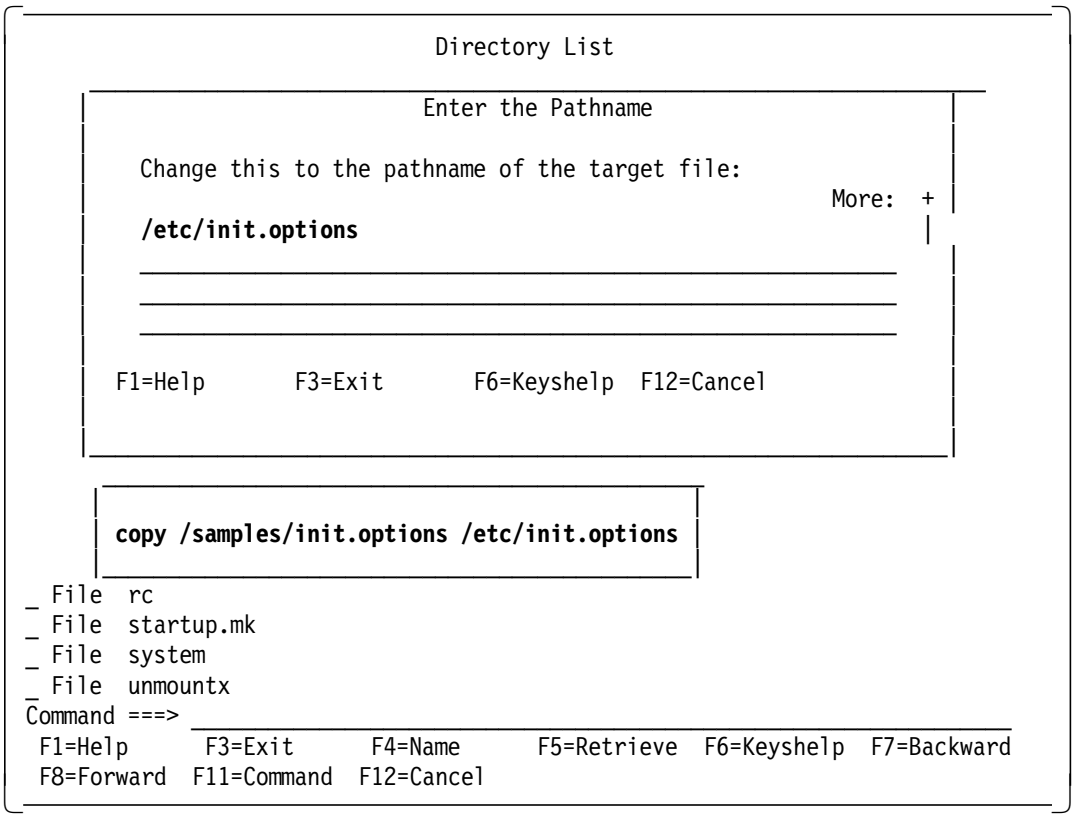


Figure 141. Build the init.options File (5 of 5)

Step 7 Copy the /samples/rc file to /etc/rc

Use the same process as shown above to copy the /samples/rc file to /etc/rc. The /etc/rc file contains customization commands for the Shell and Utilities that are executed at startup.

Step 8 Copy the /samples/profile file to /etc/profile

Use the same process as shown above to copy the /samples/profile file to /etc/profile. The /etc/profile file contains the initial setting of environment variables. OpenEdition MVS users can override these values in their own $HOME/.profile file, see 5.1.3, “Customizing $HOME/.profile” on page 131 for more information on the $HOME/.profile.

Note: Make sure you copy /samples/profile not /samples/.profile (notice the dot). The .profile file is the sample file that individual users can copy into their $HOME/.profile file, see 5.1.3, “Customizing $HOME/.profile” on page 131 for more information on the $HOME/.profile.

Step 9 Edit /etc/init.options

Use the ISHELL to edit the newly copied /etc/init.options if required. Figure 142 on page 107 contains the uncustomized /etc/init.options file.

```
-a  120                              timeout = 120 seconds
-t  1                                terminate shell = yes
-sc /etc/rc                          shell script = /etc/rc
-e  TZ=EST5EDT                       TZ environment variable
*e  LANG=USA                         LANG environment variable
*e  NLSPATH=/usr/lib/nls/msg/%L/%N   NLSPATH environment variable
*sh /bin/sh                          shell = /bin/sh
*e  PATH=/bin                        PATH environment variable
*e  SHELL=/bin/sh                    SHELL environment variable
*e  LOGNAME=ROOT                     LOGNAME environment variable
```

Figure 142. Example of an Uncustomized /etc/init.options File

You may need to change the following -e environment variable options depending on where you are located in the world:

-e TZ (timezone)

-e LANG (language)

-e NLSPATH (locale)

See Appendix B, “Setting the TZ Environment Variable” on page 199 for an explanation of the TZ variable and some examples. Alternatively, you can specify the TZ, NLSPATH and LANG operands in /etc/profile. Also see *MVS/ESA Planning: OpenEdition MVS*.

Step 10 Edit /etc/rc

The supplied sample /etc/rc does not need any further customization unless you want to add some additional shell commands or start an installation supplied daemon. The start up of the inet daemon is commented out below. Activation of the inet daemon is discussed in 7.1.3, “Customizing for inetd and rlogind Daemons” on page 165. Activation of the automount feature can also be called out from /etc/rc, see 5.1.2.2, “Customizing the Automount Facility” on page 125 for more information.

```
# Initialization shell script, pathname = /etc/rc

# Initial setup for OpenEdition MVS
export _BPX_JOBNAME='ETCRC'

# Setup utmpx file
>/etc/utmpx
chmod 644 /etc/utmpx

# Reset all slave tty files
chmod 666 /dev/tty*
chown 0 /dev/tty*
# Setup write, talk, mesg utilities
chgrp TTY   /bin/write
chgrp TTY   /bin/mesg
chgrp TTY   /bin/talk
chmod 2755  /bin/write
chmod 2755  /bin/mesg
chmod 2755  /bin/talk
# Invoke vi recovery
mkdir -m 777 /etc/recover
/usr/lib/exrecover
# Start the INET daemon for remote login activity
#_BPX_JOBNAME='INETD' usr/sbin/inetd /etc/inetd.conf
echo /etc/rc script executed, date
```

Figure 143. Contents of the /etc/rc File

Step 11 Edit /etc/profile

The environment variables and commands used by most Shell and Utilities users are placed in the /etc/profile file.

To customize the profile, copy the IBM-supplied sample provided in the /sample directory and make the appropriate changes. Figure 144 on page 109 and Figure 145 on page 110 shows our settings. For performance reasons do not define search libraries using the STEPLIB variable.

Note: A # in the /etc/profile file is a comment.

```
if Y -z "$STEPLIB" " && tty -s;
then
    echo " - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - "
    echo " - Improve performance by preventing the propagation - "
    echo " - of TSO/E or ISPF STEPLIBs                         - "
    echo " - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - "
    export STEPLIB=none
    exec sh -L

fi

# Set the time zone as appropriate.
TZ=EST5EDT
# This sets a default command path, including your current working
# directory (CWD).
PATH=/bin:.

# Sets the path for NLS files (message catalogs).
NLSPATH=/usr/lib/nls/msg/%L/%N

# Sets the language
# POSIX locale: LANG=C, For Japanese: LANG=Ja_JP
LANG=C

# Sets the name of the system mail file and enables mail notification.
MAIL=/usr/mail/$LOGNAME

# Export the values so the system will have access to them.
export TZ PATH NLSPATH MAIL LANG

# Set the default file creation mask - reference umask in the
# OpenEdition MVS Commands Reference
umask 022
# Set the LOGNAME variable readonly so it is not accidentally modified.
readonly LOGNAME
```

Figure 144. Contents of the /etc/profile File (1 of 2)

```
# ========================================
#  c89/cc customization section
# ========================================
# Version of programs and runtime library used by c89:
# ====================================================
#
#  Compiler:
#  ---------
#  export _C89_CVERSION="0x13010000"    1
#  export  _CC_CVERSION="0x13010000"    1
#
#  Prelinker and runtime library:
#  ------------------------------
#  export _C89_PVERSION="0x11050000"    2
#  export  _CC_PVERSION="0x11050000"    2
#
#
# Program (member) names of programs called by c89:
# =================================================
#
#  Compiler:
#   ---------
#   export _C89_CNAME="CBC310"     3
#   export  _CC_CNAME="CBC310"     3
#
#   Prelinker:
#   ----------
#   export _C89_PNAME="EDCPRLK"    4
#   export  _CC_PNAME="EDCPRLK"    4
#
#
# Message file (member) names used by programs called by c89:
# ===========================================================
#
#   Compiler:
#   ---------
#   export _C89_CMSGS="EDCMSGE"    5
#   export  _CC_CMSGS="EDCMSGE"    5
#
#   Prelinker:
#   ----------
#   export _C89_PMSGS="EDCPMSGE"   6
#   export  _CC_PMSGS="EDCPMSGE"   6
#
#
# High-Level Qualifier "prefixes" for data sets used by c89:
# ==========================================================
#
#   Compiler:
#   ---------
#   export _C89_CLIB_PREFIX="hlq.mlq"      7
#   export  _CC_CLIB_PREFIX="hlq.mlq"      7
#
#   Prelinker and runtime library:
#   ------------------------------
#   export _C89_PLIB_PREFIX="hlq.mlq"      8
#   export  _CC_PLIB_PREFIX="hlq.mlq"      8
#
#   High level qualifier of MVS system data sets,
#   CSSLIB, MACLIB and MODGEN.
#   -----------
#   export _C89_SLIB_PREFIX="SYS1"            9
#   export  _CC_SLIB_PREFIX="SYS1"            9
#
#
# Esoteric unit for data sets:
# ============================
#
#   Unit for (unnamed) work data sets:
#   ----------------------------------
   export _C89_WORK_UNIT="SYSALLDA"    10
   export  _CC_WORK_UNIT="SYSALLDA"    10
#
#
# ====================================
# End of c89 customization section
# ====================================
```

Figure 145. Contents of the /etc/profile File (2 of 2)

The /etc/profile file is the place were you can override c89/cc utility defaults and environment variables. Environment variables are variables that the shell or any other program can access. There are a set of defaults for the c89/cc environment variables. For a complete description of the c89/cc environment variables, see *OpenEdition MVS Command Reference*. You use the export command to override these defaults. For example, export _CC_WORK_UNIT=SYSALLDA. SYSALLDA will replace the value that _CC_WORK_UNIT is set to.

1 Uncomment this line if you are running the C/C++ compiler.

2 No need to uncomment this line, LE V1R5M0 is the default.

3 Uncomment this line if you are running the C/C++ compiler.

4 No need to uncomment this line, this is the default.

5 No need to uncomment this line, this is the default.

6 No need to uncomment this line, this is the default.

7 Depending on what _C89_CVERSION is set to will determine the default high level qualifier and middle level qualifier of the compiler libraries.

If _C89_CVERSION = ″0x1102000″ #AD/Cycle C/370 V1.2 compiler (this is the default) then, the default hlq.mlq for the C/370 V1.2 compiler is EDC.V1R2M0. If your C/370 V1.2 compiler data sets have different high level and middle level qualifiers than the default, you will need to specify those values here with the _C89_CLIB_PREFIX and _CC_CLIB_PREFIX environment variables.

If _C89_CVERSION = ″0x1301000″ #IBM C/C++ C/MVS V3.1 compiler then, the default hlq.mlq for the C/C++ C/MVS V3.1 compiler is CBC.V3R1M0. If your C/C++ C/MVS V3.1 compiler data sets have different high level and middle level qualifiers than the default, you will need to specify those values here with the _C89_CLIB_PREFIX and _CC_CLIB_PREFIX environment variables. For example, if your C compiler data sets are SYS1.CBC.**, then:

```
export _C89_CLIB_PREFIX=SYS1.CBC
export _CC_CLIB_PREFIX=SYS1.CBC
```

8 No need to uncomment this line if your LE V1R5M0 data sets are CEE.V1R5M0.**. This is the default. If your LE V1R5M0 data sets are different, for example, if your LE V1R5M0 data sets are named SYS1.CEE.**, then set:

```
export _C89_PLIB_PREFIX=SYS1.CEE
export _CC_PLIB_PREFIX=SYS1.CEE
```

9 Set this variable to the high level qualifier of your CSSLIB, MACLIB and MODGEN data sets if it is not SYS1.

10 Set this variable to the esoteric name for your work data sets. We changed it to SYSALLDA. The default is SYSDA, but that esoteric name is not defined on our system.

To complete the shell customization the following files should be copied from the /samples directory to the /etc directory:

- The mailx.rc file enables mail processing. Use the ISHELL to copy /samples/mailx.rc to /etc/mailx.rc.
- The startup.mk file contains the default rules for the make utility. Use the ISHELL to copy /samples/startup.mk to /etc/startup.mk.
- The yyparse.c and yylex.c files enable the use of the yacc and lex utilities. Use the ISHELL to copy /samples/yylex.c and /samples/yyparse.c to /etc/yylex.c and /etc/yyparse.c, respectively.
- The magic file contains a series of templates showing the different file types for the file utility. Use the ISHELL to copy /samples/magic to /etc/magic.

3.3 IPL the System to Activate OpenEdition MVS Shell and Utilities

If you installed the Shell and Utilities component you will need to IPL the system now. The Shell and Utilities component installs some elements into SYS1.LPALIB, so you will need to IPL the system with a CLPA before you start using the shell.

3.4 Customize the TERMINFO Database

Full-screen application programs like the vi editor and the more command require different terminal type characteristics to be defined. To customize the TERMINFO database, run the tic utility to build a database of different terminal types. Type in OMVS from ISPF option 6 to get into the OpenEdition MVS shell and type in the following commands:

```
tic /usr/share/lib/terminfo/ibm.ti
tic /usr/share/lib/terminfo/dec.ti
```

For more information see *OpenEdition MVS Commands Reference*.

Chapter 4. Installation Verification of OpenEdition MVS

This task will perform the steps necessary to check the installation of OpenEdition MVS. By running some of the previous tasks such as starting the OMVS kernel, running SMP/E to install OpenEdition MVS Application Services, Shell and Utilities and the dbx Debugger and using the ISHELL to perform different tasks against the file system, you have already done much of the installation verification of the OpenEdition MVS System Services component.

4.1.1 Shell and Utilities

One of the user benefits of OpenEdition MVS is the OpenEdition MVS shell. The OpenEdition MVS shell is a command processor that you use to:

- Invoke shell commands or utilities that request services from the system (similar to TSO/E)
- Write shell scripts using the shell programming language (similar to REXX)
- Run shell scripts and C-Language programs interactively (in the foreground), in the background or in batch (again, similar to REXX)

4.1.2 Invoking the Shell

Once the OpenEdition MVS Shell and Utilities features have been installed, it can be invoked through a new TSO/E command called OMVS. Like the OBROWSE, OEDIT, and ISHELL commands, the OMVS command can also be added to an ISPF selection panel, or entered as a TSO/E command.

After the OMVS command is entered, the system initializes the shell for that user.

If successful, the user is presented with:

```
IBM
Licensed Material - Property of IBM
5655-068 (C) Copyright IBM Corp. 1993, 1995
(C) Copyright Mortice Kern Systems, Inc., 1985, 1994.
(C) Copyright Software Development Group, University of Waterloo, 1989.

All Rights Reserved.

U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 - Improve performance by preventing the propagation -
 - of TSO/E or ISPF STEPLIBs                          -
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
#









 ===>                                                          INPUT

ESC=¢ 1=Help    2=SubCmd 3=HlpRetrn  4=Top      5=Bottom   6=TSO
      7=BackScr 8=Scroll 9=NextSess 10=Refresh 11=FwdRetr 12=Retrieve
```

The (#) is the character used as the shell prompt for superusers. The (¢) special character is the default to escape from certain routines. These special characters may differ from country to country depending on the code page used to interpret characters.

4.1.3 Verifying the Shell

Warning: For those users unfamiliar with the OpenEdition MVS shell, the commands are case sensitive. All commands should be entered in lower case. Upper case causes the command to be not found, and an FSUM7351 message appears.

When the OpenEdition MVS shell has been successfully initialized, you can use any of the OpenEdition MVS commands documented in *MVS/ESA OpenEdition MVS Command Reference*. The following examples have been included here to help perform some simple verification checks:

1. Enter pwd

   This results in OpenEdition MVS returning to the user the path of the present working directory. This should be the root directory, which is represented by a */*.

2. Next try env

   OpenEdition MVS returns the shell environment variables that have been used to initialize this OpenEdition MVS shell.

```
_BPX_TERMPATH=OMVS
PATH=:/bin
SHELL=/bin/sh
COLUMNS=80
_=/bin/env
LOGNAME=userid
TERM=dumb
HOME=/
LINES=20
TZ=UTC0
```

3. Enter `ls -l`

   This results in OpenEdition MVS returning a list of all the files within the present working directory and some specific file information.

```
drwxr-xr-x   3 OMVSKERN OMVSGRP        0 Oct  6 18:49 bin
drwxr-xr-x   2 OMVSKERN OMVSGRP        0 Oct 27 13:21 dev
drwxr-xr-x   4 OMVSKERN OMVSGRP        0 Oct 28 20:02 etc
drwxr-xr-x   2 OMVSKERN OMVSGRP        0 Oct 27 13:20 lib
drwxr-xr-x   3 OMVSKERN OMVSGRP        0 Oct 27 13:20 samples
drwxrwxrwx   3 OMVSKERN OMVSGRP        0 Oct 28 20:29 tmp
drwxr-xr-x   6 OMVSKERN OMVSGRP        0 Oct 28 22:16 u
drwxr-xr-x   6 OMVSKERN OMVSGRP        0 Oct 27 13:20 usr
```

4. Enter `find / -name hobbies`

   This causes OpenEdition MVS to search all of the directories in the HFS, beginning at the root directory, and display the answer on the screen. The resulting display shows the full path name for each occurrence of the file hobbies.

```
/samples/hobbies
```

5. Enter `cat /samples/profile`

   This causes OpenEdition MVS to display the contents of /samples/profile on the screen.

6. Enter `wc /samples/profile`

   The wc (word count) tells you how big a text document is.

   It counts the number of <newline>s, words, characters, and bytes in text files.

4.1.4 Batch Facilities

OpenEdition MVS provides users with a batch utility, BPXBATCH, that can be used to invoke OpenEdition MVS facilities in a batch environment.

The executable program BPXBATCH is called in the same way as any other MVS program, but because of the nature of OpenEdition MVS, there are some unique parameters and DD cards.

```
//BPXBATCH JOB (POK,999),MARTIN,MSGLEVEL=(1,1),MSGCLASS=X,
//  CLASS=A,NOTIFY=MARTIN
//*
//BPXBATCH  EXEC PGM=BPXBATCH,PARM='sh ls -al'
//STDOUT    DD PATH='/tmp/stdout',
//             PATHOPTS=(OWRONLY,OCREAT),
//             PATHMODE=SIRWXU
//STDERR    DD PATH='/tmp/stderr',
//             PATHOPTS=(OWRONLY,OCREAT),
//             PATHMODE=SIRWXU
```

In the above example, BPXBATCH is used to execute the OpenEdition MVS shell command `ls -al`. The output of this command is written to a file named stdout in the HFS. The file stdout is in the directory named /tmp. Note that this HFS path name is in lowercase. If any error messages are created, they are written to a different file, stderr in the same HFS directory as stdout.

The parameters used on the BPXBATCH DD cards are explained in *MVS/ESA OpenEdition MVS Users Guide*.

The BPXBATCH facility can be used by executing the OpenEdition MVS supplied REXX EXEC named OSHELL. Rather than code a member of a PDS or PDSE to run BPXBATCH, you can simply run OSHELL from TSO and pass OpenEdition MVS commands as a parameter. For example from ISPF option 6, type in:

```
oshell ls -al
```

Chapter 5. Customizing OpenEdition MVS for General Users

This chapter describes the tasks required to set up user HFS data sets, create home directories, place initialization files in the user's directory and change the ownership of the user's directory and initialization files so that the new user can access them.

5.1 Creating HFS Data Sets for OpenEdition MVS Users

In traditional MVS environments a general user is given their own profile and the ability to create data sets under a certain high level qualifier. In OpenEdition MVS this is also accomplished by allocating separate HFS data set(s) for each OpenEdition MVS user and mounting these separate HFS data set(s) onto the root file system at the /u directory. These separate HFS data set(s) can be used to store data unique to this OpenEdition MVS user. This practice allows each OpenEdition MVS user to use their own HFS data set(s) without impacting any other OpenEdition MVS shell users. It is also a way of isolating each OpenEdition MVS user's data for systems management purposes. IBM recommends that you adopt this approach to managing multiple file systems.

5.1.1 Creating the User File Systems

A user HFS is allocated in exactly the same way as you created the root HFS. Choose a data set name that has the user name as one of the qualifiers and a size that provides sufficient space for the user's requirements. Because all HFS data sets are under control of DFSMS, as a user adds files and extends existing files, the data set will increase in size to a maximum of 123 extents. If more space is required, you may wish to increase the size on the allocation or you may wish to create additional HFS data sets on different DASD volumes for a user and mount them at different mount points in the user's hierarchy.

Warning: The user ID you are about to define to OpenEdition MVS must already be defined to your security product with a valid OMVS segment or you will receive error messages when you try to make the user's file system available. This OMVS segment defines to OpenEdition MVS the access permissions of the user and a reference known as the UID. The OMVS segment also contains the HOME directory and the first PROGRAM that is executed when this user starts the OpenEdition MVS shell. Make sure the HOME directory in the OMVS segment matches the home directory that is defined for that user in the file system. The recommended home directory for a user is /u followed by the user ID, for example, /u/user1 would be the home directory for the user1 ID. See 2.6.7, "Define Regular OpenEdition MVS Users" on page 63 for more information on defining an OpenEdition MVS user ID to RACF. See Appendix D, "CA-ACF2 Support for OpenEdition MVS" on page 209 and Appendix E, "CA-Top Secret Support for OpenEdition MVS" on page 217 for information on defining an OpenEdition MVS user ID if you are using CA-ACF2 and CA-Top Secret.

The JCL to create a file system is in member HFSUSR in data set OMVS.JCL and in Figure 146 on page 118. Change the JCL where needed.

```
//HFSUSR   JOB (POK,999),HFSUSR,MSGLEVEL=(1,1),MSGCLASS=X,
//  CLASS=A,NOTIFY=RPETRI
//*
//STEP01  EXEC PGM=IEFBR14
//HFS       DD DSN=OMVS.USER1,SPACE=(CYL,(10,1,1)),
//             DSNTYPE=HFS,DCB=(DSORG=PO),
//             DISP=(NEW,CATLG,DELETE),
//             STORCLAS=OPENMVS,
//             VOL=SER=OP2HFS,UNIT=3390
```

Figure 146. JCL to Allocate User HFS Data Sets

5.1.2 Making User File Systems Available

After the OpenEdition MVS user's HFS data set is allocated you need to get it mounted at a mount point off the root directory to make it available. The preferred place to mount all user HFS data sets is the /u mount point. In MVS/ESA 5.2.2 there are two ways to accomplish this:

1. Direct Mount
   - Allocate an intermediate HFS data set (we called it OMVS.USERS) to be mounted between the root file system and all user file systems. Create a mount point using the `mkdir` command and issue the `mount` command (To make the mount permanent you will also need to add the HFS data set name and its mount point to the BPXPRMxx member of parmlib). This is shown in Figure 147 on page 119.

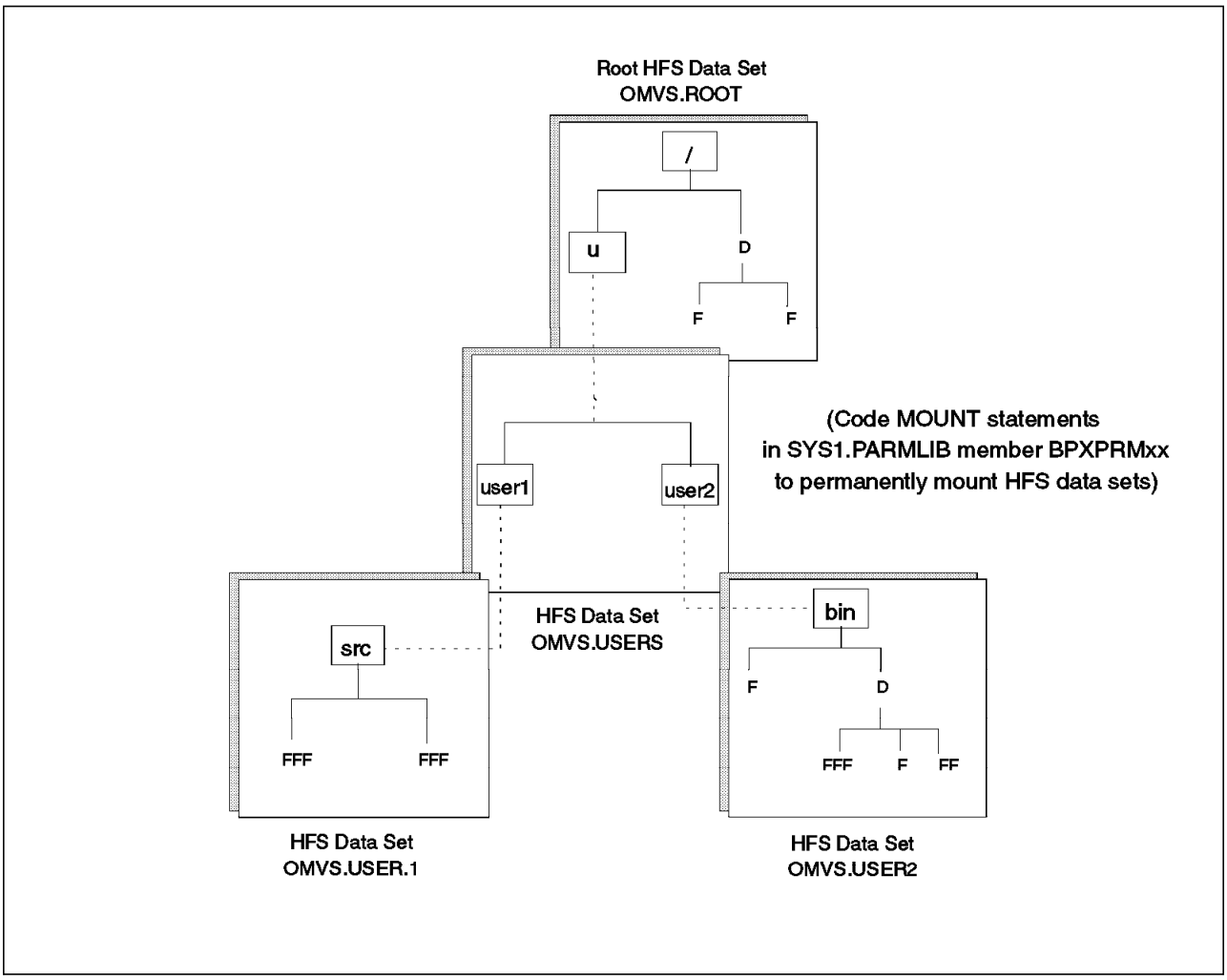


Figure 147. Direct Mount

2. Automount Facility

   - Customize the automount facility to control all user file systems to automatically mount them when they are needed. This is the preferred method to manage user HFS data sets because it saves administration time. This is shown in Figure 148 on page 120. See 5.1.2.2, "Customizing the Automount Facility" on page 125 for more information on setting up the automount facility.

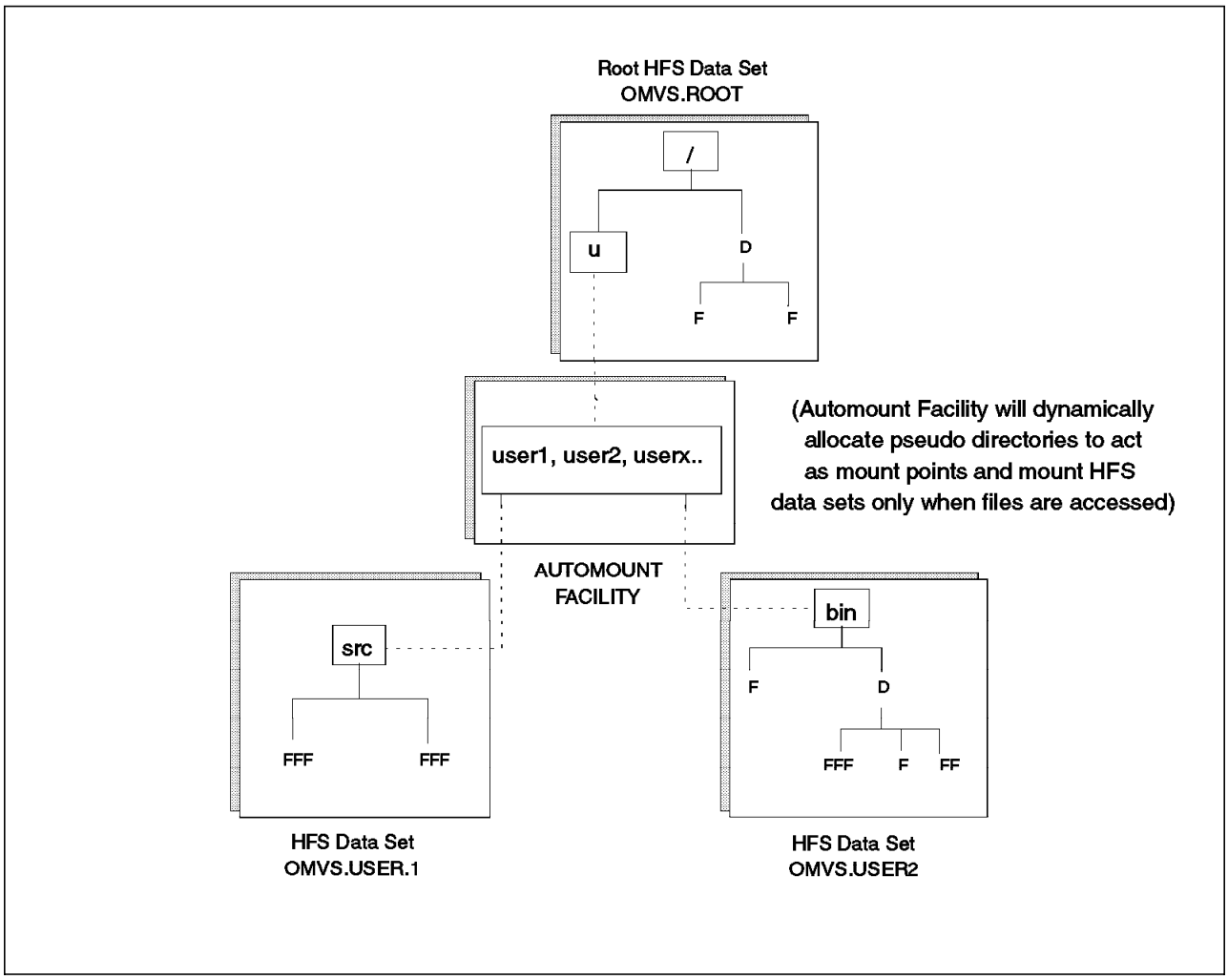


Figure 148. Automount Facility

5.1.2.1 Using Direct Mount

The root file system should be set up so that it doesn't require frequent changes or updates (outside of SMP/E maintenance). To achieve this we will allocate an intermediate HFS data set called OMVS.USERS and mount it at the /u directory. All user directories that are added will reside in this new physical HFS and not in the root HFS. The JCL to allocate this intermediate HFS is in member HFSINT in data set OMVS.JCL and in Figure 149 on page 121. Change the JCL to fit your environment.

```
//HFSINT   JOB (POK,999),HFSINT,MSGLEVEL=(1,1),MSGCLASS=X,
//  CLASS=A,NOTIFY=RPETRI
//*
//STEP01  EXEC PGM=IEFBR14
//HFS       DD DSN=OMVS.USERS,SPACE=(CYL,(5,1,1)),
//             DSNTYPE=HFS,DCB=(DSORG=PO),
//             DISP=(NEW,CATLG,DELETE),
//             STORCLAS=OPENMVS,
//             VOL=SER=OP2HFS,UNIT=3390
```

Figure 149. JCL to Allocate Intermediate HFS Data Set

The next thing to do is mount this new intermediate HFS data set at the /u directory. The mount can be performed from an ID that has superuser authority by:

- Using the /samples/mountx REXX exec from the shell
- Using the TSO MOUNT command
- Using the ISHELL File_systems pull-down
- Adding an entry to the BPXPRMxx member in SYS1.PARMLIB and stopping/restarting OpenEdition MVS

An example of the commands required, including issuing the /samples/mountx REXX exec from the OpenEdition MVS shell is shown in Figure 150 on page 122. Type in OMVS from ISPF option 6 to enter the shell and execute the highlighted commands to mount the HFS data set OMVS.USERS.

```
IBM
Licensed Material - Property of IBM
5655-068 (C) Copyright IBM Corp. 1993, 1995
(C) Copyright Mortice Kern Systems, Inc., 1985, 1994.
(C) Copyright Software Development Group, University of Waterloo, 1989.

All Rights Reserved.

U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 - Improve performance by preventing the propagation -
 - of TSO/E or ISPF STEPLIBs                          -
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# /samples/mountx /u omvs.users          1
OMVS.USERS is now mounted at
/u
# df -P                          2
Filesystem           512-blocks        Used  Available  Capacity Mounted
OMVS.USERS                7200          40       7160        1% /u
OMVS.ROOT                82800       79608       3192       97% /
# chmod 755 /u                  3

 ===>

ESC=¢ 1=Help    2=SubCmd 3=HlpRetrn  4=Top      5=Bottom   6=TSO
      7=BackScr 8=Scroll 9=NextSess 10=Refresh 11=FwdRetr 12=Retrieve
```

Figure 150. Mounting the New Intermediate HFS Data Set

1 Run the mountx REXX exec to mount the HFS data set, OMVS.USERS on mount point, /u.

2 Run the display free space command to display the mounted file systems.

3 Change the permission bits to allow access to the /u directory.

Now that the OMVS.USERS HFS data set is mounted at mount point /u you can create the /user1 mount point from a superuser ID by:

- Using the mkdir command in the OpenEdition MVS shell
- Using the TSO MKDIR command
- Using the ISHELL Directory pull-down

Figure 151 on page 123 shows the sequence of commands performed by a superuser in the OpenEdition MVS shell to create a mount point for a new user off the /u directory. Type in OMVS from ISPF option 6 to enter the shell and execute the highlighted commands to create the user1 mount point. Remember, before creating a mount point directory for a new user, the new user must be defined to your security product and have a valid OMVS segment.

```
IBM
Licensed Material - Property of IBM
5655-068 (C) Copyright IBM Corp. 1993, 1995
(C) Copyright Mortice Kern Systems, Inc., 1985, 1994.
(C) Copyright Software Development Group, University of Waterloo, 1989.

All Rights Reserved.

U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 - Improve performance by preventing the propagation -
 - of TSO/E or ISPF STEPLIBs                         -
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# cd /u                          1
# pwd                           2
/u                              3
# mkdir -m 700 user1            4
# chown -R user1 /u/user1       5
# ls -l                         6
total 0
drwx------   2 USER1   OMVSGRP        0 Nov  7 09:07 user1
#
 ===>

ESC=¢ 1=Help    2=SubCmd 3=HlpRetrn  4=Top      5=Bottom   6=TSO
      7=BackScr 8=Scroll 9=NextSess 10=Refresh 11=FwdRetr 12=Retrieve
```

Figure 151. Creating a User's Mount Point Directory

1 Change to make the /u directory your current working directory.

2 Check to make sure /u is the current working directory.

3 /u is the current working directory.

4 Create a new directory for user1 setting the permission bits to 700. See 2.6.1.1, "UNIX Security and Files" on page 33 for information on permission bit settings.

5 The `mkdir` command issued above will cause the owner of this new directory to be the superuser ID that issued the `mkdir` command. In order for this new user (USER1) to use this new file system you will need to issue the `chown` command to change the ownership. Issue this command to set the owner field of this mount point directory to the user1 ID.

6 Issue a list command to display the new directory for user1.

The user HFS that was previously created can now be mounted at the /u/user1 directory. The mount can be performed by:

- Using the /samples/mountx REXX exec from the shell
- Using the TSO MOUNT command
- Using the ISHELL File_systems pull-down

- Adding an entry to the BPXPRMxx member in SYS1.PARMLIB and stopping/restarting OpenEdition MVS

An example of the commands required, including issuing the /samples/mountx REXX exec from the OpenEdition MVS shell is shown in Figure 152. Type in OMVS from ISPF option 6 to enter the shell and execute the highlighted commands to mount the HFS data set OMVS.USER1.

```
IBM
Licensed Material - Property of IBM
5655-068 (C) Copyright IBM Corp. 1993, 1995
(C) Copyright Mortice Kern Systems, Inc., 1985, 1994.
(C) Copyright Software Development Group, University of Waterloo, 1989.

All Rights Reserved.

U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 - Improve performance by preventing the propagation -
 - of TSO/E or ISPF STEPLIBs                          -
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# /samples/mountx /u/user1 omvs.user1     1
OMVS.USER1 is now mounted at
/u/user1
# df -P                            2
Filesystem          512-blocks        Used  Available  Capacity Mounted
OMVS.USER1               12960          40      12920        1% /u/user1
OMVS.ROOT                82800       79608       3192       97% /
#

 ===>

ESC=¢ 1=Help    2=SubCmd 3=HlpRetrn  4=Top      5=Bottom   6=TSO
      7=BackScr 8=Scroll 9=NextSess 10=Refresh 11=FwdRetr 12=Retrieve
```

Figure 152. Mounting the New HFS Data Set

1 Run the mountx REXX exec to mount the HFS data set, OMVS.USER1 on mount point, /u/user1.

2 Run the display free space command to display the mounted file systems.

If you want to make the mounting of the OMVS.USERS and OMVS.USER1 HFS data sets permanent, you have to add the following entry in the BPXPRMxx member of SYS1.PARMLIB as shown in Figure 153 on page 125. These two mount statements should follow the ROOT statement for the root file system as shown in Figure 125 on page 90.

```
MOUNT    FILESYSTEM('OMVS.USERS')
         TYPE(HFS)
         MOUNTPOINT('/u')
         MODE(RDWR)

MOUNT    FILESYSTEM('OMVS.USER1')
         TYPE(HFS)
         MOUNTPOINT('/u/user1')
         MODE(RDWR)
```

Figure 153. Adding HFS Data Set Mount Statements to the BPXPRMxx Member of PARMLIB

5.1.2.2 Customizing the Automount Facility

The automount facility lets you designate directories as containing only mount points. This is the preferred method of managing user HFS data sets. As each of these mount points is accessed, an appropriate file system is mounted. The mount point directories are internally created as they are required. Later, when the file system is no longer in use the mount point directories are deleted. Try to think of automount as an administrator that has total control over a directory: when a name is accessed in this directory it looks up in its policy what file system is supposed to be associated with that name. If it finds one, it (logically) does a mkdir followed by a mount and quietly moves out of the way. Once out of the way, the root directory of that newly mounted file system is now accessed as that name. For example, in the previous section we created the user1 directory with the `mkdir` command. With automount active and the correct automount policy in place, there is no need to create a user1 directory with the `mkdir` command, the user1 directory will be dynamically allocated and the OMVS.USER1 data set will be automatically mounted at the /u/user1 mount point. Later, if the /u/user1 file system has not been accessed based on certain criteria in your automount policy, the OMVS.USER1 data set will automatically be unmounted. See *MVS/ESA Planning: OpenEdition MVS* and *MVS/ESA OpenEdition MVS Command Reference* for additional information about this facility.

In the following steps we will show how to set up the automount facility to mount user file systems.

Step 1 To use the automount facility, add the following statement to your BPXPRMxx parmlib member and stop and restart the OpenEdition MVS kernel. If you used the BPXPRM00 member supplied in data set OMVS.PARMLIB, this statement is already included.

```
FILESYSTYPE TYPE(AUTOMNT) ENTRYPOINT(BPXTAMD)
```

Step 2 Customize the definition files

The automount facility uses two definition files, a master file and a MapName file. The default file name of the master file is /etc/auto.master. The /etc/auto.master file contains the directory or directories that will be monitored by automount and the associated MapName file(s) which contain the mount

parameters. Figure 154 on page 126 contains an example of a /etc/auto.master file.

```
BROWSE -- /etc/auto.master  ------
COMMAND ===>
*************** Top of Data **********
/u                /etc/u.map
*************** Bottom of Data *******
```

Figure 154. Example of an /etc/auto.master File

The master file defines that automount should manage the /u directory. Which means that as soon as someone using OpenEdition MVS services tries to access a directory that is mounted off the /u directory, automount will automatically mount the HFS data set based on the MapName policy in Figure 155.

Note: In order to make the most efficient use of an automount MapName policy that contains generic entries, it is important to come up with a consistent HFS data set naming convention. In our examples here, all our HFS data sets have a highlevel qualifier of OMVS and the lower level qualifier is equal to the user ID.

```
 BROWSE -- /etc/u.map -------------
 COMMAND ===>
*************** Top of Data **********
name              *
type              HFS
filesystem        OMVS.<uc_name>
mode              rdwr
duration          nolimit
delay             0
*************** Bottom of Data *******
```

Figure 155. Example of a Generic Entry in a MapName File

The MapName file contains the mapping between a subdirectory of a directory managed by automount and the mount parameters.

The MapName file can contain specific entries and a generic entry. There should be only one generic entry in a MapName file and it has to be the first one. When the automounter tries to resolve a lookup request, it attempts to find a specific entry. If a specific entry does not exist for the name being looked up, it attempts to use the generic entry.

The variable *<uc_name>* means convert the name being looked up to upper case. Whenever this variable is encountered it is replaced by the name being looked up. A directory with the looked up name is created and used as a mount point for the file system to be mounted. The *<uc_name>* can be used to replace any level qualifier in the data set. For example, if the

name of the directory that is being looked up is user1, automount will resolve the name in the following ways:

```
OMVS.<uc_name> = OMVS.USER1
OMVS.<uc_name>.HFS = OMVS.USER1.HFS
```

The *<uc_name>* variable is replaced with the uppercase name absolutely anywhere in the string.

Examples of both the /etc/auto.master and /etc/u.map files that will enable the automatic mounting of all user HFS data sets off the /u directory can be found in data set OMVS.JCL under the member names AMNTMSTR and AMNTMAP respectively. Create these files in the file system.

Step 3 Start the automount facility

Figure 156 on page 128 contains an example of starting the automount facility (from the shell) and how file systems are automatically mounted. The automount command can only be issued from a superuser ID. It has the following syntax:

```
automount [-s][Master filename]
```

When running the command with no arguments, the automount facility reads the /etc/auto.master file to determine the directories to be monitored and the file names that contain their configuration specifications, the MapName file(s). If automount is used with a master file name specified, that file name is used instead of /etc/auto.master.

The -s option only checks the syntax of the configuration file. The automount policy is not activated.

Figure 156 on page 128 shows how *<uc_name>* works with the /etc/auto.master and /etc/u.map files from Figure 154 on page 126. HFS data sets, OMVS.RPETRI, OMVS.SLEKKA and OMVS.USER1 have already been allocated. The low level qualifier of the HFS data sets is the user ID which is also the directory mount point that automount will dynamically allocate. With the automount facility as soon as a user tries to access any directory in their HFS file system, the HFS data set will be automatically mounted off the /u directory. Type in OMVS from ISPF option 6 to access the OpenEdition MVS shell.

Warning: When an HFS data set is first allocated for a new user (in our case, USER1 shown in Figure 146 on page 118), and automount is used to dynamically allocate a mount point, this new mount point directory has permission bits that are set to 700 and the owner field is set to a superuser ID name. In order for USER1 to be able to use this new file system you will need to issue the chown command to change the owner field of this new mount point directory that was created by automount. Figure 156 on page 128, item **7** shows a superuser issuing the chown command against the /u/user1 directory to change the ownership.

```
IBM
Licensed Material - Property of IBM
5655-068 (C) Copyright IBM Corp. 1993, 1995
(C) Copyright Mortice Kern Systems, Inc., 1985, 1994.
(C) Copyright Software Development Group, University of Waterloo, 1989.

All Rights Reserved.

U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 - Improve performance by preventing the propagation -
 - of TSO/E or ISPF STEPLIBs                          -
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# df
Mounted on     Filesystem                Avail/Total    Files      Status
/              (OMVS.ROOT)               1432/89280     0          Available
# automount                  1
FOMF0107I Processing file /etc/u.map
FOMF0108I Managing directory /u             2
# df                                 3
Mounted on     Filesystem                Avail/Total    Files      Status
/u             (*AMD/u)                  0/8            0          Available
/              (OMVS.ROOT)               1432/89280     0          Available
# cd /u/user1                               4
# cd /u/slekka/testdir                      4
# cd /u/rpetri                              4
# df                                        5
Mounted on     Filesystem                Avail/Total    Files      Status
/u             (*AMD/u)                  0/8            0          Available
/u/rpetri      (OMVS.RPETRI)             4256/4320      0          Available
/u/slekka      (OMVS.SLEKKA)             4232/4320      0          Available
/u/user1       (OMVS.USER1)              4232/4320      0          Available
/              (OMVS.ROOT)               1432/89280     0          Available
# ls -l /u                                  6
total 0
drwxr-xr-x   2 RPETRI   OMVSGRP       0 Nov  2 09:59 rpetri
drwxr-xr-x   2 SLEKKA   OMVSGRP       0 Nov  1 09:47 slekka
drwx------   2 OMVSKERN OMVSGRP       0 Nov  7 09:07 user1
# chown -R user1 /u/user1                   7
# ls -l /u                                  8
total 0
drwxr-xr-x   2 RPETRI   OMVSGRP       0 Nov  2 09:59 rpetri
drwxr-xr-x   2 SLEKKA   OMVSGRP       0 Nov  1 09:47 slekka
drwx------   2 USER1    OMVSGRP       0 Nov  7 09:07 user1
#

 ===>

ESC=¢ 1=Help    2=SubCmd 3=HlpRetrn  4=Top      5=Bottom   6=TSO
      7=BackScr 8=Scroll 9=NextSess 10=Refresh 11=FwdRetr 12=Retrieve
```

Figure 156. Result of Usage of <uc_name> in MapName File

1 Here we see the automount command being issued from a superuser ID to start the automount facility from the OpenEdition MVS shell.

2 Automount will scan the /etc/master.auto file first to see what MapName file(s) should be read. Here we are managing the /u directory.

Note: If the automount command is erroneously called twice this doesn't create any problems regardless of whether a file system is already mounted or not. Automount will re-read the /etc/master.auto file and associated MapName file(s) and pick up any changes.

3 The display free space command (df) is issued that shows that the automount facility is started and it is monitoring the /u directory (*AMD/u).

4 Change directory commands are issued to access directories in three file systems to be mounted off the /u directory. In this case, the directories user1, rpetri and slekka are used to resolve the <uc_name> symbol in the /etc/u.map file. The rpetri, slekka and user1 directory names are translated to uppercase and substituted to build the HFS data set names, OMVS.RPETRI, OMVS.SLEKKA and OMVS.USER1 respectively. The rpetri, slekka and user1 directories do not physically exist in any file system but they will be created as pseudo mount points by the automount facility on which the HFS data sets OMVS.RPETRI, OMVS.SLEKKA and OMVS.USER1 are mounted.

5 Output from another display free space command (df) shows (*AMD/u) is monitoring the /u directory. It also shows the OMVS.RPETRI, OMVS.SLEKKA and OMVS.USER1 data sets are now mounted at pseudo mount points /u/rpetri, /u/slekka and /u/user1 respectively.

Note: When automount is actively monitoring a particular mount point (in this case /u) it is no longer possible to add a file to this directory (/u) or create a new subdirectory off the /u directory using the mkdir command. If attempted, you will receive message:

```
EDC515I Dynamic allocation error
```

6 The `ls -l /u` command is issued against the /u directory and the directory attributes are displayed.

7 The `chown` command is issued to change the ownership of the /u/user1 directory from OMVSKERN to USER1.

8 The `ls -l /u` command is issued again to show that the owner field of the /u/user1 directory is now set to USER1.

Step 4 Automatically starting automount when the OMVS kernel is started

When you have everything customized and working you can have the automount facility started when the OpenEdition MVS kernel is started by adding it to the /etc/rc file. Add the following line to the /etc/rc file:

```
# Start the automount facility
usr/sbin/automount
```

Step 5 Using a specific entry in a MapName file

Use specific entries for directory names when the parameters you wish to use differ from the generic entry. Any parameters that are not specified are inherited from the generic entry. A specific entry defines a directory name called wjs in the name parameter of the MapName file rather then an * as shown in Figure 157. Also notice in this example the duration for generic mounts is set to unmount idle file systems after 60 minutes, but in our specific mount entry, idle file systems will stay mounted indefinitely.

```
BROWSE -- /etc/auto.master  ------
COMMAND ===>
*************** Top of Data **********
/u         /etc/u.map
*************** Bottom of Data *******
 BROWSE -- /etc/u.map  ---------
 COMMAND ===>
*************** Top of Data **********
name               *
type               HFS
filesystem         OMVS.<uc_name>
mode               rdwr
duration           60
delay              0
/*
name               wjs
type               HFS
filesystem         OMVS.WJS.HFS
mode               rdwr
duration           nolimit
delay              0
*************** Bottom of Data *******
```

Figure 157. Specific Entry in a MapName File

Whenever the directory /u/wjs is referenced by a command such as cd or cp, automount will mount HFS data set OMVS.WJS.HFS.

Things to be aware of for generic and specific entries

- Do not use a **/** in front of the name of the directory to be mounted in a specific entry in a MapName file. For example, in /etc/u.map:

  ```
  name          wjs
  ```

  is correct.

  ```
  name          /wjs
  ```

  is NOT correct.

- The directory name and the data set name qualifier for the HFS data set which is replaced by the variable *<uc_name>* have to be the same. Otherwise you will get error messages like:

  EDC129I No such file or directory
  or
  EDC515I Dynamic allocation error.

5.1.3 Customizing $HOME/.profile

When you start the OpenEdition MVS shell, it uses information in three files to set up environmental variables that determines system and user defaults and preferences. The information is gathered in these files, in this order:

1. /etc/profile
2. $HOME/.profile (notice the . dot)
3. The file that the ENV variable specifies (for example, $HOME/.setup)

The /etc/profile was discussed in 3.2, “Customize the Shell and Utilities” on page 102. It is the first file that is searched and sets up a default system wide user environment. This file is normally set up by the system programmer or administrator to reflect any system-wide requirements like setting the local time zone. The /etc/profile file is also used to set c89/cc compiler and run-time library environment variables.

The $HOME/.profile file (where $HOME is a variable for the individual user’s home directory) is an individual user profile. You can override any values that are set in the /etc/profile file by coding them in your own $HOME/.profile file. A sample $HOME/.profile file is supplied in /samples/.profile, and should be copied into each user’s home directory. Figure 158 on page 132 shows the commands to copy the /samples/.profile file into a user’s home directory. Type in OMVS from ISPF option 6 to enter the OpenEdition MVS shell. Run this from a user ID that has superuser authority.

```
IBM
Licensed Material - Property of IBM
5655-068 (C) Copyright IBM Corp. 1993, 1995
(C) Copyright Mortice Kern Systems, Inc., 1985, 1994.
(C) Copyright Software Development Group, University of Waterloo, 1989.

All Rights Reserved.

U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 - Improve performance by preventing the propagation -
 - of TSO/E or ISPF STEPLIBs                          -
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# cd /                          1
# pwd                          2
/
# cp /samples/.profile /u/user1/.profile   3
#
# cd /u/user1                              4
# pwd                                      5
/u/user1
# chown user1 .profile                     6
# ls -ali .profile                         7
 10 -rwxr-xr-x   1 USER1   OMVSGRP       979 Nov  8 17:16 .profile


 ===>

ESC=¢ 1=Help    2=SubCmd 3=HlpRetrn  4=Top      5=Bottom   6=TSO
      7=BackScr 8=Scroll 9=NextSess 10=Refresh 11=FwdRetr 12=Retrieve
```

Figure 158. Copying the .profile File into the User's Home Directory

1 Change to make the root your current working directory.

2 Check to make sure / is the current working directory.

3 Copy the /samples/.profile file to the /u/user1 directory.

4 Change to make the /u/user1 the current working directory.

5 Check to make sure /u/user1 is the current working directory.

6 The cp command issued above will cause the owner of this new file to be the superuser ID that issued the cp command. In order for this new user (USER1) to use this file you will need to issue the chown command to change the ownership. Issue this command to set the owner field of this file to the user1 ID.

7 Run the `ls -ali` command to list out the .profile file.

Figure 159 on page 133 shows the /samples/.profile file.

```
# This is a sample .profile defining login environment parameters for
# an individual user. This should be copied to the user's $HOME/.profile
# and modified to their individual taste. More information may be
# found in the MVS/ESA OpenEdition User's Guide, in the chapter on
# customizing the shell.
#
# You may add or remove a # to disable or enable the settings below.

# The following variable may be set to point to a login script.
# In this case it is set to a file called .setup in the home
# directory. Refer to the User's Guide for more information on how to
# use this variable.

# ENV=$HOME/.setup
# export ENV

# This line appends your home directory to the current path.
PATH=$PATH:$HOME:

# This line sets the default editor to ed.
EDITOR=ed

# This line sets the prompt to display your login name, and current
# directory.
PS1='$LOGNAME':'$PWD':' >'

# This line exports the variable settings so that they are known to the
# system.
export PATH EDITOR PS1
```

Figure 159. Contents of /samples/.profile File

For most OpenEdition MVS users this starter .profile will suffice. See *MVS/ESA OpenEdition MVS User's Guide* for more information about how a user can further customize their own $HOME/.profile file.

5.2 Printing from the Shell

You can print to JES controlled printers from the OpenEdition MVS shell.

Figure 160 on page 134 shows the printer initialization statements in a JES2 initialization deck. The DESTID statement specifies a system programmer defined name for the JES2 route code to a printer. For more information on JES2 printer initialization statements, see *MVS/ESA SP V5 JES2 Initialization and Tuning Reference*.

```
                          /**** Printer Parameters ****************/
PRINTER1 FSS=FSS382C,MODE=FSS,PRMODE=(PAGE,LINE)
PRINTER1 CLASS=IUC,UCS=0,CKPTPAGE=100
PRINTER1 ROUTECDE=U10,START=NO
PRINTER2 WS=(/R,PRM,R,Q),CLASS=IUC,CKPTPAGE=150
PRINTER2 MODE=FSS,FSS=FSS24,TRKCELL=YES,PRMODE=(LINE,PAGE)
PRINTER2 ROUTECDE=U11,UNIT=B10
PRINTER3 FSS=FSS382B,MODE=FSS,PRMODE=(PAGE,LINE),START=NO
PRINTER3 CLASS=CIU,UCS=0,CKPTPAGE=100
PRINTER3 ROUTECDE=U12
                         /**** Symbolic Destinations *************/
DESTID(PRINTER1) DEST=U10      /* 3820 RAISED FLOOR AREA            */
DESTID(PRINTER2) DEST=U11      /* 3827 IN ROOM 2E16 B/008           */
DESTID(PRINTER3) DEST=U12      /* 3820 IN ROOM 2C7                  */
```

Figure 160. Printer Definitions for JES2

Figure 161 shows the printer initialization statements in a JES3 inish deck. The DEVICE statement specifies a system programmer defined name for the JES3 printer. For more information on JES3 printer initialization statements, see *MVS/ESA SP V5 JES3 Initialization and Tuning Reference*.

```
*  P20N4 IS A 3820 IN AISLE 11 BLDG. 005S
*
DEVICE,DTYPE=PRT3820,JNAME=P20N4,
JUNIT=(,NB3,S30,OFF),FSSNAME=P20N4,MODE=FSS,
WS=(D,P,U),PM=(LINE,PAGE),BURST=NO,CKPNTPG=25
```

Figure 161. Printer Definitions for JES3

Tell the application programmers the destinations or system programmer defined printer names in order to print from the shell. The dest option of the lp command uses the destinations or system programmer defined printer names to send the output. For example, in the shell if you entered:

```
lp -d U10,i hello.c
```

the file, hello.c would be printed on printer U10 as shown in Figure 160. If you entered:

```
lp -d P20N4,i server.c
```

the file, server.c would be printed on printer P20N4 as shown in Figure 161.

The destinations for lp can also be set in environment variables LPDEST and PRINTER. The LPDEST environment variable takes precedence over the PRINTER environment variable. If there is only one printer, the system programmer can consider adding that to /etc/profile or if there are more than one printer, each user could add the one that they prefer to their own .profile. For example:

```
export LPDEST=U11
```

This statement in either /etc/profile or a user's .profile file will set the printer destination to U11. With the printer destination set, a user can leave off the printer destination on the lp command. For example:

```
lp hello.c
```

You can also print using batch. In this example, the printer is a 3280 attached to a remote VM system:

```
//DORNIAKA JOB 'C9003P,B9211092','DORNIAK',NOTIFY=DORNIAK,
// MSGLEVEL=(1,1),CLASS=A,MSGCLASS=T
//*       print a HFS file
//STEP1    EXEC PGM=IKJEFT01
//JES      DD  SYSOUT=I,DCB=(RECFM=V,LRECL=256),
//          DEST=(WTSCPOK,3820C16)
//HFSFILE1 DD  PATH='/usr/adsm/README',PATHOPTS=ORDONLY
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  *
PROF MSGID WTPMSG
 /* print with ocopy                    */
ocopy indd(HFSFILE1) outdd(JES)
 /* print with lp                       */
 bpxbatch sh /bin/lp -d wtscpok.3820c16,i /usr/adsm/options.doc
/*
```

5.3 Help Facilities

An optional step for the installation of the OpenEdition MVS environment is to install books and a bookshelf for the online help facility. Once the books are installed, this facility is accessed via the TSO/E command OHELP.

In order to be able to use this facility, it is necessary to have BookManager READ/MVS installed on your system.

If you have BookManager READ/MVS installed, see *MVS/ESA Planning: OpenEdition MVS* for more information on installing books for the OHELP command.

You can include as many books as you want, and to obtain information from the books, just type the following command.

```
OHELP refid name
```

Figure 162. TSO/E Command for Help (1 of 3)

The refid operand represents a specific book you want to search, and the name operand represents the element for which you are looking for information. Every book is assigned a refid, so if you know it, just include the refid.

If you don't specify a refid, as in the next figure, but you include the element you want to search, by default you will be pointed to the book with the refid of 1. This is normally setup as the commands reference book.

```
OHELP name
```

Figure 163. TSO/E Command for Help (2 of 3)

If you only specify a refid, and do not include an element, as in the next figure, you will be shown the table of contents of the book required by refid.

```
OHELP refid
```

Figure 164. TSO/E Command for Help (3 of 3)

If you want to see the list of available books and the refid associated with them, just type the OHELP command.

The OHELP command can be entered in TSO/E, ISPF, and in the OpenEdition MVS shell.

For more information on the OHELP facilities and the `man` command facilities in the OpenEdition MVS environment refer to the *MVS/ESA OpenEdition MVS User's Guide* manual.

5.4 Performance

System Resources Management (SRM) is the MVS component that determines which address spaces should be given access to system resources, and the rate at which the address spaces are allowed to consume those resources. The access to resources is controlled by assigning address spaces to performance groups. Installation policy for access to resources is on a performance group basis and is defined in the Installation Performance Specification (IPS) (IEAIPSxx member of SYS1.PARMLIB). The IPS will associate a dispatching priority and swapping objectives with a performance group to control access to the processor resources relative to other performance groups. Assignment of performance groups to MVS address spaces is controlled by the Installation Control Specification (IEAICSxx member of SYS1.PARMLIB).

Installing the OpenEdition MVS component requires further customization steps towards the control and performance specifications of the installation represented by the ICS and IPS SYS1.PARMLIB members.

Step 1 Figure 165 on page 137 shows the IEAIPSxx parmlib member statements that could be used to support the performance groups in the IEAICSxx parmlib member in Figure 166 on page 138 for OpenEdition MVS.

```
DMN=40,CNSTR=(4,8), ...         /* OpenMVS init processes */


PGN=9,(DMN=9,DP=F83)
PGN=10,(DMN=10,DP=F81)          /* OpenMVS kernel         */
PGN=2,(DMN=2,DP=F54,DUR=400)
      (DMN=3,DP=F52,DUR=1400)
      (DMN=4,DP=F44)
PGN=20,(DMN=20,DP=F54,DUR=400)
       (DMN=3,DP=F52,DUR=1400)
       (DMN=4,DP=F44)
PGN=5,(DMN=5,DP=F53,DUR=2K)   /* OpenMVS forked children */
      (DMN=6,DP=F51,DUR=4K)
      (DMN=7,DP=F43)
PGN=40,(DMN=40,DP=F64)         /* OpenMVS init processes */
PGN=50,(DMN=50,DP=F53,DUR=2K) /* Special for user super1 */
       (DMN=6,DP=F51,DUR=4K)
       (DMN=7,DP=F43)
PGN=60,(DMN=60,DP=F53,DUR=2K) /* Special for acct D001 */
       (DMN=6,DP=F51,DUR=4K)
       (DMN=7,DP=F43)
```

Figure 165. Parameters for OpenEdition MVS in the IEAIPSxx Parmlib Member

Perform the following when customizing the IEAIPSxx parmlib member statements for OpenEdition MVS.

- Provide more than one performance period for forked address spaces. The service requirements for these address spaces are the following:
  - May be similar to medium-length TSO/E transactions, and for others similar to long-running batch jobs.
  - May be similar to started tasks, in that they will run better if not penalized for being long running. Daemons and long running compiles fit into this category.
- Give the OpenEdition MVS initialization process, OMVSINIT, with PGN=40 in our example in Figure 166 on page 138, a dispatching priority higher than all other OpenEdition MVS address spaces. INIT needs a high priority to get control when needed and to clean up orphan processes, which remain after their parent processes have ended.
- Allow the OpenEdition MVS initialization process a multiprogramming level (MPL) high enough to support the OpenEdition MVS initialization process (OMVSINIT), the /etc/init process, and any forked child processes (daemons).
- The MPL value for started tasks must be high enough to include all APPC initiators.

```
SUBSYS=STC,PGN=9
  TRXNAME=OMVS,PGN=10               /* OpenMVS kernel */


SUBSYS=TSO,PGN=2
  USERID=SUPER1,PGN=20

 .

 SUBSYS=OMVS,PGN=5              /* OpenMVS forked children */
   USERID=OMVSKERN,PGN=40       /* OpenMVS init processes */
   USERID=SUPER1,PGN=50         /* Special for user super1 */
   ACCTINFO=D001(1),PGN=60      /* Special for acct D001 */
```

Figure 166. Parameters for OpenEdition MVS in the IEAICSxx Parmlib Member

TRXNAME=OMVS specifies a performance group for the OpenEdition MVS kernel address space. The OpenEdition MVS kernel is nonswappable, but its performance group and dispatching priority are defined by the IEAICSxx and IEAIPSxx parmlib members.

You should specify a SUBSYS=OMVS section for all MVS systems with OpenEdition MVS This section assigns performance groups to forked address spaces. These performance group assignments do not apply to dubbed address spaces, such as batch programs that issue OpenEdition MVS callable services. If a TSO/E batch, or started task address space uses OpenEdition MVS services. It does not change subsystem type; that is, it does not use SUBSYS=OMVS.

In the SUBSYS=OMVS section, use USERID and ACCTINFO statements to assign performance groups. The section cannot contain TRXNAME and TRXCLASS parameters.

- USERID statements. The section should specify USERID=OMVSKERN to provide a performance group for the OpenEdition MVS initialization process (INIT), the /etc/init process, and any forked child processes (daemons). USERID statements can also be used to provide different users with different performance groups.

  A forked child address space inherits the user ID of its parent. It may still be in a different performance group. For example, if TSO/E user SUPER1 with PGN=20 issues a fork, the forked address space has PGN=50 from the USERID=SUPER1 statement in the SUBSYS=OMVS section.

- ACCTINFO statements. OpenEdition MVS fork processing always propagates accounting data from the parent to the child. In addition, when a daemon creates a process for another user, accounting data is taken from the WORKATTR segment of the RACF user profile. In the example, the following forked child processes have been placed in PGN=60. The ACCT section begins with D001 as shown in Figure 166.

Step 2 Placing Language Environment Modules in Shared Storage

Placing certain Language Environment routines in shared storage reduces overall system storage requirements. Also initiate/terminate (init/term) is reduced for each application, since load time decreases. This also improves compiler performance since the compiler uses the LE run-time libraries. See *Language Environment Installation and Customization on MVS* for more information on what modules are LPA eligible.

Note: Putting the entire SCEERUN data set as a STEPLIB defeats the purpose of placing the modules in LPA since STEPLIB, if allocated is read first. Figure 167 on page 140 shows the IEALPAxx member that can be used to place frequently used run-time library routines in the ELPA. This member is also found in member IEALPA00 in data set OMVS.PARMLIB.

```
INCLUDE LIBRARY(SYS1.LINKLIB)
     MODULES(BPXEV003,BPXZV,BPXOLVD,BPXPLPKA,BPXOV,BPXOVI,BPXZ24,
             BPXBINIT,IEFIB600)
INCLUDE LIBRARY(CEE.V1R5M0.SCEERUN)
              MODULES(CEEBINIT,
              CEEBLIBM,CEEBLIIA,CEEBLRR,CEEBPICI,CEECCICS,
              CEECOPP,CEECTCB,
              CEELCLE,CEELRRIN,CEELRRTR,
              CEEMENU0,CEEMENU2,CEEMENU3,CEEMENU4,CEEMENU5,
              CEEMUEN0,CEEMUEN2,CEEMUEN3,CEEMUEN4,CEEMUEN5,
              CEEOLVD,CEEPIPI,CEEPLPKA,CEEQMATH,
              CEEEV003,EDCZ24,EDCNOSN1,
              EDC$LCNM,EDCUCSNM,EDCOV,EDCOVI,
              EDC$FRAN,EDC$GERM,EDC$ITAL,EDC$SPAI,EDC$UK,EDC$S370,
              EDC$USA,EDCZEMSG,EDCZUMSG,EDCZV,IEDCMSGT,
              EDCNINSP,EDCALIAS,EDCPRLK)
/* IF USING THE V1.2 COMPILER, COMMENT THE FOLLOWING:
*/
INCLUDE LIBRARY(CBC.V3R1M0.SCBC3CMP)
     MODULES(CBC310PP,CBC310,CBC3P,CBC30,CBC3T,CBC3R,
             CBC30PTP)
/* IF USING THE V1.2 COMPILER, UNCOMMENT THE FOLLOWING:
     INCLUDE LIBRARY(EDC.V1R2M0.SEDCDCMP)
          MODULES(EDCDC120,EDCDP,EDCDO,EDCDT)
*/

/* THIS EXAMPLE ASSUMES THE FOLLOWING:
   SYS1.LINKLIB
   CEE.V1R5M0.SCEERUN
   CBC.V3R1M0.SCBC3CMP  ** IF USING THE V3.1 C/C++ COMPILER

  -OR-

  EDC.V1R2M0.SEDCDCMP  ** IF USING THE V1.2 C COMPILER

  ARE ALL APF AUTHORIZED AND CATALOGED.  THESE DATA SETS SHOULD ALSO
  BE INCLUDED IN THE LNKLSTXX MEMBER OF SYS1.PARMLIB THAT IS USED
  AT IPL TIME.  IF A JOBLIB OR STEPLIB IS USED INSTEAD OF LNKLSTXX,
  MAKE SURE THE MODULES LIST HERE ARE NOT INCLUDED IN THE JOBLIB
  OR STEPLIB OR YOU WILL LOSE THE BENEFIT OF MAKING THESE MODULES
  LPA ELIGIBLE.
*/
```

Figure 167. Sample IEALPAxx Member to Load Modules into ELPA

Step 3 Add libraries to Virtual Lookaside Facility (VLF)

VLF is designed primarily to improve performance by retrieving frequently used objects from virtual storage rather than performing repetitive I/O operations from DASD. VLF requires that a class statement be defined in parmlib member COFVLFxx for the class of objects you want VLF to cache. If you have the following class active you can then select members of the Library Lookaside Facility (LLA) to be cached:

```
CLASS NAME(CSVLLA)        /* Class name for Library Lookaside */
      EMAJ(LLA)           /* Major name for Library Lookaside */
```

Once VLF is active you can set up a CSVLLAxx member in SYS1.PARMLIB to specify which libraries (in addition to the LNKLST concatenation) library lookaside (LLA) is to manage.

Note: If running IBM AD/Cycle C/370 V1.2, 5688-216 then it will include data set, EDC.V1R2M0.SEDCDCMP if you have added EDC.V1R2M0.SEDCDCMP to your LNKLST concatenation.

If running IBM C/C++ for MVS/ESA V3.1, 5655-121 then it will include data set, CBC.V3R1M0.SCBC3CMP if you have added CBC.V3R1M0.SCBC3CMP to your LNKLST concatenation.

For more information on VLF and LLA see *MVS/ESA Initialization and Tuning Reference*. Member, CSVLLA00 in data set OMVS.PARMLIB shows an example of a customized CSVLLAxx member that can be used to cache data sets that are used in an OpenEdition MVS environment.

Step 4 Caching HFS data sets

Put the HFS data sets behind a DASD cache controller if possible and turn on the DFSMS/MVS DASD Fast Write (DFW) option.

Step 5 STEPLIB

Specify `export STEPLIB=NONE` in the etc/profile file to avoid excessive searching of STEPLIB data sets. For more on setting the STEPLIB environment variable, see *MVS/ESA OpenEdition MVS User's Guide* and *MVS/ESA Planning: OpenEdition MVS*.

Step 6 TCP/IP performance checks

There are a set of performance PTFs available for TCP/IP V3.1. These PTFs can significantly expand capacity, increase efficiency, and reduce overhead of TCP/IP, especially in high-volume Telnet, TN3270 and FTP environments. See informational APARs II08848 and II08849 for a description of these performance PTFs and some general TCP/IP performance tuning tips.

Step 7 NFS performance

The overall performance of the MVS NFS Server is affected by the performance of the client workstation, the network environment, the storage subsystem, the tuning of the MVS system, the TCP/IP parameters specified, as well as the MVS NFS Server parameters.

When TCP/IP and NFS are assigned to the same mean-time-to-wait dispatching priority in MVS/ESA, there can be times when NFS will have a higher dispatching priority than TCP/IP. This seems to affect NFS performance adversely. To correct this, assign fixed priorities to TCP/IP and NFS such that TCP/IP has a higher priority than NFS.

The nfstasks(n,m) statement defines the number of NFS tasks (or threads) to spawn. The n is the number of subtasks which

handle the asynchronous I/O operations or short blocking operations (the minimum number of concurrent NFSS requests). The m is the number of subtasks which handle the long blocking operations (the maximum number of concurrent NFSS recall and HFS requests.). Increase this value if your server supports lots of active recall or HFS clients, but the sum of n and m should not exceed 25.

On the client machine, depending on workload, increase the Basic I/O Daemons (BIODs). This setting determines the number of Remote Procedure Calls (RPCs) that can be executed in parallel in addition to the RPC call executed on behalf of the NFS client. The udp_sendspace and udp_receivespace values should be increased to 32768.

For more information on NFS tuning parameters, see *DFSMS/MVS Version 1 Release 3.0, Presentation Guide*.

5.4.1 Setting Up the REXX Parameter Modules

Among other things, the function call `syscalls('ON')` ensures that the SYSCALL host command environment is available in your REXX environment. If the call detects that SYSCALL is not available in your environment, it dynamically adds it.

Performance characteristics for dynamically added host commands are not as good as for host commands that are included in the initial environment. To get the best performance include the SYSCALL host command in the three default TSO/E environments as follows:

Run jobs IRXPARMS, IRXTSPRM and IRXISPRM in the OMVS.JCL data set. Review the JCL to fit you own installation's requirements.

For more information on customizing the default environments, refer to *TSO/E Procedures Language Reference for MVS/REXX*.

5.5 OpenEdition MVS Accounting

There are unique functions in OpenEdition MVS that need to be considered when doing accounting. The exec family of functions causes step termination and a new substep to be started. However, this new substep still has the same step number, but the substep number is incremented. Therefore, accounting applications must look for substep_number in addition to step_number, jobname and job_start_time.

There is a new macro, BPXESMF, to assist installations in collecting accounting information for address spaces that are OpenEdition MVS processes. This macro is found in SYS1.MACLIB.

5.5.1 Using RMF V5 with OpenEdition MVS

Resource Measurement Facility (RMF) Version 5, provides OpenEdition MVS report information using existing RMF records together with some new ones:

- Record type 42 subtype 6 provides information about data set level performance.

- Record type 74 subtype 3, provides data on kernel activity, numbers of users, processes and so forth.
- Record type 92 provides information about the file system activity, mount, demount, open, close, suspend and resume.

A new RMF report, OMVS kernel activity report, provides information on:

- OMVS system call activity
- OMVS process activity

5.5.1.1 RMF Monitor III

RMF invokes the Monitor III procedure RMFGAT to obtain OpenEdition MVS data. The RMFGAT started task must be associated with a user ID that has an OMVS segment, see 2.6.5, “Provide OpenEdition MVS Definitions for Other Resources” on page 51 and 2.6.6.2, “Define the OMVS Cataloged Procedure to RACF” on page 57 for more information.

Note: Gathering options for OMVS are not included in the default parmlib member for Monitor I, OMVS is gathered by Monitor III, and not by Monitor I.

The Monitor III data gatherer collects OpenEdition MVS data for input to the RMF post processor. This data can then be used to create the OMVS kernel activity report.

To create the OMVS kernel activity report the following JCL example, RMFPOST can be used. You can copy RMFPOST from the OMVS.JCL data set.

```
//ERBSAMPP JOB (ACCT),'PGMRNAME',CLASS=A,REGION=32M,NOTIFY=SLEKKA
//*******************************************************************
//*
//*01* MODULE-NAME: ERBSAMPP
//*
//*01* DESCRIPTIVE-NAME: SAMPLE RMF POST PROCESSOR JOB
//*
//*******************************************************************
//RMFSORT  EXEC PGM=SORT
//SORTIN   DD   DISP=SHR,DSN=SLEKKA.MVSOE.SMFDATA
//*        DD   DISP=SHR,DSN=<INPUT_SMFDATA_SYSTEM_2>
//*        :
//*        :
//*        DD   DISP=SHR,DSN=<INPUT_SMFDATA_SYSTEM_N>
//SORTOUT  DD   DISP=(NEW,PASS),UNIT=SYSDA,SPACE=(TRK,(200,200))
//SORTWK01 DD   DISP=(NEW,DELETE),UNIT=SYSDA,SPACE=(CYL,(10))
//SORTWK02 DD   DISP=(NEW,DELETE),UNIT=SYSDA,SPACE=(CYL,(10))
//SORTWK03 DD   DISP=(NEW,DELETE),UNIT=SYSDA,SPACE=(CYL,(10))
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//SYSIN    DD   *
  SORT FIELDS=(11,4,CH,A,7,4,CH,A),EQUALS
  MODS E15=(ERBPPSRT,500),E35=(ERBPPSRT,500)
//*        RMF POSTPROCESSING
//*
//RMFPP    EXEC PGM=ERBRMFPP
//MFPINPUT DD   DISP=(OLD,DELETE),DSN=*.RMFSORT.SORTOUT
//MFPMSGDS DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//SYSIN    DD   *
  SUMMARY(INT,TOT)
```

```
REPORTS(OMVS)
SYSOUT(A)
```

Figure 168 shows an example of this RMF report.

```
                                                                                                                     PAGE
                                                  O M V S   K E R N E L   A C T I V I T Y

           MVS/ESA                SYSTEM ID 3090          DATE 08/23/95          INTERVAL 10.34.000
           SP5.2.2                RPT VERSION 5.2.0       TIME 17.15.00          CYCLE 1.000 SECONDS
TOTAL SAMPLES =   633
                                                      OMVS SYSTEM CALL ACTIVITY
------------------------------------------------------------------------------------------------------------------------------

                 MINIMUM  AVERAGE  MAXIMUM
------------------------------------------------------------------------------------------------------------------------------

 SYSCALLS (N/S)   1.000    1.842    61.00
 CPU TIME (H/S)   0.000    0.003    1.000
                                                        OMVS PROCESS ACTIVITY
------------------------------------------------------------------------------------------------------------------------------

                         PROCESSES                       USERS                    PROCESSES PER USERS
 MAXIMUM  (TOT)             200                            50                             25
------------------------------------------------------------------------------------------------------------------------------

                 MINIMUM  AVERAGE  MAXIMUM      MINIMUM  AVERAGE  MAXIMUM      MINIMUM  AVERAGE  MAXIMUM
------------------------------------------------------------------------------------------------------------------------------

 CURRENT  (TOT)       7    9.848       10            0    0.000        0
 OVERRUNS (N/S)   0.000    0.000    0.000       0.000    0.000    0.000        0.000    0.000    0.000
                                                  OMVS INTER-PROCESS COMMUNICATION
------------------------------------------------------------------------------------------------------------------------------

                    MESSAGE QUEUE IDS              SEMAPHORE IDS              SHARED MEMORY IDS           SHARED MEMORY PAGES
 MAXIMUM  (TOT)              500                        500                         500                          262K
------------------------------------------------------------------------------------------------------------------------------

                 MINIMUM  AVERAGE  MAXIMUM    MINIMUM  AVERAGE  MAXIMUM    MINIMUM  AVERAGE  MAXIMUM    MINIMUM  AVERAGE  MAXIMUM
------------------------------------------------------------------------------------------------------------------------------

 CURRENT  (TOT)    0.000    0.000    0.000     0.000    0.000    0.000     0.000    0.000    0.000     0.000    0.000    0.000
 OVERRUNS (N/S)    0.000    0.000    0.000     0.000    0.000    0.000     0.000    0.000    0.000     0.000    0.000    0.000
                                                            OMVS MEMORY MAP
------------------------------------------------------------------------------------------------------------------------------

                 MEMORY MAP STORAGE PAGES       SHARED STORAGE PAGES
 MAXIMUM  (TOT)             4096                       131K
------------------------------------------------------------------------------------------------------------------------------

                 MINIMUM  AVERAGE  MAXIMUM    MINIMUM  AVERAGE  MAXIMUM
------------------------------------------------------------------------------------------------------------------------------

 CURRENT  (TOT)    0.000    0.000    0.000     0.000    2.379     1024
 OVERRUNS (N/S)    0.000    0.000    0.000     0.000    0.000    0.000
 Units:   (TOT) = Total Value, (N/S) = Number per Second, (H/S) = Hundredths of seconds per Second
```

Figure 168. Sample OpenEdition MVS RMF Kernel Activity Report

For a detailed description of the RMF OMVS system activity report, please refer to *RMF V5 User's Guide*.

Chapter 6. Maintaining OpenEdition MVS

This chapter will explain how you maintain and service the OpenEdition MVS product set.

6.1 HFS Data Set Recovery

Data in an HFS data set can be backed up and restored using one of three components that are a part of DFSMS/MVS:

- Hierarchical Storage Manager (DFSMShsm)
- Data Facility Data Set Services (DFSMSdss)
- ADSTAR Distributed Storage Manager (ADSM)

6.1.1 Backing Up and Restoring HFS Data Sets Using DFSMShsm

If you use DFSMShsm you must define a user ID for the DFSMShsm address space. In order for DFSMShsm to be able to access the HFS data sets, it must run under a user ID that is set up for OpenEdition MVS access:

- The default group for the DFSMShsm user ID must have an OMVS segment defined and a group ID associated with it.
- The home directory should be the root file system.
- The user ID should be defined as a superuser (with a UID of 0), or the DFSMShsm address space should be defined as TRUSTED in the RACF started procedures table. For more information see *DFSMS/MVS: Storage Administration Guide for DFSMSdss*.

6.1.2 Backing Up and Restoring HFS Data Sets Using DFSMSdss

The following JCL shown in Figure 169 on page 146 and Figure 170 on page 147 can be used to dump and restore an HFS data set. This JCL can be found as members DSSDUMP and DSSREST in data set OMVS.JCL.

```
//DSSDUMP  JOB  (999,POK),'HFS DUMP',MSGLEVEL=(1,1),
// CLASS=A,MSGCLASS=T,NOTIFY=SLEKKA
//*****************************************************************
//*
//* THIS JOB WILL CREATE A NEW HFS AND COPY THE EXISTING ROOT HFS
//* INTO THE NEWLY CREATED HFS.  THIS IS DONE BY USING THE
//* LOGICAL DUMP FUNCTION OF DFSMSDSS.
//*
//* EXISTING ROOT HFS: SPECIFY THE EXISTING ROOT HFS NAME AT THE DSN
//*                    PARM OF HFSVOL DD STATEMENT.
//*
//* NEW HFS NAME     : SPECIFY THE NEW HFS NAME AT THE DSN
//*                    PARM OF HFSOUT DD STATEMENT.
//*
//* STORCLAS         : SPECIFY YOUR STORAGE CLASS.
//*
//* OMVS.ROOT        : DSN OF THE EXISTING ROOT FILE SYSTEM
//*
//* OMVS.ROOT.SEQ    : DSN OF THE NEW ROOT FILE SYSTEM
//*
//*****************************************************************
//SU       EXEC  PGM=ADRDSSU,REGION=6M
//SYSPRINT DD  SYSOUT=*
//HFSVOL   DD  UNIT=3390,VOL=SER=OP2HFS,DISP=SHR
//HFSOUT   DD  DSN=OMVS.ROOT.SEQ,
//             SPACE=(CYL,(100,100),RLSE),
//             UNIT=SYSALLDA,STORCLAS=OPENMVS,
//             DISP=(NEW,CATLG,DELETE)
//SYSIN    DD  *
 DUMP DATASET(INCLUDE(OMVS.ROOT))   -
      COMPRESS TOL(ENQF) -
      LOGINDDNAME(HFSVOL) OUTDDNAME(HFSOUT) ALLDATA(*) ALLEXCP
/*
```

Figure 169. JCL Used to Dump an HFS Data Set

```
//DSSREST JOB (999,POK),'HFS RESTORE',NOTIFY=SLEKKA,
//    CLASS=A,MSGCLASS=T,TIME=1439,
//    REGION=5000K,MSGLEVEL=(1,1)
//*****************************************************************
//*
//* THIS JOB WILL RESTORE AN HFS DATA SET THAT WAS PREVIOUSLY
//* DUMPED USING DFDSS.  A RENAMEU PARAMETER IS USED
//* TO RENAME THE HFS DATA SET BEING RESTORED AS NOT TO
//* CONFLICT WITH AN EXISTING HFS DATA SET ON THE PACK.
//*
//* HFSSEQ           : SPECIFY THE SEQUENTIAL BACKUP DATA SET THAT
//*                    CONTAINS THE PREVIOUSLY BACKED UP HFS
//*                    DATA SET TO THE
//*                    HFSSEQ DD STATEMENT.
//*
//* HFSOUT           : SPECIFY THE PACK WHERE THE RESTORED HFS DATA
//*                    SET WILL RESIDE.
//*
//* RENAMEU PARM     : SPECIFY THE OLD HFS NAME AND THE NEW HFS
//*                    NAME, USING THE RENAMEU PARM.
//*
//* STORCLAS         : SPECIFY YOUR STORAGE CLASS.
//*
//*****************************************************************
//STEP1    EXEC PGM=ADRDSSU,REGION=6M
//SYSPRINT DD   SYSOUT=*
//HFSSEQ DD DSN=OMVS.ROOT.SEQ,DISP=SHR,
//       VOL=SER=OP2HFS,UNIT=3390
//HFSOUT   DD   UNIT=3390,VOL=SER=OP2HFS,DISP=SHR
//SYSIN    DD   *
 RESTORE  INDD(HFSSEQ) OUTDD(HFSOUT) TOL(ENQF) -
 DATASET(INCLUDE(OMVS.ROOT)) -
 RENAMEU((OMVS.ROOT,OMVS.SERVICE.ROOT)) -
 STORCLAS(OPENMVS)        -
 CANCELERROR
/*
```

Figure 170. JCL Used to Restore an HFS Data Set

6.1.2.1 Increasing the Size of an Existing HFS Data Set

At some point it may be necessary to increase the size of an existing HFS data set if it has run out of extents. To increase the size:

1. Use the JCL in Figure 169 on page 146 to dump the HFS data set.
2. Preallocate a larger HFS data set with the same name as the old smaller HFS data set.
3. Use the JCL in Figure 170 to restore the HFS data set replacing the RENAMEU keyword with the REPLACE keyword as follows:

```
RESTORE  INDD(HFSSEQ) OUTDD(HFSOUT) TOL(ENQF) -
DATASET(INCLUDE(OMVS.ROOT)) -
STORCLAS(OPENMVS)        -
REPLACE                  -
CANCELERROR
```

For more information on the REPLACE keyword, see *DFSMSdss Storage Administration Reference*.

6.1.3 Backing Up and Restoring Files Using ADSM

ADSM is discussed in 7.3.1, “Customizing the ADSM Clients for OpenEdition MVS” on page 175.

6.2 Installing Service

There is no single recommended way of updating a system with service. All MVS users have different configurations, product sets, business requirements, support personnel, and so on. In other words, every user is unique, and a recommendation that would work well for one user may be unworkable for another. It would be beyond the scope of this book to explore all aspects of managing software on an MVS/ESA system, but a general methodology of how one might manage maintenance when the OpenEdition MVS product is involved is discussed.

6.2.1 Installing Service to Products in the HFS

Service for OpenEdition MVS is installed in much the same way as it is installed for traditional MVS/ESA products. Service is applied using System Modification Program Extended (SMP/E). Differences include:

- The target libraries for OpenEdition MVS are in DFSMS/MVS managed data sets called HFS data sets. DFSMS/MVS must be up and running and managing at least one DASD volume; the volume that contains the OpenEdition MVS target libraries.
- The OpenEdition MVS kernel must be running on the driving system in order to apply service to HFS data sets.
- The user ID that is running the SMP/E jobs must have an OMVS security segment and they must be running as a superuser (UID=0).

In a typical MVS/ESA installation, an active, production-level system is called a *driving system*. This system could be used to run business critical applications or it could be an LPAR carved out of a larger processor used for system build and testing activity. You should never directly upgrade your driving system. Rather, you should install changes on a copy or clone of your production system, called a *target system*, test the changes, and then migrate the changes into your production environment. This process minimizes the risk of new code causing an outage to your production system. Figure 171 on page 149 shows the process of cloning the OMVS.ROOT and OMVS.ADSM HFS data sets.

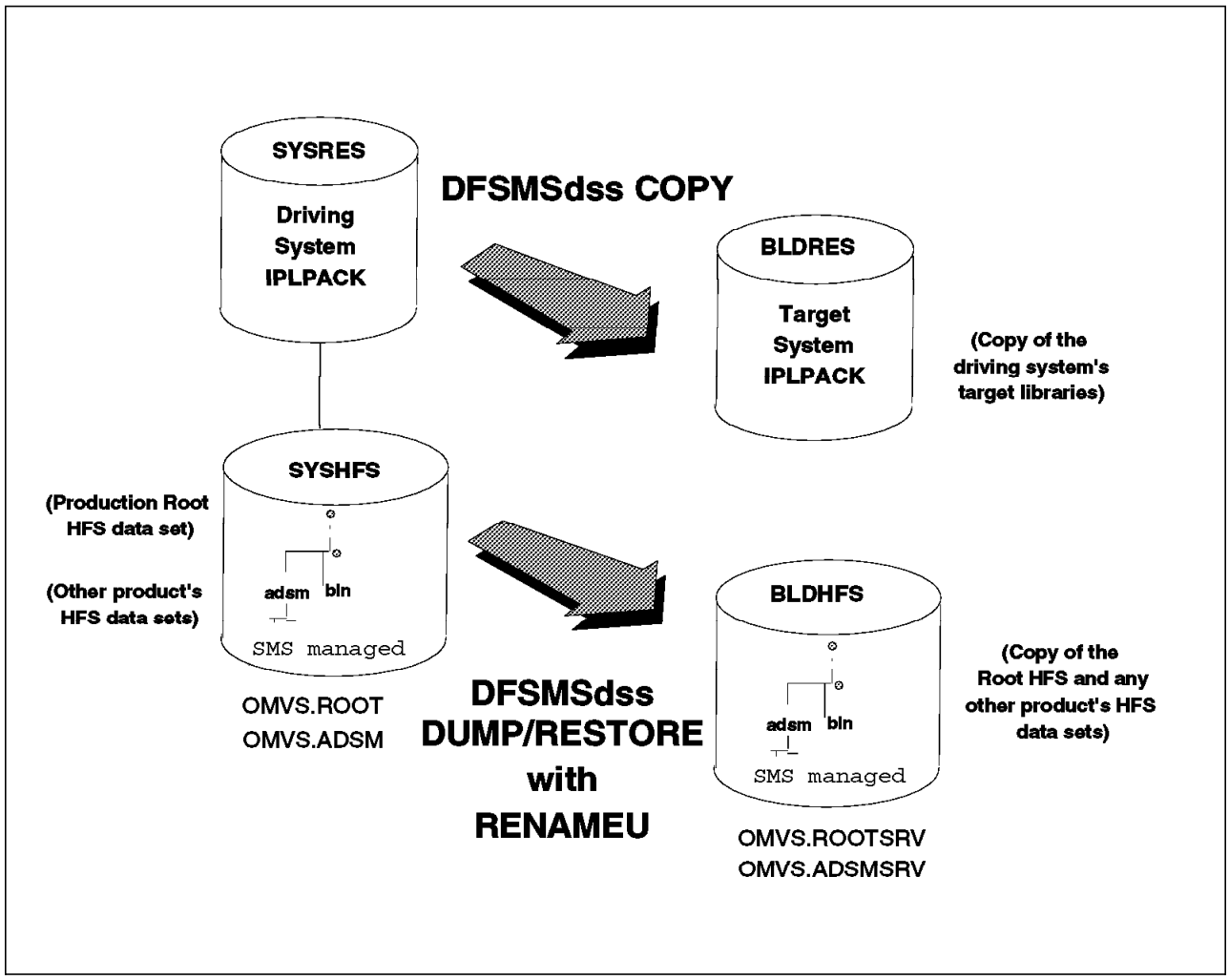


Figure 171. Make a Clone of Your System Using DFSMSdss

The following steps could be followed to apply service to OpenEdition MVS.

Step 1 Build a clone of your production system using DFSMSdss

In our example, the traditional product target libraries are copied to a new pack called BLDRES. The OpenEdition MVS and any related product target libraries are also copied, but they must be copied to a pack that is SMS managed (BLDHFS). JCL examples that show how to dump and restore the OpenEdition MVS root HFS data set using DFSMSdss can be found as members DSSDUMP and DSSREST respectively in data set OMVS.JCL.

Note: The only way currently to copy HFS data sets is to use the DUMP and RESTORE DFSMSdss commands. The RENAMEU parameter is used to rename and catalog the copied HFS data sets to eliminate any problems with duplicate data set names while running in an DFSMS/MVS environment.

Step 2 The new copied root file system has to be mounted at a directory within the active file hierarchy. To be able to mount the new HFS a mount point directory has to be created. Our

mount point directory is called SERVICE and is a subdirectory of the root. The SERVICE mount point directory is created by using the mkdir command from the shell or by using the MKDIR command from TSO. The mount itself can be done by a superuser in two ways:

Using the 'mount' command from the ISHELL:

Enter the ISHELL. Select File_systems from the action bar. Select Mount option from pull-down menu. You are now prompted to
enter the file system name and the mount point.

OR

Issuing the TSO/E MOUNT command:

```
mount filesystem('OMVS.ROOTSRV') mountpoint('/SERVICE')
mode(rdwr) type(hfs)
```

If you have other products that are under SMP/E control and use HFS data sets for their target libraries, you will need to mount those HFS data sets off the /SERVICE directory as follows:

Issuing the TSO/E MOUNT command for other HFS data sets:

```
mount filesystem('OMVS.ADSMSRV') mountpoint('/SERVICE/adsm')
mode(rdwr) type(hfs)
```

The df command (display file-system) issued from the shell will check to see that the mount command was done properly.

Step 3 Change SMP/E DDDEFs

The next step is to change the DDDEF's or DD statements used by SMP/E to point to the new target directories that are now mounted off the /SERVICE directory. For example, DDDEF entry SFSUMBIN when first customized, points to /bin/IBM. We now want to apply service to the cloned data sets that are mounted off the /SERVICE directory so we have to go in and change the path to where the SFSUMBIN DDDEF is pointing to /SERVICE/bin/IBM.

Check the program directories of all products that install their elements in HFS data sets for the list of DDDEF names and their associated paths that need to be changed. For OpenEdition MVS, members FSUMUSDD, FDBXUSDD and FOMISDDD of SYS1.SAMPLIB contain the list of all OpenEdition MVS DDDEFs that point to the HFS file system. All paths need to be changed to have the /SERVICE prefix. For more information on changing DDDEFs see *SMP/E R8.1 Reference*.

Step 4 Install Service with SMP/E

After the DDDEFs have been changed, service can be installed with SMP/E. Figure 172 on page 151 shows the new cloned HFS data sets, OMVS.ROOTSRV and OMVS.ADSMSRV are mounted at the /SERVICE mount point. The OpenEdition MVS kernel address space must be active on the driving system because the special read and write functions required by SMP/E to install the

OpenEdition MVS and related product elements in the HFS data sets are only supported when it is up and running. As was mentioned earlier, the user ID that is running the SMP/E jobs to apply service to the HFS file system, must have superuser authority (UID=0).

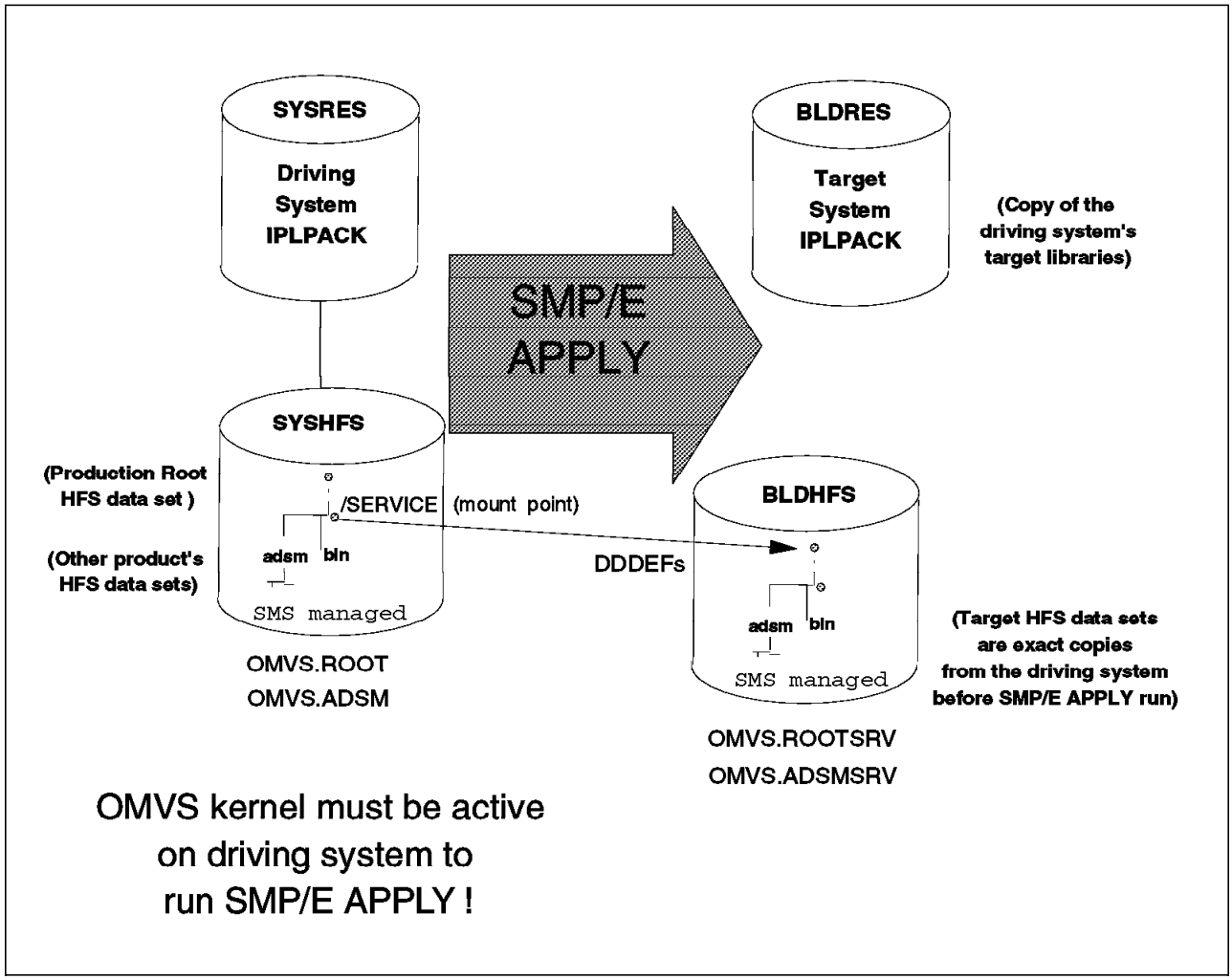


Figure 172. SMP/E APPLY of OpenEdition MVS Service

Step 5 Testing

After installing the service, the new OpenEdition MVS target libraries should be tested. The level of testing depends on the amount of service and what OpenEdition MVS components were hit with service. Check the DDDEF list in your SMP/E APPLY listing for the data sets that were changed when service was applied. You know if service hits SYS1.LPALIB, you have to IPL the BLDRES pack as shown in Figure 172 with a Clear Link Pack Area (CLPA). If service only hits paths in the HFS file system you most likely only have to change the BPXPRMxx member to point to the new cloned HFS data sets, (OMVS.ROOTSRV and OMVS.ADSMSRV in our example) and stop and start the OpenEdition MVS kernel. If in doubt, IPL the BLDRES pack with a CLPA, change the BPXPRMxx member to point to the new

cloned HFS data sets and restart the OpenEdition MVS kernel. See Chapter 4, “Installation Verification of OpenEdition MVS” on page 113 for information on running Installation verification.

Note: There can only be one OpenEdition MVS kernel running at one time on any one system. If you are going to test the newly serviced OpenEdition MVS system on the same processor or LPAR as the driving system, you must IPL off the BLDRES (only if data sets like SYS1.LPALIB and SYS1.LINKLIB were hit with service) and BLDHFS packs as in our example. It is not possible to have the OpenEdition MVS kernel use STEPLIB to get to any files in the new HFS data sets. The BPXPRMxx member of parmlib must be updated with the names of the new HFS data sets and the OpenEdition MVS kernel must be restarted. If the test has been successfully completed this new cloned system can now be propagated to other systems in your enterprise using the same DFSMSdss methods as were discussed earlier in this chapter. Remember that the HFS data sets are now part of your overall cloning and system propagation scheme and must be copied to other systems along with the traditional target libraries, if you plan to run OpenEdition MVS on any other systems in your enterprise.

Chapter 7. Other Licensed Programs that Interact with OpenEdition MVS

OpenEdition MVS provides a set of services within MVS and it also interacts with other licensed programs.

TCP/IP	User written socket applications can use TCP/IP as a communication vehicle. Both client and server socket applications can use the OpenEdition MVS socket interface to communicate over the network (AF_INET) and between other OpenEdition MVS socket applications by using local sockets (AF_UNIX). TCP/IP is the transport provider when using rlogin from a UNIX workstation to log in directly to the OpenEdition MVS shell.
NFSS	With the DFSMS/MVS Network File System feature, workstation users can mount all or part of their OpenEdition MVS file system and make it appear as part of their local file system. With their file system mounted, they can use NFS to create new and delete old, OpenEdition MVS files and directories and read and write against existing OpenEdition MVS files and directories.
ADSM	With ADSTAR Distributed Storage Manager, OpenEdition MVS files can be automatically backed up and archived based on policies that are set by the OpenEdition MVS administrator or MVS systems programmer.

7.1 Customizing and Starting TCP/IP

TCP/IP V3R1 for MVS provides networking support for OpenEdition MVS. It would go beyond the scope of this book to detail TCP/IP base installation. The installation of TCP/IP V3R1 for MVS is documented in product documentation and in *IBM TCP/IP V3R1 for MVS Implementation Guide*. *TCP/IP Tutorial and Technical Overview* contains a description of TCP/IP concepts and protocols.

7.1.1 Setting Up a Simple TCP/IP Environment

If you do not have TCP/IP customized on your system, you will have to customize it and get it running for OpenEdition MVS. The following steps will help you in preparing a mini TCP/IP environment in order to run OpenEdition MVS.

Step 1 Choose a High Level Qualifier

TCP/IP V3R1 for MVS uses dynamic allocation for several parameter data sets. The data set name search sequence in many cases relies on accepted High Level Qualifiers (hlq). Valid hlqs for example are a job name, node name, and user ID. We recommend to use the DATASETPREFIX items in the TCP/IP profile data set and the TCP/IP client parameter file to specify a hlq for your installation. We used this method and specified TCPIP as the hlq. For more information on hlq rules see *IBM TCP/IP for MVS Customization and Administration Guide*.

Step 2 Install TCP/IP

Run the installation procedures as described in the TCP/IP Program Directory.

Step 3 Customize the TCP/IP procedure

Copy the sample TCP/IP procedure from *hlq*.SEZAINST(TCPIPROC) to your procedure library and adjust the data set names in the DD statements. The TCPIP procedure that we used can be found in data set, OMVS.PROCLIB as member TCPIP.

Step 4 Define the required RACF profiles for the TCP/IP started tasks.

See 2.6.5, “Provide OpenEdition MVS Definitions for Other Resources” on page 51 for more information.

Step 5 Customize the TCP/IP profile data set

Copy *hlq*.SEZAINST(SAMPPROF) to *hlq*.PROFILE.TCPIP and make the necessary changes. Depending on what kind of device you are using to connect your MVS system to the network, and depending on the services you want to provide you can comment most statements in the sample. We have provided an example of a TCP/IP profile data set which is customized for a 3172 Token-Ring LAN connection and other TCP/IP services on the diskette as file name TCPIP.PRF. Be sure to rename this data set when you copy it off the diskette onto your MVS machine. We named it: TCPIP.PROFILE.TCPIP.

Step 6 Customize the TCP/IP client parameter file

Copy *hlq*.SEZAINST(TCPDATA) to *hlq*.TCPIP.DATA and adjust it to your environment. As long as you are not using a nameserver it should only contain the host name and the name of the TCP/IP transport provider started task. We have provided an example of a TCP/IP client parameter file on the diskette as file name TCPIP.DTA. Be sure to rename this data set when you copy it off the diskette onto your MVS machine. We named it: TCPIP.TCPIP.DATA.

Step 7 Customize hosts site table data sets *hlq*.HOSTS.LOCAL, *hlq*.HOSTS.SITEINFO and *hlq*.HOSTS.ADDRINFO.

The *hlq*.HOSTS.LOCAL data set contains the internet addresses and corresponding host names. A sample data set is provided with the installation. It has to be customized to reflect your environment and it is used as an input file to the TCP/IP MAKESITE command to build the *hlq*.HOSTS.SITEINFO and *hlq*.HOSTS.ADDRINFO data sets. These data sets are used by MVS clients to resolve IP addresses from host names if there is no name server available. The NFS Server, on MVS uses them to derive client names from IP addresses. Client names are used for NFS authorization checking. Other commands like

HOMETEST and NETSTAT CONN also use the host site table data sets. We have provided an example of a *hlq*.HOSTS.LOCAL file on the diskette as file name TCPIP.HOS. Be sure to rename this data set when you copy it off the diskette onto your MVS machine. We named it: TCPIP.HOSTS.LOCAL.

Step 8

Customize the *hlq*.ETC.SERVICES, *hlq*.ETC.PROTO and *hlq*.ETC.RPC data sets.

The TCP/IP V3R1 for MVS product comes with the following data sets: *hlq*.ETC.SERVICES, *hlq*.ETC.PROTO and *hlq*.ETC.RPC. They are used by socket and RPC calls. In most cases you don't need to change these data sets, but it is important that the *hlq* fits the naming conventions you chose in *Step 1*. See *MVS/ESA Planning: OpenEdition MVS* for more information on TCP/IP V3R1 for MVS data set name rules with OpenEdition MVS.

Step 9

Customize the OpenEdition MVS BPXPRMxx parmlib member. Copy the following in Figure 173 into your BPXPRMxx member to activate TCP/IP support for a single transport provider. An example is also provided in OMVS.PARMLIB member BPXPRMT1.

```
/* The following definitions reflect the        */
/* environment with one TCP/IP started task     */
/*                                              */
/* Parameters for Integrated Socket support     */

   FILESYSTYPE TYPE(INET) ENTRYPOINT(BPXTIINT)
   NETWORK DOMAINNAME(AF_INET)
           DOMAINNUMBER(2)
           MAXSOCKETS(10000)
           TYPE(INET)
```

Figure 173. BPXPRMxx Entries for a Single TCP/IP Transport Provider

Step 10

Start the TCPIP started task and do some connectivity tests using TSO client commands.

The following TSO commands are samples you could use to check that the steps you did up to now were successful.

- HOMETEST
- NETSTAT DEVLINKS
- TESTSITE
- PING <hostname>

Step 11

Start OMVS

If you start TCPIP when OMVS is up you should get messages like the following:

```
EZY2140I OpenEdition-TCP/IP connection established for TCPIP
```

Figure 174. MVS Console Output after TCP/IP and OpenEdition MVS Connect

If you issue the TSO NETSTAT command now you still will not see a socket connected to the TCPIP started task. This is because the OpenEdition MVS inetd daemon is not started. The inetd daemon is discussed in 7.1.3, “Customizing for inetd and rlogind Daemons” on page 165.

The following figure shows all the the started tasks and OE processes we used in our simple TCP/IP environment. The names in the boxes reflect the names of the started tasks.

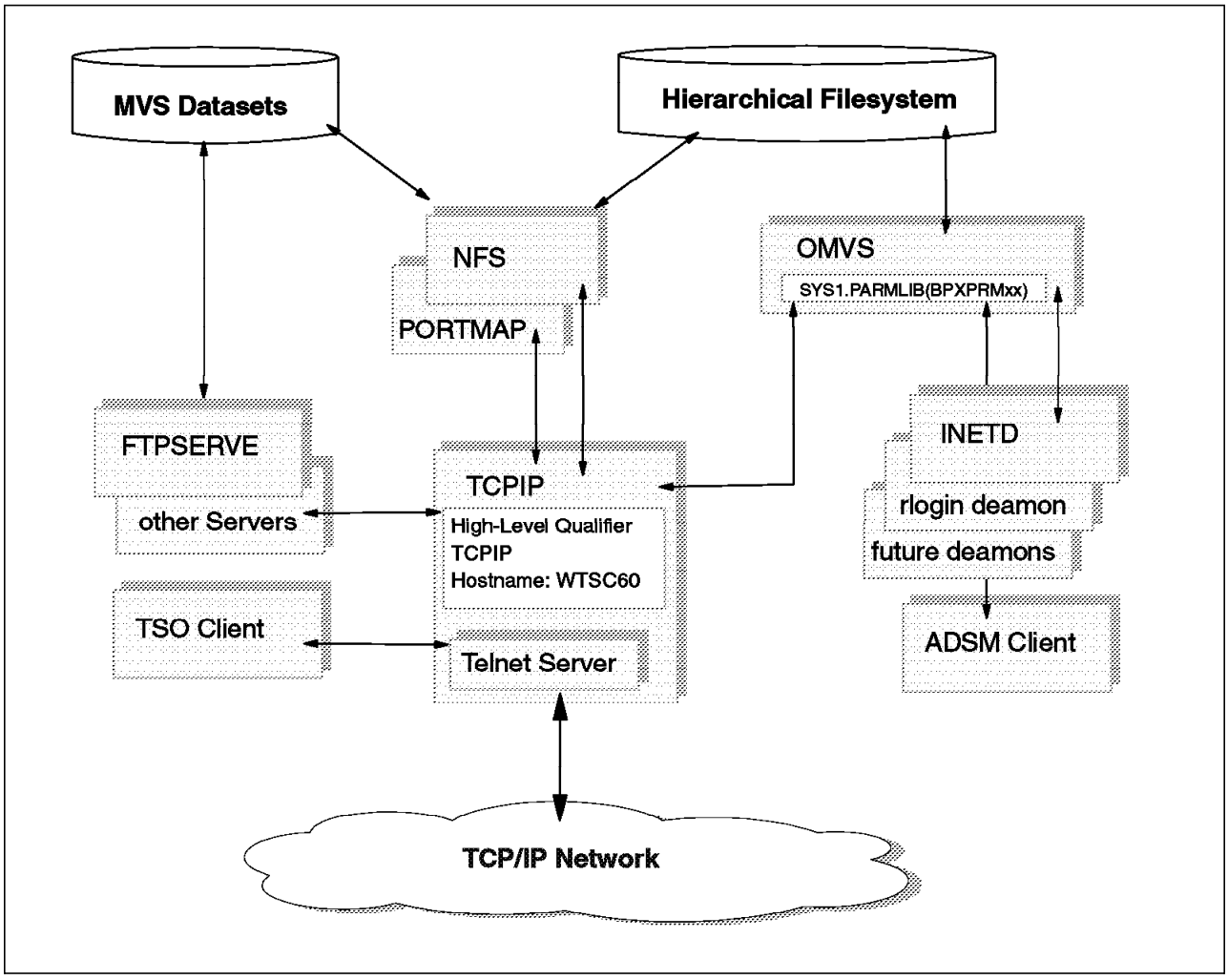


Figure 175. Overview of Server and Client Started Tasks in a Simple TCP/IP Environment

With the customization steps above, you should be able to use the TSO client, and to start up the TCPIP and the OMVS started tasks. A description on the NFS and PORTMAP started tasks can be found in 7.2, “Customizing and Starting NFS” on page 167. See 7.1.3, “Customizing for inetd and rlogind Daemons” on page 165 on how to customize the inetd daemon.

See 7.3, “ADSM Client for OpenEdition MVS” on page 175 on how to customize the ADSM client.

We used the FTP server for file transfer to/from traditional MVS data sets. See *IBM TCP/IP for MVS Customization and Administration Guide* on further information on how to set up the started task for the FTP server.

7.1.2 Setting Up the Multiple Transport Driver Support

If TCP/IP V3R1 for MVS is already installed in your system, it will in many cases be useful to set up a separate TCP/IP started task for OpenEdition MVS services, for example:

- It is easier to separate a production and test environment with two TCP/IP started tasks.
- Some future OpenEdition MVS server applications will use the same port numbers as traditional MVS servers. For example, the TCP/IP Internal Telnet server uses port 23 for TN3270 support, whereas OpenEdition MVS will use this port for line mode and raw-mode terminal support. Although you can override these well-known port numbers, it may not be convenient for all environments.
- Although one single MVSNFS server can handle both access to HFS files and access to traditional MVS data sets, you might want to use different security schemes on HFS and MVS data sets which is not possible with a single server. In fact, as long as you can trust your NFS client system administrators, EXPORTS security might be appropriate for HFS file access and SAF security for MVS data set access. See 7.2.2, “Security Settings for the MVS NFS Environment” on page 170 for more information.
- With the converged socket subfile support, it is now possible to have more than one transport provider to OpenEdition MVS. A transport provider can be a TCP/IP started task or the Anynet feature of ACF/VTAM V4.3.

One crucial step in implementing multiple TCP/IP transport providers is to decide on a naming convention for parameter data sets. Naming conventions for TCP/IP V3R1 for MVS are described in *IBM TCP/IP for MVS Customization and Administration Guide*; how naming conventions affect OpenEdition MVS is described in *MVS/ESA Planning: OpenEdition MVS*.

In implementing two TCP/IP transport providers we tried to:

1. Avoid redundancy. Where possible, we were able to share TCP/IP definition data sets for all transport providers.
2. Assign one default transport provider for OpenEdition MVS services and the other as a default transport provider for traditional MVS purposes.
3. Provide flexibility. In principle, every transport provider started task should be able to serve traditional MVS client and server processes as well as OpenEdition MVS processes.

We implemented two transport providers. The names of the started tasks are TCPIPMVS (to be the default transport provider for traditional MVS processes) and TCPIPOE (to be the default transport provider for OpenEdition MVS processes). Both tasks were connected via IUCV, a communication method used for TCP/IP communication within one physical MVS system. To ease network management, we created for both systems RouteD daemons that provide RIP (Routing Information Protocol) support. DNS (Domain Name Server) support was provided by another system in our network.

Dynamic allocation of sequential MVS data sets seems to be the only way to provide parameter definitions for both traditional MVS processes and OpenEdition MVS processes. Explicit allocation using DD cards, does not work with forked OpenEdition MVS processes. HFS files can not be used by traditional MVS processes based on current BSAM/QSAM access methods.

With this in mind we chose the following naming conventions:

1. All parameter data sets, to be used by both transport providers have a high level qualifier of TCPIP.
2. All parameter data sets, to be used by TCPIPMVS and the related client and server processes have a high level qualifier of TCPIPMVS.
3. All parameter data sets, to be used by TCPIPOE and the related client and server processes have a high level qualifier of TCPIPOE.
4. Explicit allocation using DD statements are used for traditional MVS processes either to find TCPIPMVS related parameter data sets or process specific parameter data sets.
5. The HFS /etc/inetd.conf file and environment variables are used for OpenEdition MVS processes either to find TCPIPOE related parameter data sets or to set process specific parameters.

By adhering to the above naming conventions we were able to share the following data sets between the TCPIPMVS and TCPIPOE transport providers.

- TCPIP.ETC.PROTO, TCPIP.ETC.RPC, and TCPIP.ETC.SERVICES
- TCPIP.HOSTS.LOCAL, and the corresponding files TCPIP.HOSTS.ADDRINFO and TCPIP.HOSTS.SITEINFO
- TCPIP.STANDARD.TCPXLBIN

For each transport provider we provided separate TCP/IP profile and TCP/IP client data sets, *hlq*.PROFILE.TCPIP and *hlq*.TCPIP.DATA respectively.

The status of a TCP/IP transport provider is monitored with the TSO NETSTAT command. You can specify the name of the transport providers started task to be monitored if you have multiple transport providers installed.

Configuration changes to the TCP/IP transport providers during run time are issued with the TSO OBEYFILE command. The following small REXX exec was used to ease the operating of multiple transport providers. This exec can be used to start or stop devices or to switch traces on or off. A copy of this exec can be found as member OBEY in data set OMVS.JCL.

```
/*REXX*/
/*JUST TO SIMPLIFY TCPIP OPERATING*/
trace o;
arg tp cmd;
pref = left(tp,5)
"FREE FI(SYSTCPD)"
"FREE FI(OBEYCMD)"
Select
   when pref = TCPIP
   then do
      "ALLOC FI(SYSTCPD) da('"tp".TCPIP.DATA') SHR"
      if RC \= 0 then do
         say 'Can''t allocate SYSTCPD'
         exit
         end
      end
   otherwise do
      say 'Syntax is: OBEY <TCPIP-STC name> <TCPIP Config. statement>'
      say 'TCPIP-STC name must start with ''TCPIP'' '
      exit
      end
   end;
"ALLOC FI(OBEYCMD) da('TCPIP.OBEY.CMD') OLD REUSE"
queue cmd
"EXECIO 1 DISKW OBEYCMD (FINIS"
"FREE FI(OBEYCMD)"
"OBEYFILE 'TCPIP.OBEY.CMD' "
exit
```

The following figure gives an overview of the started tasks and OpenEdition MVS processes we used in our installation. The names in the boxes reflect the names of the started tasks in our environment.

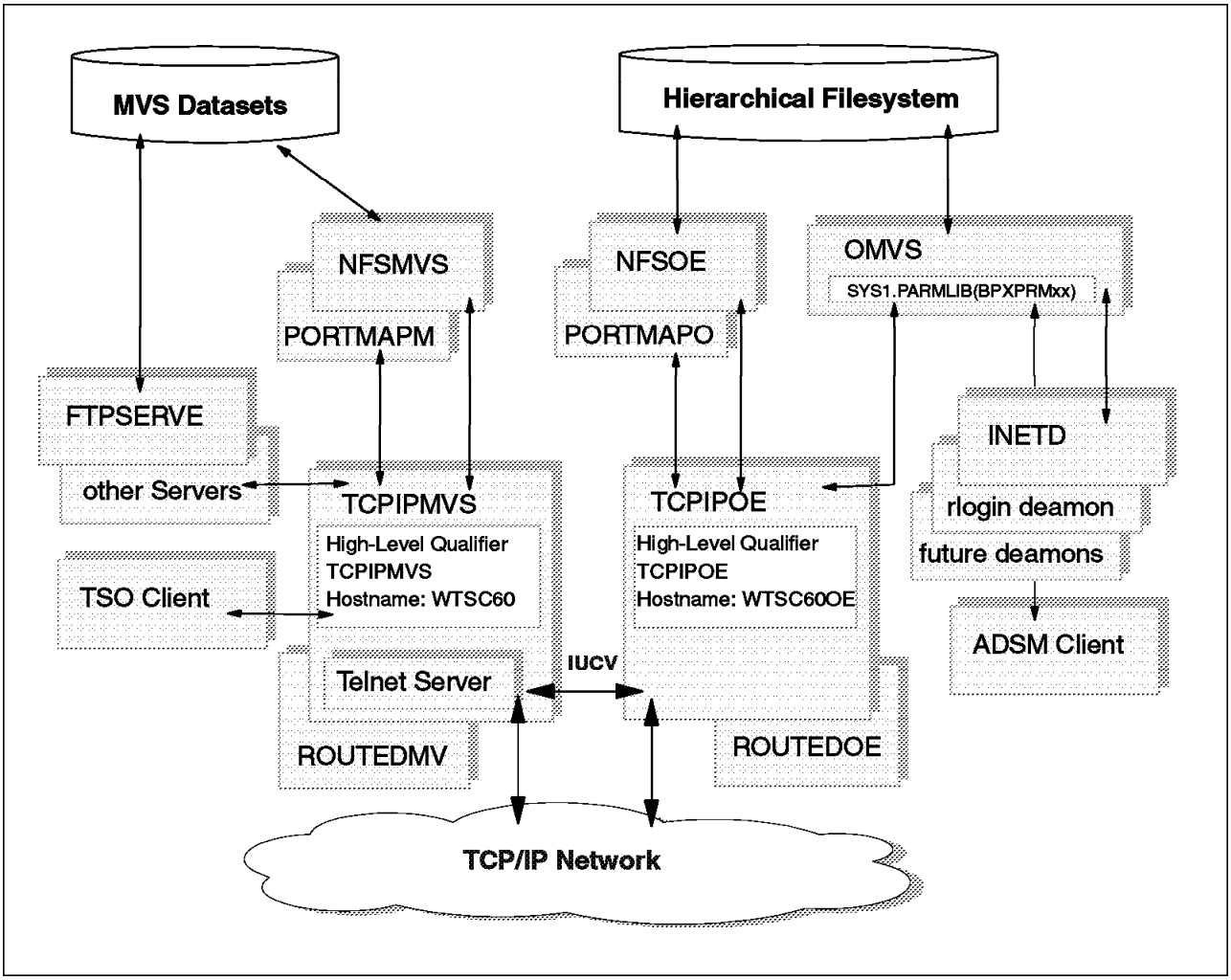


Figure 176. Overview of Server and Client Started Tasks Using Multiple TCP/IP Transport Providers

Notes:

1 The drawing shows both transport providers connected to the TCP/IP network (vertical arrows) and interconnected via IUCV. This configuration seems to be reasonable for performance and backup reasons. Unfortunately, as of now some LAN connections cause problems when connected to different TCP/IP transport providers. See APAR PN67698 for more information. When PN67698 has been solved you could for example use a channel attached IBM RS/6000 system to provide network connectivity to OpenEdition MVS and an IBM 3172 to provide network connectivity to traditional MVS services.

2 There can be more than one transport provider for OpenEdition MVS. This function called Converged Socket support is sometimes also referred to as Common INET support. To select the converged socket support you must specify:

```
FILESYSTYPE TYPE(CINET) ENTRYPOINT(BPXTCINT)
```

in the BPXPRMxx member of SYS1.PARMLIB instead of

```
FILESYSTYPE TYPE(INET) ENTRYPOINT(BPXTIINT)
```

APAR PN69843 provides the support for converged sockets with TCP/IP V3R1 for MVS. The AnyNet feature announced with VTAM V4R3 for MVS/ESA provides the support for converged sockets with ACF/VTAM.

To implement a system like pointed out in Figure 176 on page 161 you have to go through the following steps.

Step 1 Plan your naming conventions.

Step 2 Define the transport provider started tasks and RouteD daemons, and interconnect them via IUCV.

This is done by creating procedures TCPIPMVS and TCPIPOE as transport providers and ROUTEDMV and ROUTEDOE for routing services in your procedure library and, providing unique TCP/IP profile and TCP/IP client data files. We have provided the procedures for the started tasks on the diskette, as member names TCPIPMVS, TCPIPOE, ROUTEDMV, and ROUTEDOE in data set OMVS.PROCLIB. The TCP/IP profile and TCP/IP client data files can be found on the diskette as file names TCPIPMVS.PRF and TCPIPMVS.DTA for TCPIPMVS and TCPIPOE.PRF and TCPIPOE.DTA for TCPIPOE. After these files are unloaded off the diskette we used data set names TCPIPMVS.PROFILE.TCPIP and TCPIPMVS.TCPIP.DATA for the TCPIPMVS started task and TCPIPOE.PROFILE.TCPIP and TCPIPOE.TCPIP.DATA for the TCPIPOE started task.

Remember, that we used a name server on another TCP/IP host in our network. This is reflected in the TCP/IP client data files.

Step 3 Define the required RACF profiles for the started tasks.

See 2.6.5, “Provide OpenEdition MVS Definitions for Other Resources” on page 51 for related information.

Step 4 Customize your name resolution environment.

- If you use a name server for name resolution, reflect your transport providers in the name server database.
- If you use hosts site tables for name resolution, reflect the names and IP addresses of foreign IP-hosts in your *hlq*.HOSTS.LOCAL data set. Use the TCP/IP MAKESITE command to compile the *hlq*.HOSTS.LOCAL data set into the *hlq*.HOSTS.SITEINFO and *hlq*.HOSTS.ADDRINFO data set. Make sure, that the name and IP-addresses of your transport providers are also entered in the name tables of other hosts.

  We used a mixed environment with host tables and a name server. Parts of our network are reflected in the TCPIP.HOSTS.LOCAL file found on the diskette as TCPIP.HOS.

Step 5 Customize the *hlq*.ETC.SERVICES, *hlq*.ETC.PROTO, and *hlq*.ETC.RPC data sets.

The TCP/IP V3R1 for MVS product provides the following data sets: *hlq*.ETC.SERVICES, *hlq*.ETC.PROTO and *hlq*.ETC.RPC. They are used by socket and RPC calls. You usually don't have to change the contents of these data sets. Depending on the naming conventions you choose you might have to provide multiple copies of these data sets using different high level qualifiers or you will have to change the related DD statements or HFS files. See *MVS/ESA Planning: OpenEdition MVS* for more information on TCP/IP V3R1 for MVS data set names with OpenEdition MVS. With the naming conventions we choose, we could use the default data sets even with multiple TCP/IP transport providers

Step 6 Add a TCPDATA DD statement in your TSO logon procedure. In our environment it looked like:

```
//SYSTCPD  DD  DSN=TCPIPMVS.TCPIP.DATA,DISP=SHR
```

Note: The //SYSTCPD DD statement points to the TCP/IP MVS client parameter file. The default DSN would be TCPIP.TCPIP.DATA or SYS1.TCPPARMS(TCPDATA). If you do not use one of these defaults you have to add a SYSTCPD DD statement in the TSO logon procedure in order to use TCP/IP client functions and some administrative functions like (OBEYFILE) under TSO. TCPIPMVS.TCPIP.DATA is the parameter data set that we used so we had to add the //SYSTCPD DD statement to our logon procedure.

Step 7 Start your transport providers and do some connectivity tests using TSO client commands.

The following commands are samples you could use to test that the steps you did up to now were successful:

- HOMETEST
- NETSTAT DEVLINKS TCP <transport provider name>
- TESTSITE if you use host tables
- NSLOOKUP <hostname> if you use a nameserver
- PING <hostname>

Step 8 Customize the OpenEdition MVS BPXPRMxx parmlib member.

The RACF user ID associated with the started task name of the transport provider for OpenEdition MVS has to be reflected in the NAME option

```
SUBFILESYSTYPE NAME(racf user ID)
TYPE(CINET)
ENTRYPOINT(BPXTIINT)
DEFAULT
```

in the BPXPRMxx parmlib member. Copy the following in Figure 177 on page 164 to your BPXPRMxx member to activate TCP/IP support for multiple transport providers. An example is also provided in OMVS.PARMLIB member BPXPRMT2.

```
/* The following definitions reflect the          */
/* environment with multiple TCP/IP started tasks */
/*                                                */
/* Parameters for Common Internet Socket support   */


  FILESYSTYPE TYPE(CINET) ENTRYPOINT(BPXTCINT)
  SUBFILESYSTYPE NAME(TCPIPOE)        1
                 TYPE(CINET)
                 ENTRYPOINT(BPXTIINT)
                 DEFAULT

  SUBFILESYSTYPE NAME(LINET)          2
                 TYPE(CINET)
                 ENTRYPOINT(BPXTLINT)

  NETWORK DOMAINNAME(AF_INET)
          DOMAINNUMBER(2)
          MAXSOCKETS(10000)
          TYPE(CINET)
          INADDRANYPORT(4901)
          INADDRANYCOUNT(100)
```

Figure 177. BPXPRMxx Entries for a Multiple TCP/IP Transport Provider

1 Reflects the name of the RACF user ID associated with the the TCPIP started task TCPIPOE.

Note: In our case the RACF user ID had the same name as the name of the TCP/IP started task. We used TCPIPOE as our RACF user ID and TCPIPOE as our started task name. This made things easier when debugging connection problems because the RACF user ID is the one that is referenced in OpenEdition MVS to TCP/IP connection messages (EZY214xx, BPXF206I and BPXF207I), not the started task name.

2 Reflects the local INET definition.

Step 9 Start OMVS

If you start your TCP/IP transport providers when OMVS is up you should get a message like the following in Figure 178 for each transport provider.

```
EZY2141E OpenEdition-TCP/IP Connection error for TCPIPMVS-BPX1IOC ,
FFFFFFFF,0000009E,12B2005A
EZY2140I OpenEdition-TCP/IP connection established for TCPIPOE
BPXF206I ROUTING INFORMATION FOR TRANSPORT DRIVER TCPIPOE HAS BEEN
INITIALIZED OR UPDATED.
```

Figure 178. MVS Console Output after TCP/IP Transport Provider Startup

Note: You get a connection error for TCPIPMVS because there is no SUBFILESYSTYPE entry for it in the BPXPRMxx member in Figure 178. This is correct for our configuration because we are not using TCPIPMVS for any OpenEdition MVS work.

If you issue the TSO NETSTAT command now you still will not see a socket connected to the transport provider. This is because the OpenEdition MVS inetd daemon is not started. See 7.1.3, “Customizing for inetd and rlogind Daemons” on page 165 for information on how to customize the inetd daemon.

7.1.3 Customizing for inetd and rlogind Daemons

Step 1 Customize /etc/inetd.conf

The inetd daemon opens the TCP/IP ports for server processes under OpenEdition MVS after startup, and waits for incoming requests. If a request arrives, inetd starts the required server process and transfers the connection to the server process for further processing.

Inetd expects to find a configuration file in /etc/inetd.conf. We used the file /samples/inetd.conf as a sample and changed the user ID entry for rlogind to use the OMVSKERN ID, for example:

```
login    stream tcp nowait OMVSKERN usr/bin/rlogind rlogind -m
```

Inetd uses another configuration file to map the service_name field from /etc/inetd.conf to a port number. With the naming conventions we chose, we could use the default data set TCPIP.ETC.SERVICES even with multiple TCP/IP transport providers. If you want to have a separate file for OpenEdition MVS purposes you should use the /etc/service file. See *MVS/ESA Planning: OpenEdition MVS* for more information on TCP/IP V3R1 for MVS data set names with OpenEdition MVS.

Rlogind has no customization file in the HFS.

See 7.1.3, “Customizing for inetd and rlogind Daemons” and 2.6.6.2, “Define the OMVS Cataloged Procedure to RACF” on page 57 for information on RACF definitions for started tasks and deamons.

Step 2 Start the inetd daemon

To start the inetd daemon when OpenEdition MVS initializes, you can uncomment the following line in the /etc/rc file:

```
# Start the INET daemon for remote login activity
_BPX_JOBNAME='INETD' /usr/sbin/inetd  /etc/inetd.conf &
```

or a superuser can issue the inetd command from the OpenEdition MVS shell, or you can set up a procedure to start inetd from the MVS system console. A sample procedure for inetd can be found on the diskette. It is member INETD in the OMVS.PROCLIB data set.

Don’t be confused by the fact, that the INETD started task ends immediately after it is started. An OpenEdition MVS process INETDn is created by INETD. This new process represents the inetd daemon under OpenEdition MVS.

The successful startup of the inetd daemon can be verified by the following commands:

- From TSO: NETSTAT INTERVAL 10 TCP TCPIPOE

  The command should show a response like the following:

```
08/18/95             MVS TCP/IP Real Time Network Monitor
                            Connections   1 to  20
User Id     B Out    B In  L Port Foreign Socket           State
--------    -----    ----- ------ --------------           -----------
 .
 .
OMVS            0        0 7      *..*                     Listen
OMVS            0        0 7      *..*                     UDP
OMVS            0        0 9      *..*                     Listen
OMVS            0        0 9      *..*                     UDP
OMVS            0        0 13     *..*                     Listen
OMVS            0        0 13     *..*                     UDP
OMVS            0        0 19     *..*                     Listen
OMVS            0        0 19     *..*                     UDP
OMVS            0        0 37     *..*                     Listen
OMVS            0        0 37     *..*                     UDP
OMVS            0        0 513    *..*                     Listen
 .
 .
```

Figure 179. NETSTAT INTERVAL 10 TCP TCPIPOE Command

- MVS console: D OMVS,A=ALL

  The command response should contain an entry like the following:

```
OMVSKERN INETD    003C      65540          1 1FI   10.58.49      3.570
  LATCHWAITPID=         0 CMD=/usr/sbin/inetd  /etc/inetd.conf
```

Figure 180. D OMVS,A=ALL Command

7.2 Customizing and Starting NFS

Network File System is a feature of DFSMS/MVS V1R2 and provides NFS server functions under MVS/ESA. Client systems in a TCP/IP network, that support the NFS client protocol can use traditional MVS datasets and OpenEdition MVS HFS files as part of their file system.

7.2.1 Setting Up a MVS NFS Environment

Step 1 Choose the FMID to install

- HDZ11NE NFS with HFS-Support and OE-Sockets
- HDZ11NP NFS with HFS-Support and TCP/IP-Sockets

We used HDZ11NP. Be sure you have installed C/C++ Language Support Feature of MVS/ESA SP Version 5 or Language Environment for MVS and VM V1R5 or later.

Note: The MVS/DFP V3 NFS Server, (HDZ11N0) does not support OpenEdition MVS HFS.

Step 2 Run the SMP/E jobs as described in the Program Directory

Step 3 Customize the MVSNFS procedures

Copy the sample MVSNFS procedure from *hlq*.NFSSAMP(GFSAPROC) to your procedure library and customize it. Assign a user ID to the NFS procedure (we used the same user ID as TCPIP) with an OMVS segment.

NFS uses the TCP/IP RCP protocol for the client-server communication. This is why the portmapper function has to be enabled. If not already done, copy the sample PORTMAP procedure from *hlq*.SEZAINST(PORTPROC) to your procedure library. PORTMAP needs READ access to a *hlq*.ETC.RPC file. Copy it from *hlq*.SEZAINST(ETCRPC) file. We have provided sample NFS and PORTMAP procedures on the diskette, member names NFS and PORTMAP in the OMVS.PROCLIB data set for the simple TCP/IP environment described in 7.1.1, “Setting Up a Simple TCP/IP Environment” on page 153 and members NFSOE, PORTMAPO, NFSMVS and PORTMAPM in the OMVS.PROCLIB data set for the TCP/IP environment described in 7.1.2, “Setting Up the Multiple Transport Driver Support” on page 158.

Step 4 Define the required RACF profiles for the NFS started tasks

See 2.6.5, “Provide OpenEdition MVS Definitions for Other Resources” on page 51 and 2.6.6, “RACF Requirements to Run the OpenEdition MVS Kernel” on page 53 for the related information.

Step 5 Allocate the mount handle data sets. Sample JCL is provided in *hlq*.NFSSAMP(GFSAMHDJ).

Step 6 Customize the NFS attributes file and if you use EXPORTS or SAFEXP security, the NFS exports file.

For security considerations and assignment of UIDs and GIDs in your NFS network see 7.2.2, “Security Settings for the MVS NFS Environment” on page 170. We used EXPORTS security for OpenEdition MVS HFS files (NFSOE started task) and SAF security for traditional MVS data sets (NFSMVS started task) and SAFEXP security with the NFS started task, which provides access to both OpenEdition MVS HFS files and traditional MVS data sets in the single transport provider environment.

We have provided several NFS attributes and exports data sets on the diskette. Members EXPORTS and NFSATTR in the OMVS.NFS data set reflect the TCP/IP environment described in 7.1.1, “Setting Up a Simple TCP/IP Environment” on page 153 and member names EXPORTSO, NFSATTRO, and NFSATTRM in the OMVS.NFS data set reflect the TCP/IP environment described in 7.1.2, “Setting Up the Multiple Transport Driver Support” on page 158.

Step 7 Provide a code page for translation when using NFS

Create a translation table using the TSO CONVXLAT command:

```
CONVXLAT 'TCPIP.V3R1M0.SEZATCPX(OEMVS311)' 'TCPIP.MVSNFS.OEMVS311'
```

See *hlq*.SEZATCPX(OEMVS311) for the source file. MVS NFS requires the translation table to be a member of a partitioned data set. This data set is referred to by the NFSXLAT DD statement. The default translation table to be used by a NFS started task is selected by the xlat(membername) statement in the NFS attribute file. The default translation table can be overwritten by a client mount command.

Step 8 Start the appropriate Network File Systems and Portmapper started tasks.

You should now see these started tasks on the screen produced by a TSO NETSTAT INTERVAL 10 TCP TCPIPOE command shown in Figure 181 on page 169.

```
08/18/95             MVS TCP/IP Real Time Network Monitor
                            Connections   1 to  20
User Id     B Out    B In  L Port Foreign Socket           State
--------    -----    ----- ------ --------------           -----------
 ........
PORTMAPO        0        0 PMAP   *..*                     Listen
PORTMAPO     1116     1736 PMAP   *..*                     UDP
 ........
NFSOE           0        0 1029   *..*                     UDP
NFSOE           0        0 1032   *..*                     UDP
NFSOE         376      392 1035   *..*                     UDP
NFSOE         176      256 1038   *..*                     UDP
NFSOE           0        0 1041   *..*                     UDP
NFSOE       11388    26192 2049   *..*                     UDP
```

Figure 181. NETSTAT INTERVAL 10 TCP TCPIPOE Command

Step 9

Download the mvslogin and mvslogout client commands.

If you specified security(SAF) or security(SAFEXP) in your NFS attributes file you must download the mvslogin and mvslogout client commands to the NFS client system. See *DFSMS/MVS Network File System Customization and Operation* for further information.

Step 10

You should now be able to access an OpenEdition MVS HFS from a NFS client.

We will demonstrate the command sequence to be used to mount an OpenEdition MVS HFS directory from AIX and from OS/2. As mentioned above, we used EXPORTS security so we did not use the mvslogin command.

```
[C:]showexp wtsc60oe
export list for wtsc60oe:
/HFS/bin            everyone
/HFS/u/aixuser      risc01 dorniak
/HFS/u/eckhard      risc01 dorniak
/HFS/usr/adsm       risc01 dorniak

[C:\]mount z: wtsc60oe:/HFS/u/aixuser
mount: wtsc60oe:/HFS/u/aixuser
user name: dorniak
password:

NFS Drive 'z:' was attached successfully.

[C:\]qmount

Type    Name    FSDName         FSAData
Local   C:      FAT
Local   D:      FAT
Remote  J:      LAN             \\SCTD10OF\FLW
Remote  K:      LAN             \\SCTD10OF\FLG
Remote  L:      LAN             \\SCTD10OF\FL
Remote  M:      LAN             \\SCTD10OF\LICENSE
Remote  T:      LAN             \\SCTD10OF\TOOLS
Remote  Z:      NFS             wtsc60oe:/HFS/u/aixuser
```

Figure 182. Command Sequence to Mount a HFS Directory from OS/2

```
# showmount -e wtsc60oe
export list for wtsc60oe:
/HFS/bin       (everyone)
/HFS/u/aixuser risc01,dorniak
/HFS/u/eckhard risc01,dorniak
/HFS/usr/adsm  risc01,dorniak
# mount wtsc60oe:"/HFS/u/aixuser" /mnt
# df
Filesystem    Total KB    free %used   iused %iused Mounted on
/dev/hd4         53248   21600   59%    1632    12% /
/dev/hd9var       8192    5588   31%     219    10% /var
/dev/hd2        696320  213468   69%   26366    15% /usr
/dev/hd3         16384    8104   50%      62     1% /tmp
/dev/hd1         12288    1388   88%     392     9% /home
/dev/cd0        309446       0  100%  154723   100% /usr/lpp/info/En_US
/dev/lv00      1003520  954816    4%    1271     0% /usr/temp
wtsc60oe:/HFS/u/aixuser    44640    2416   94%   -       -    /mnt
```

Figure 183. Command Sequence to Mount a HFS Directory from AIX

For more information on NFS see *DFSMS/MVS Network File System User's Guide*.

7.2.2 Security Settings for the MVS NFS Environment

Mapping of UNIX security and MVS security is always cumbersome, especially if NFS is involved. The following section will explain some hints and provide customization hints.

7.2.2.1 MVS NFS Security levels

First we should mention, that four different security levels can be selected in the MVS NFS attributes data set:

1. NONE - No security checking is performed.
2. EXPORTS - EXPORTS file is used to check security.
3. SAF - System Authorization Facility checking is performed. RACF provides the security information.
4. SAFEXP - Both SAF and EXPORTS file check are performed.

For most MVS installations `security(NONE)` will be unacceptable so, security checking has to be done by RACF using the MVS SAF support or by MVS NFS using the EXPORTS file or by both mechanisms.

EXPORTS security implements the way security checking is usually done by UNIX systems with NFS Version 2 implementations. A parameter file on MVS known by DD-name EXPORTS and on UNIX systems in most cases the etc/exports file, contains the names of directories that are available for NFS requests and the host names of those NFS-client systems, that have permission to mount these directories for read and/or write purposes. The parameter file does not contain any user ID associated with the NFS mount, but the IP-host name of the NFS client.

Note: It should be mentioned, that there are several protocols that implement security functions going beyond the NFS V2 EXPORT security.

1. NFS Version 4 implements encryption and, together with the NIS protocol, the central administration of user IDs in an NFS network.

2. The OSF Distributed File System has additional advantages like file access tables, and better caching mechanisms. The OpenEdition DCE Distributed File Service for MVS/ESA is now available on MVS and is provided at no additional cost.

SAF security implements a more MVS-like approach. It requires that the NFS-client has to provide a valid RACF user ID and password to MVS NFS. The access to MVS data sets and HFS files are checked by RACF.

SAFEXP security requires that both security checks (SAF and EXPORTS) grant access to the files.

7.2.2.2 Security Information Exchange Between NFS Client and Server

How is an IP host name and a RACF user ID provided to MVS NFS?

1. How is the client IP host name resolved by the NFS server?

   If a NFS client connects to the NFS server only the IP address is transmitted over the network. On the NFS server side the client IP address is resolved into a client host name using local hosts tables or a query to a local or remote name server. It should be noted, that in most implementations it is not a big deal to use a fake IP address on the client side.

2. Let′s now concentrate on the RACF user ID.

   - For single user PC systems this is usually not a big deal.

     Most PC systems support an extension to the NFS protocol called PC-NFS client. MVS/ESA has, like many NFS implementations, a PC-NFS server. PC-NFS support is enabled by specifying the pcnfsd verb in the NFS attributes data set. If PC-NFS support is enabled on the MVS/ESA NFS server, the server prompts after a mount command was received from the NFS client asking for a user ID and password. Both are checked by RACF.

   - The problem arises with UNIX systems, which usually do not provide a PC-NFS client, because they are by design multi-user systems.

     In most cases on a UNIX system, users must have superuser authority usually called root authority to issue a mount command. Because standard mount processing does not provide user ID and password checking, NFS running on MVS/ESA has implemented a protocol extension to mount called mvsmount.

     The mvsmount protocol extension is implemented by the commands `mvslogin` and `mvslogout`, that allow the entry of a RACF user ID and password on several UNIX platforms.

7.2.2.3 Impact on HFS

How does all this affect access to the HFS?

HFS security is based on permission bits associated with a HFS file, UID and GID values associated with file and the requesting RACF user ID.

A UID associated with a user is a number specified in the OMVS segment of a RACF user ID. A GID associated with a user is a number specified in the OMVS segment of the default RACF group to which the user belongs.

Permission bits specify whether read, write, or execute permission is granted to the file owner, the group to which the file owner belongs or to everyone. When a file is created, is is automatically associated with the UID of the user that creates the file (the file owner) and the GID of the directory it is in (the parent directory).

If an OpenEdition MVS user tries to access a HFS file their UID and GID are compared with the UID and GID associated with the file. Depending on whether the values are equal OpenEdition MVS grants the access rights of the file owner, the owners group or the rights that are granted to everyone.

If NFS is used to access HFS files we must take into account which UIDs and GIDs are in effect on the NFS client system, and which security scheme is used with MVS NFS.

First, because the UID and GID number is associated with a HFS file (not related user name or group name), the `ls -l` command on a NFS client system will return different file owning user names and file owning group names than what is on the NFS server system if the assignment of UIDs to user names and GID to group names is not consistent within the NFS network.

Second, we have to consider which UID and GID is assigned to the NFS client user when they access an HFS file. This is dependent on the NFS security scheme that is in use.

- If EXPORTS security is used, there is no identification to RACF. For this reason the UID and GID values associated with the user ID acting on the NFS client system are used for OpenEdition MVS security checking. An exception from this rule is UID=0 (superuser). It is mapped to -2.
- If SAF security is used the NFS client user has to identify itself to RACF by either PC-NFS mount or the `mvslogin` command. RACF associates UID and GID values with the NFS-client user. These values are used in further processing on the MVS NFS side. The problem is that additional security checking is done in the NFS client side. This makes things complicated when the UID/GID values are not consistent in your network.

For example, we have a NFS connection between a UNIX NFS-client and a MVS NFS-server. There are four user IDs involved:

1. A user ID defined under OpenEdition MVS to be used with mvslogin. Lets call it MVSUSER with UID=200, GID=200.
2. The NFS client system user ID issuing the mount. Lets call it ROOT (ROOT has a UID=0).
3. Two other NFS client user IDs accessing the HFS file. Lets call them LUCKY with UID=200, GID=200 and UNLUCKY with UID=100, GID=100.

Independent of the security scheme used, ROOT issues the mount command and the mount is successful (On UNIX machines, you need ROOT authority to issue NFS mounts).

If EXPORTS security is used, no `mvslogin` command is needed. LUCKY will get access to MVSUSERs files as if it were MVSUSER because they have the same UID and GID values (200), UNLUCKY which has a UID=100 and GID=100 gets world access.

If SAF security is used the `mount` command will return OK, but as long as no `mvslogin` command is issued all NFS client users will get security violation

messages if they try to access a HFS file. Both LUCKY and UNLUCKY issue the `mvslogin` command with the user ID MVSUSER and the corresponding password. Now MVS NFS will grant to both users the rights of MVSUSER to HFS files. LUCKY will be happy after that, but UNLUCKY will not. Why?

MVS NFS sends the UID and GID associated with the mounted files to the NFS client system. The NFS client system will show LUCKY as owner of the files that MVSUSER owns in HFS (UID and GID both equal 200). Because the NFS client system also does security checking, UNLUCKY will be stopped by the local security system and again only get world access rights. In addition, if by chance there is a user JOE with UID=100 and GID=100 on OpenEdition MVS the NFS client system will show UNLUCKY to be the owner of JOE′s files, but if UNLUCKY tries to access the files he will be stopped by SAF security because SAF security already granted MVSUSERs rights to UNLUCKY when UNLUCKY used the `mvslogin` command earlier. Finally UNLUCKY will be totally confused and upset as you might be if you didn′t read this section slowly.

7.2.2.4 Recommendations

What are our recommendations for using NFS with OpenEdition MVS?

1. Regardless of the security scheme you use, assign consistent UID and GID values in your NFS network. Each user should have the same UID and GID on every system they are working on.
2. Enable the PC-NFS support in MVS/ESA NFS and use PC-NFS where possible.
3. Use EXPORTS security when you can trust your UNIX system administration.
4. Use SAF security when your environment has additional security requirements.

   With SAF security:

   - Follow this sequence of commands when mounting to MVS NFS:

     a. Mounting to MVS NFS:

        1) Log in to user ID root on the NFS client (if using a UNIX workstation).
        2) Issue the `mvslogin` command. This is not required to mount but useful to check whether everything is okay.
        3) Issue the `mount` command.
        4) Check the access to the HFS directory using the `df` command on UNIX or the `qmount` command on DOS and OS/2.

     b. Using HFS:

        1) Log in to the user ID you want to work with in your UNIX environment.
        2) Issue the `mvslogin` command.
        3) Now you should now be able to access the mounted file system as permitted.
        4) If you logged out from UNIX, issue the `mvslogin` command after the next UNIX login.

     c. Unmounting from MVS NFS:

1) Log in to user ID root on the NFS client (if using a UNIX workstation).

2) Be sure that no other user needs the mounted directory any more.

3) Issue the `umount` command.

4) Issue the `mvslogout` command.

- Tell the UNIX users which directories are mounted from MVS NFS, and that they may have different access rights for HFS files than for local files if UID and GID values don't match.
- Because the EXPORTS file is not used, the NFS clients `showmount` command (under OS/2 showexp) will reply

  ```
  no exported file systems
  ```

  The information would be useless anyway.
- Be aware of the fact, that the user ID used for mvslogin could also be used for other services like rlogin, ftp, and telnet.

5. SAFEXP security combines EXPORT and SAF security, making it the most secure NFS security level, but because of it's complexity, (checking UID and GID values along with user ID and passwords) this level of security may also provide additional confusion.
   - Use SAFEXP security only if you want to hide parts of the HFS from the outside world.
   - Be aware, that faking the IP-host address is not a difficult task, especially on PC systems in an office environment.
   - Keep the EXPORTS file as simple as possible. You have more flexibility, if you assign access rights to HFS files by using RACF user IDs with different UID and GID values for security checking.
   - With SAFEXP security the `showmount` command will give a response but it just reflects the EXPORTS file.
   - All other security exposures mentioned for SAF security apply also for SAFEXP security.

7.3 ADSM Client for OpenEdition MVS

The ADSM client for OpenEdition MVS is used to backup hierarchical file system files.

With DFSMSdss you can back up whole HFS file systems whereas with ADSM you can back up and restore, archive and retrieve data in the HFS on a directory and/or file level. It would go beyond the scope of this book, to describe all possible functions of ADSM. We will not describe how to set up an ADSM server but we will provide samples on how to set up the client parameter files, and show the usage of some of the ADSM client functions under OpenEdition MVS.

7.3.1 Customizing the ADSM Clients for OpenEdition MVS

Step 1 Installation with SMP/E

The ADSM client code for OpenEdition MVS consists of two FMIDs:

- HDS2AM1 containing the backup-archive client and the administrative client.
- JDS2AM1 containing the client API.

The actual installation uses normal MVS installation procedures with SMP/E. See the ADSM Program Directory for further information.

It should be mentioned that the ADSM client for OpenEdition MVS is only supported by ADSM V2 servers.

Step 2 Information provided by the ADSM server administrator

You need at least the following information from your ADSM server administrator:

- TCPPort: The port number the server uses to connect to TCP/IP.
- TCPServerAddress: The IP-address or domain name of the server.
- The type of registration scheme that is used.
  - Closed registration: Every ADSM client has to be registered by a server administrator.
  - Open registration: Every ADSM client system is allowed to register itself at the ADSM server.
- Your ADSM backup-archive client nodename and password.
- Your ADSM administrative client user ID and password.

Some other information you should get from your ADSM administrator:

- Schedule times
- Which policy domains you are allowed to use

- What backup-archive rules are associated with the policy domains. For example, the number of backup versions of a file, the retention period, and serialization logic.

Step 3 Client parameter files.

We used the following client parameter files. See the comments in the files and *ADSM Version 2 Using the UNIX Backup-Archive Clients* for more information.

/usr/adsm/dsm.sys

```
************************************************************************
* ADSTAR Distributed Storage Manager                                   *
*                                                                      *
* Client System Options file containing ADSM client system wide options*
************************************************************************
*  For information about additional options you can set in this file,
*  see the options.doc file in the directory /usr/adsm.
************************************************************************

*  There can be several ADSM Server defined in the dsm.sys file
*  We just had one ADSM Server in our network
*  WTSCMXA ADSM Server related options
Servername WTSCMXA
    TCPPort           1500
*   TCPServerAddress  wtscmxa.itsc.pok.ibm.com
    TCPServerAddress  9.12.15.5
    PasswordAccess    Generate
    Mailprog          /bin/mailx dorniak
    Inclexcl          /usr/adsm/dsm.inc

*  Client nodename, if you dont want to use your IP-Hostname
NODename   WTSC60OE

*  Incremental backup should only backup a specific part of the tree
VIRTUALMountpoint /u
VIRTUALMountpoint /etc

*  Schedule related options
*  The path where the schedule log should go
SCHEDLOG /usr/adsm/dsmsched.log

*  Default SCHEDMODe for TCP/IP connection is PRompted
SCHEDMODe POlling

*  For SCHEDMODe POlling: QUERYSCHedperiod in hours
QUERYSCHedperiod 4

*  For SchedMODe POlling: retry after minutes if a scheduled cmd failed
RETRYPeriod 10

*  SCHEDMODe PRompted requires TCPCLIENTAddress
TCPCLIENTAddress 9.12.17.2

*  SCHEDMODe PRompted requires TCPCLIENTPort
TCPCLIENTPort    1501
```

```
*  Performance related options
*  Buffersize used for internal comm. to TCP/IP transport provider
TCPBuffsize 8

*  Windowsize used for communication to ADSM server
*TCPWindowsize 32
```

/usr/adsm/dsm.opt

```
************************************************************************
* ADSTAR Distributed Storage Manager
*
* Client User Options file for MVS OpenEdition client WTSC60OE
* Contains ADSM client session specific options
************************************************************************
*  For information about additional options you can set in this file,
*  see the options.doc file in the directory /usr/adsm.
************************************************************************

*  Points to SErvername option in dsm.sys
SErvername WTSCMXA

*  Incremental backup should only backup a specific file system
*DOMain     /SERVICE

*  Incremental backup should only backup a part of a file system
*  a VIRTUALMountpoint has to be defined in dsm.sys
*DOMain     /etc

*  Include subdirectories in RESTORE, RETRIEVE, SELECTIVE BACKUP,...
SUBDIR YES
```

/usr/adsm/dsm.inc

```
************************************************************************
* ADSTAR Distributed Storage Manager                                   *
*                                                                      *
* Include/Exclude list                                                 *
************************************************************************
*  For information about additional options you can set in this file,
*  see the options.doc file in the directory /usr/adsm.
************************************************************************

*   We use VIRTUALMountpoint to specify where to start within the tree

*   Exclude all files ending with core
exclude /.../*core
*   Exclude any .sh_history file
exclude /.../.sh_history
*   Exclude /u/test/files and subdirectories
exclude /u/test/.../*
*   Exclude some /etc subdirectories
exclude /etc/recover/.../*
exclude /etc/sample/.../*
*   Exclude all files and directories under any IBM directory.
exclude /.../IBM/.../*
```

Step 4 Connect to the ADSM server

Issuing the dsmc command from the OpenEdition MVS shell will verify the installation and query some management setup information that resides on the ADSM server. Use the following commands:

```
# dsmc ADSTAR Distributed Storage Manager
Command Line Backup Client Interface - Version 2, Release 1, Level 0.0
(C) Copyright IBM Corporation, 1990, 1995, All Rights Reserved.

dsmc> q mgmtclas            1
ANS4928E PASSWORDACCESS is GENERATE, but password needed for server 'WTSCMXA'.
Either the password is not stored locally, or it was changed at the server.
Please enter password for node "WTSC60OE":


Session established with server G&M_INC: MVS
  Server Version 2, Release 1, Level 0.1
  Server date/time: 08/28/1995 21:57:27   Last access: 08/28/1995 21:57:26

Domain Name               : STANDARD
Activated Policy Set Name : STANDARD
Activation date/time      : 08/01/1995 17:50:16
Default Mgmt Class Name   : STANDARD
Grace Period Backup Retn. : 30 day(s)
Grace Period Archive Retn.: 365 day(s)


MgmtClass Name                  : ARCH2MIG
Description                     : archive to the migration spool


MgmtClass Name                  : STANDARD
Description                     : Installed default management class.

dsmc> q schedule            2

    Schedule Name: WTSC60OE
      Description: Overnight incremental backup
           Action: Incremental
          Options: -Quiet
          Objects: /u    /etc
         Priority: 5
   Next Execution: 5 Hours and 21 Minutes
         Duration: 1 Hour
           Period: 1 Day
      Day of Week: Any
           Expire: Never


dsmc> quit  #
```

Figure 184. Verifying Your ADSM Client Installation from the OE Shell

Note:

1 The response from the q mgmtclas command shows:

- A connection to the ADSM server is established, and you are prompted for a password. You may get additional prompts if

the ADSM server administrator chose to use Open Registration.

- Policy information assigned to your ADSM client system.

2 There is an incremental backup for the directories /u and /etc scheduled for you.

Step 5 The ADSM client can be started automatically every time OpenEdition MVS is brought up by adding the following to the /etc/rc file:

```
# Start the ADSM client schedule
_BPX_JOBNAME='ADSM' /usr/adsm/dsmc schedule &
```

This completes the ADSM OpenEdition MVS client installation.

7.3.2 Using the ADSM Clients for OpenEdition MVS

Basically the ADSM Client for OpenEdition MVS works like any other ADSM UNIX client. In addition, you have the ability to submit batch jobs for ADSM client processes.

By default all necessary files are put into the HFS directory /usr/adsm and related subdirectories. For the executables a symbolic link to /bin is set up. If you use this structure and the standard file names for the parameter files, no additional environment variables have to be defined.

7.3.1, “Customizing the ADSM Clients for OpenEdition MVS” on page 175 already showed an interactive session sample. We will now show how to use the ADSM Client for OpenEdition MVS from a batch job to start an incremental backup. Then we will show an interactive session to restore a file, and another interactive session with administrative commands.

7.3.2.1 How to Use the Backup/Restore Client from a Batch Job

The following JCL shows a sample of a job issuing an incremental backup. A copy of this JCL is also found in member DSMCLIEN in data set, OMVS.JCL.

```
//DORNIAKA JOB 'C9003P,B9211092','DORNIAK',NOTIFY=DORNIAK,
// MSGLEVEL=(1,1),CLASS=A,MSGCLASS=T
//* Batch job to Backup /u and /etc directories
//* See the following files for processing options:
//* dsm.sys, dsm.opt, and dsm.inc in /usr/adsm
//SHELL  EXEC  PGM=BPXBATCH,REGION=30M,          1
//         PARM='sh /usr/adsm/dsmc incremental -DOM="/u /etc" '
//STDOUT   DD  PATH='/usr/adsm/stdout',
//         PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//         PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIROTH)
//STDERR   DD  PATH='/usr/adsm/stderr',
//         PATHOPTS=(OWRONLY,OCREAT,OAPPEND),
//         PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIROTH)
//*       print the stdout file to class t (hold class)
//PRINT  EXEC  PGM=BPXBATCH,REGION=30M,          2
//         PARM='sh /bin/lp -d LOCAL,t /usr/adsm/stdout'
```

Notes:

1 The shell command to be issued is entered as a parameter string to the BPXBATCH program. An incremental backup to the directories /u and /etc is issued.

2 The second step just prints the ADSM output messages.

7.3.2.2 How to Use the Backup-Archive Client from the OMVS Shell

The ADSM backup-archive client can be used from the OpenEdition MVS shell by issuing the `dsmc` command as shown in Figure 185 on page 181.

To give an example of an interactive session we will issue some query commands. We used a user ID with superuser authority under OpenEdition MVS for the commands so we could see the whole ADSM OpenEdition MVS backup-archive client information. Users without superuser authority would only see the directories and files they own. See *ADSM Version 2 Using the UNIX Backup-Archive Clients* for more information.

```
# dsmc ADSTAR Distributed Storage Manager
Command Line Backup Client Interface - Version 2, Release 1, Level 0.0
(C) Copyright IBM Corporation, 1990, 1995, All Rights Reserved.

dsmc> q filespace                          1

Num     Last Incr Date      Type    File Space Name
---     --------------      ----    ---------------
  1   08/24/1995 21:25:02   HFS      /
  2   08/25/1995 18:44:13   HFS      /etc
  3   08/25/1995 18:44:07   HFS      /u
  4   00/00/0000 00:00:00   HFS      /SERVICE

dsmc> q backup -dirsonly /u                2
         Size      Backup Date       Mgmt Class A/I File
         ----      -----------       ---------- --- ----
            0  08/24/1995 18:41:37   STANDARD    A  /u/
            0  08/24/1995 18:41:37   STANDARD    A  /u/aixuser
            0  08/24/1995 18:41:37   STANDARD    A  /u/dorniak
            0  08/24/1995 18:41:37   STANDARD    A  /u/eckhard
            0  08/24/1995 18:41:37   STANDARD    A  /u/guerrer
            0  08/24/1995 18:41:37   STANDARD    A  /u/jesus
            0  08/24/1995 18:41:37   STANDARD    A  /u/rpetri
            0  08/24/1995 18:41:37   STANDARD    A  /u/slekka
            0  08/24/1995 18:41:37   STANDARD    A  /u/test
            0  08/24/1995 18:41:40   STANDARD    A  /u/eckhard/tmp

dsmc> q backup -ina  /u/dorniak/dsm*       3
         Size      Backup Date       Mgmt Class A/I File
         ----      -----------       ---------- --- ----
        1,345  08/25/1995 18:44:06   DEFAULT     A  /u/dorniak/dsmerror.log
          885  08/25/1995 18:44:06   DEFAULT     A  /u/dorniak/dsmsched.log
          522  08/24/1995 19:53:41   DEFAULT     I  /u/dorniak/dsmerror.log


dsmc> restore -inactive -fromdate=08/24/1995 -todate=08/24/1995 /u/dorniak/dsme
rror.log /u/dorniak/dsmerror.oldlog        4
Restore function invoked.



ANS4098I Specified directory branch structure has been restored
Restoring          522 /u/dorniak/dsmerror.log --> /u/dorniak/dsmerror.oldlog
File /u/dorniak/dsmerror.oldlog exists, do you want to replace it? (Yes/No) yes
 . Done

Restore processing finished.

dsmc> quit                                 5
#
```

Figure 185. The dsmc Command from the OMVS Shell

Notes:

1 Shows the file spaces that have been used by client commands and the related last successful incremental back up.

2 Shows which directories within the /u file space have been backed up.

3 Shows all backup versions of files starting with /u/dorniak/dsm. For the file /u/dorniak/dsmerror.log there are two backup versions; the latest one is called the active version.

4 Restore of the older (inactive) version of /u/dorniak/dsmerror.log. You get a prompt if an existing file is to be overwritten by the restore operation.

5 Quit the ADSM session.

The pick parameter of the restore command, gives a full screen like behavior even with the OpenEdition MVS shell. If you issue:

```
restore -pick -ina /u/dorniak*
```

out of a dsmc session. You will get the following screen:

```
ADSM Scrollable PICK Window - Restore

     #    Backup Date/Time       File Size A/I  File
        ---------------------------------------------------------------------
     1. | 08/24/1995 20:22:16        2166  I   /u/dorniak/.sh_history
     2. | 08/24/1995 19:53:41           0  I   /u/dorniak/cbc3uaan
     3. | 08/24/1995 19:53:41         405  A   /u/dorniak/cbc3uaan.h
     4. | 08/24/1995 18:41:37         405  I   /u/dorniak/cbc3uaan.h
     5. | 08/24/1995 19:53:41       20480  A   /u/dorniak/ctof
     6. | 08/24/1995 18:41:37       20480  I   /u/dorniak/ctof
     7. | 08/24/1995 19:53:41         923  A   /u/dorniak/ctof.c
     8. | 08/24/1995 18:41:37         923  I   /u/dorniak/ctof.c
     9. | 08/24/1995 19:53:41        4160  A   /u/dorniak/ctof.o
    10. | 08/24/1995 18:41:40        4160  I   /u/dorniak/ctof.o
    11. | 08/25/1995 18:44:06        1345  A   /u/dorniak/dsmerror.log
    12. | 08/24/1995 19:53:41         522  I   /u/dorniak/dsmerror.log
    13. | 08/25/1995 18:44:06         885  A   /u/dorniak/dsmsched.log
    14. | 08/24/1995 19:53:41           0  I   /u/dorniak/err
    15. | 08/24/1995 19:53:41        1070  A   /u/dorniak/find_strings
    16. | 08/24/1995 18:41:40        1070  I   /u/dorniak/find_strings
    17. | 08/24/1995 19:53:41          18  A   /u/dorniak/lntest
    18. | 08/24/1995 18:41:40          18  I   /u/dorniak/lntest
    19. | 08/24/1995 19:53:41          18  A   /u/dorniak/lntest1
        0---------10--------20--------30--------40--------50--------60--------7
<U>=Up  <D>=Down  <T>=Top  <B>=Bottom  <R#>=Right  <L#>=Left
<G#>=Goto Line #  <#>=Toggle Entry  <+>=Select All  <->=Deselect All
<#:#+>=Select A Range <#:#->=Deselect A Range  <O>=Ok  <C>=Cancel
pick>
```

For further processing you can use the scroll commands shown at the bottom of the screen.

7.3.2.3 How To Use the Administrator Client from the OMVS Shell

The ADSM administrative client can be used from the OpenEdition MVS shell by issuing the `dsmadmc` command.

You have to be registered as an additional administrator by the ADSM server administrator in order to get an administrator user ID and password and associated privileges. For ADSM administration you should have more knowledge of ADSM functions, than we can provide you in this book. See *ADSM Version 2 Administrator's Reference* and *ADSM Version 2 Administrator's Guide* for more information. An example of the `dsmadmc` command follows:

```
# dsmadmc
ADSTAR Distributed Storage Manager
Command Line Administrative Interface - Version 2, Release 1, Level 0.0
(C) Copyright IBM Corporation, 1990, 1995, All Rights Reserved.

Enter your user id:  dorniak
Enter your password:

ANS5100I Session established with server G&M_INC: MVS

adsm> q status   ADSM Server for MVS - Version 2, Release 1, Level 0.1/0.1


                     Server Name: G&M_INC
    Server Installation Date/Time: 07/07/1995 20:44:36
         Server Restart Date/Time: 08/21/1995 19:26:27
                   Authentication: On
       Password Expiration Period: 90 Day(s)
                     Registration: Open
                     Availability: Enabled
                       Accounting: On
    Activity Log Retention Period: 65 Day(s)
             License Audit Period: 3 Day(s)
               Last License Audit: 08/28/1995 19:14:29
        Server License Compliance: Valid
                Central Scheduler: Active
                 Maximum Sessions: 25
       Maximum Scheduled Sessions: 12
    Event Record Retention Period: 65 Day(s)
Schedule Randomization Percentage: 25
            Query Schedule Period: Client's Choice
          Maximum Command Retries: Client's Choice
                      Retry Period: Client's Choice
                  Scheduling Modes: Any
                          Log Mode: RollForward
          Database Backup Trigger: Enabled

adsm> q admin dorniak f=d

       Administrator Name: DORNIAK
    Last Access Date/Time: 08/29/1995 15:28:26
   Days Since Last Access: <1
   Password Set Date/Time: 07/28/1995 12:36:36
  Days Since Password Set: 32
                  Locked?: No
                  Contact: Rainer Heyt
         System Privilege: Yes
         Policy Privilege: ** Included with system privilege **
        Storage Privilege: ** Included with system privilege **
        Analyst Privilege: ** Included with system privilege **
       Operator Privilege: ** Included with system privilege **
   Registration Date/Time: 07/28/1995 12:36:36
Registering Administrator: RAINER

adsm> quit
```

Chapter 8. Using OpenEdition MVS

Now that OpenEdition MVS is installed and customized you may want to try to compile and run a few programs. You may want to transfer some code from a UNIX workstation to OpenEdition MVS to start some porting activities. A description of a full porting methodology is beyond the scope of this book. For more information on porting applications to OpenEdition MVS see *Porting Applications to the OpenEdition MVS Platform*. We have also provided a few small sample programs on the diskette with this book to get you started.

8.1 Transferring Source Files to OpenEdition MVS

Probably the most significant consideration when moving any data from a workstation to OpenEdition MVS is that the MVS/ESA operating system uses a completely different set of hexadecimal character representations than UNIX systems called EBCDIC (Extended Binary-Coded Decimal Interchange Code).

Given there are eight-bits to a byte, then there are 256 different possible bytes available for representing characters. Each different byte can then arbitrarily be assigned to a particular character. This process is called character encoding (or decoding). IBM defined EBCDIC is one particular character encoding scheme for use in its computers. The American National Standards Institute (ANSI) defined a different code called ASCII (American National Standard Code for Information Interchange).

All UNIX and PC systems use ASCII in one form or another, but IBM mainframes, among others, continue to use EBCDIC. The OpenEdition MVS hierarchical file system is an EBCDIC- based, byte-oriented file system. Therefore, when transferring data from any UNIX platform to OpenEdition MVS, ASCII and EBCDIC dependencies must be taken into account. This is a unique consideration for OpenEdition MVS.

A code page for a specific character set determines the graphic character produced for each hexadecimal encoding. The code page used is determined by the programs and national languages being used. For internal processing, OpenEdition MVS uses EBCDIC. To be specific, it uses the character set in the *EBCDIC Latin 1/Open Systems Interconnection Code Page 01047*. Any text to be used in OpenEdition MVS shell processing must be converted to code page 01047. Depending on its origins, it could be in any one of a number of different code pages, both ASCII and EBCDIC. For example, a tar file would normally be stored using an ASCII code set. See *OpenEdition MVS User's Guide* for more information.

8.1.1 Using NFS to Transfer Files

If you have DFSMS/MVS NFS 1.2 installed on your OpenEdition MVS system and you have NFS client code installed on your workstation, NFS mounting your HFS directly to your workstation is the preferred way to copy data to, and access data from, the hierarchical file system. See 7.2.1, “Setting Up a MVS NFS Environment” on page 167 for more information.

ASCII-EBCDIC conversion for single-byte text files is performed automatically by means of default standard conversion tables. NFS does not provide conversion

for double-byte text files. Binary data can also be copied if the HFS data set is mounted from the workstation with the binary parameter, for example:

```
mount c1jes2: "/hfs/u/smitha,binary" /u/smitha.oe
```

Using NFS to copy and access data from the workstation has shown to be the preferred method.

8.1.2 Using TCP/IP FTP to Transfer Files

If TCP/IP is installed on both the workstation and the MVS system, you can use the File Transfer Protocol (FTP) facility to transfer your source data. If transferring single-byte text data, FTP will convert the data from ASCII to EBCDIC for you. Binary data can also be sent using FTP if the binary parameter is specified.

On UNIX systems, files are often bundled together in single files by utilities like `pax`, `tar` and `cpio`. When these files are bundled into a single file, it is called an archive file. Usually the file name of the archive will indicate the utility that was used when the file was built. For example, a file named mvsport.tar indicates that the `tar` utility was used. The three utilities basically provide the same function: reading and writing of archive files. The important thing to know is `tar` and `cpio` can only read and write files of their respective formats, but `pax` can read or write in either format. So given a `pax`, `tar`, or `cpio` file, you can use `pax` from the OpenEdition MVS shell to explode or unwind the archive into its individual files.

To save disk space and transmission time, archive files can also be compressed with the `tar`, `cpio` and `pax` utilities by using the *-z* option. The naming convention that is generally used for a compressed archive file is to end the filename with a *.Z*, for example, mvsport.tar.Z.

Using FTP to transfer a file to the HFS is a two step process. You cannot transfer a file using FTP directly into the HFS. On the FTP PUT command, you must specify the name of a traditional MVS data set where you will temporarily load your archive file, then use the OPUT TSO/E command to copy the file into a directory in the HFS.

New TCP/IP function

A set of new TCP/IP functions called the OpenEdition MVS Applications Feature for IBM TCP/IP Version 3 has just been announced. One feature in particular is the OE FTP Server which will allow you to FTP a file directly into the HFS file system and out of the HFS file system without having to use an intermediate step of copying to a traditional MVS data set then using the TSO commands, OPUT, OGET, and OCOPY. Details have not been released as of the printing of this book.

Note: We have found that it is best to preallocate a very large MVS data set to temporally hold your archive file before you invoke the FTP PUT command. When we let TCP/IP allocate the data set, it generally didn't allocate the data set large enough and the FTP PUT operation failed. Alternately, you can use the FTP SITE command to pass the data set size attributes. See *TCP/IP User's Guide* for more information.

It is recommended with archive files to always use the BINARY transfer option. This applies when using the FTP GET and PUT commands as well as the TSO/E OPUT and OGET commands. If you don't use BINARY mode everywhere, your archive will most likely be corrupted and unusable.

To transfer a group of source files from an AIX workstation to OpenEdition MVS:

1. Archive the files you want to send with the `tar`, `cpio` or `pax` utilities.
2. Use FTP to send the archive to your MVS node using the binary parameter.
3. Preallocate a large MVS data set or use the FTP SITE command to pass the data set attibutes to hold the archive file.
4. Use the OPUT command to copy it to the HFS using the binary parameter.
5. Use the pax utility with the translation parameters under OpenEdition MVS to explode or unwind the archive into its original files.

The example in Figure 186 shows the process of sending the C source code that resides in a directory on the RS/6000 called /u/smith/mvsport.

```
$ cd /u/smith
$ pwd
/u/smith
$ ls -l
-rwxr-----   1 smith    system       254 Jan  7 15:22 .profile
-rw-------   1 smith    system      2228 Jan  7 15:22 .sh_history
drwxr-xr-x   2 smith    system       512 Jan  7 15:22 mvsport
$ cd mvsport
$ pwd
/u/smith/mvsport
$ ls -l
-rw-r--r--   1 smith    system      3951 Jan  7 15:22 server.c
-rw-r--r--   1 smith    system      1395 Jan  7 15:22 passwd.c
-rw-r--r--   1 smith    system      4968 Jan  7 15:22 users.c
-rw-r--r--   1 smith    system      5147 Jan  7 15:22 sql.c
$ tar -cvf mvsport.tar /u/smith/mvsport
$ ftp c1jes2
Connected to c1jes2.itsc.pok.ibm.com.
220-FTPSERVE IBM MVS V3R1 at C1JES2.ITSC.POK.IBM.COM, 14:58:31 on 10/20/95
220 Connection will close if idle for more than 5 minutes.
Name (c1jes2:root): wellie2
331 Send password please.
Password:
230 WELLIE2 is logged on.
ftp> site blk=32720 lrecl=80 recfm=fb tracks unit=3390 primary=500
200 Site command was accepted
ftp> binary
200 Representation type is IMAGE.
ftp> put /u/smith/mvsport/mvsport.tar 'wellie2.archive'
200 Port request OK.
125 Storing data set WELLIE2.ARCHIVE
250 Transfer completed successfully.
215040 bytes sent in 8.414 seconds (24.96 Kbytes/s)
ftp>
```

Figure 186. Copying Your Source Files to OpenEdition MVS Using FTP from AIX

The example in Figure 187 on page 188 shows the process of copying your archive file from the traditional MVS data set called WELLIE2.ARCHIVE, to a directory called /u/wellie2/mvsport that exists in the HFS using the OPUT TSO/E command. As you can see, we are using the `tso` shell command to execute the

OPUT command from the shell. For more information on the tso shell command, see *MVS/ESA OpenEdition MVS Command Reference*.

```
IBM
Licensed Material - Property of IBM
5655-068 (C) Copyright IBM Corp. 1993, 1995
(C) Copyright Mortice Kern Systems, Inc., 1985, 1993.
(C) Copyright Software Development Group, University of Waterloo, 1989.

All Rights Reserved.

U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

/u/wellie2: >
tso oput "'wellie2.archive' '/u/wellie2/mvsport/mvsport.tar' binary"
oput 'wellie2.archive' '/u/wellie2/mvsport/mvsport.tar' binary




 ===>

ESC=# 1=Help    2=Subcmd 3=HlpRetrn  4=Top      5=Bottom   6=TSO
      7=BackScr 8=Scroll 9=NextSess 10=Refresh 11=FwdRetr 12=Retrieve
```

Figure 187. Copying Your Source Files from an MVS Data Set to the HFS from the Shell

Alternatively, you can enter the OPUT command from option 6 of the ISPF menu, as shown in Figure 188.

```
  Menu  List  Mode  Functions  Utilities  Help
------------------------------------------------------------
                              ISPF Command Shell
Enter TSO or Workstation commands below:

===> OPUT 'WELLIE2.ARCHIVE' '/u/wellie2/mvsport/mvsport.tar' BINARY
```

Figure 188. Copying Your Source Files from an MVS Data Set to the HFS from TSO/E

The pax command can read and write archives that are in cpio, tar or pax formats along with converting the data between the ISO8859-1 (ASCII) and IBM-1047 (EBCDIC) code pages. We used the pax utility from the shell to unwind and convert (from ASCII to EBCDIC) the archive file into its original files. This is shown in Figure 189 on page 189.

```
IBM
Licensed Material - Property of IBM
5655-068 (C) Copyright IBM Corp. 1993, 1995
(C) Copyright Mortice Kern Systems, Inc., 1985, 1993.
(C) Copyright Software Development Group, University of Waterloo, 1989.

All Rights Reserved.

U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

/u/wellie2: >cd /u/wellie2/mvsport
/u/wellie2/mvsport: >




 ===> pax -o from=ISO8859-1,to=IBM-1047 -rf mvsport.tar

ESC=# 1=Help    2=Subcmd 3=HlpRetrn  4=Top      5=Bottom   6=TSO
      7=BackScr 8=Scroll 9=NextSess 10=Refresh 11=FwdRetr 12=Retrieve
```

Figure 189. Using the pax Utility to Unwind and Convert the Archive

8.2 Compiling and Running Programs under OpenEdition MVS

Now that you have your source files in a directory in the HFS file system you now need to compile and link this code. OpenEdition MVS services can be used from C/C++ functions and assembler calls. Applications written in COBOL or PL/I can use OpenEdition MVS services indirectly through C/C++ functions or assembler calls.

8.2.1 Language Installation Requirements

The new function introduced with OpenEdition MVS requires one of the following run time environments:

- Language Environment for MVS and VM, Program Number 5688-198 (formerly LE/370).

  OR

- C/C++ Language Support Feature of MVS/ESA SP Version 5, Program Number 5655-068 (JES2) or 5655-069 (JES3).

If you plan to write applications in the C/C++ high-level language (HLL) you will need to install and customize IBM C/C++ for MVS/ESA Version 3 Release 1, Program Number 5655-121. C HLL support is also provided by IBM AD/Cycle C/370 Version 1 Release 2, Program Number 5688-216.

8.2.2 Compiling with c89

An OpenEdition C/MVS application program with source code in HFS files or MVS data sets must be compiled to create output object files residing either in HFS files or MVS data sets.

Application source code can be compiled and link-edited at one time or compiled and then link-edited at another time with other application source files or compiled objects.

To compile, prelink, and link-edit an OpenEdition MVS application program from the OpenEdition MVS shell, use the c89 utility. The syntax is:

```
c89 [-options ...] [file.c ...] [file.a ...] [file.o ...] [-l libname]
```

where:

options - are the c89 compiler options.

file.c - is the source file.

file.o - is the object file.

file.a - is the archive file.

libname - is the archive library.

Note: You can compile and link-edit application program source and objects from within the shell using the OpenEdition MVS c89 utility. If you use c89, you must keep track of and maintain all the source and object files for the application program. However, you can take advantage of the make utility and create makefiles to maintain your OpenEdition MVS application source and object files automatically when you update individual modules. The make utility runs c89 for you. For more information on using the make utility, see *OpenEdition MVS Advanced Application Programming Tools*. For more information on compiling and link-editing a C/MVS program under OpenEdition MVS, see *C/C++/MVS User's Guide*.

Refer to 3.2, “Customize the Shell and Utilities” on page 102 for information on how to customize the compiler.

8.2.3 Compiling, Linking and Running the TCPS and TCPC Socket Programs

We used some small programs to verify both, the C HLL programming support and our TCP/IP configuration. Based on the TCPS and TCPC members of *hlq*.SEZAINST, we created TCP/IP server and client programs under OpenEdition MVS and AIX. We had to modify the error handling routines and change some of the include statements definitions for certain header files.

The C source code of the programs can be found on the diskette, file tcps.c for the server code and file tcpc.c for the client code. The same source code could be used for both OpenEdition MVS and AIX.

In this example, we will compile, link and run the server code (tcps) on OpenEdition MVS and compile, link and run the client code (tcpc) on an AIX machine.

Note: We will only show how to run the server on OpenEdition MVS and the client on AIX. The server can also be compiled to run on AIX and the client to run on OpenEdition MVS, or both the client and server can be run from two

different shell sessions under OpenEdition MVS. The source code was compiled with the following command from the OpenEdition MVS shell:

```
IBM
Licensed Material - Property of IBM
5655-068 (C) Copyright IBM Corp. 1993, 1995
(C) Copyright Mortice Kern Systems, Inc., 1985, 1994.
(C) Copyright Software Development Group, University of Waterloo, 1989.

All Rights Reserved.

U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 - Improve performance by preventing the propagation -
 - of TSO/E or ISPF STEPLIBs                          -
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
# c89 -D _OE_SOCKETS  -o tcps tcps.c   1
#
# tcps 999                             2
Starting to listen

Server ended successfully                        3


 ===>

ESC=¢   1=Help      2=SubCmd    3=HlpRetrn  4=Top       5=Bottom    6=TSO
        7=BackScr   8=Scroll    9=NextSess 10=Refresh  11=FwdRetr  12=Return
```

Note:

1 Use c89 to compile and link the tcps program.

2 Everything is perfect, the server now waits for a client to connect.

3 A client connected successfully.

The source code was compiled with the following command under AIX:

```
$ cc -D _AIX -o tcpc tcpc.c          1
$
$ tcpc wtsc60oe 999                  2
Client Ended Successfully
```

Note:

1 Use cc to compile and link the tcpc program.

2 The connection was successful using the host name of our MVS system, wtsc60oe and the socket address of 999.

After startup the server issues the message:

```
Starting to listen
```

and waits for a client request. If the client request comes in, the server accepts the request, receives the message sent by the client, sends the message back and ends with the message:

```
Server ended successfully
```

The caller of the client program has to provide the server's host name (or nickname) and the same port number that the caller of the server used, as arguments. Any user can issue the client program.

After startup, the client connects to the server, sends a message to the server, waits for the return from the server and ends after it received the return with the message:

```
Client Ended Successfully
```

The caller of the server program has to provide the port number to be used as an argument. On OpenEdition MVS the server has to be called from a user ID that has superuser authority or you will receive the following message:

```
Error: bind   call failed - errno=139

EDC5139I Operation not permitted.
```

If the port number that you choose happens to be in use, you will receive the following message:

```
Error: bind   call failed - errno=1115

EDC8115I Address already in use.
The socket number, 999 is in use by another program.
```

8.2.4 Compiling, Linking and Running the CTOF Program

The following example shows a simple program that converts temperatures in Celsius to Fahrenheit. You can either enter the temperature on the command line or be prompted for the temperature. The C source code for this program can be found on the diskette as file ctof.c. Type in OMVS from ISPF option 6 to get into the OpenEdition MVS shell.

```
IBM
Licensed Material - Property of IBM
5655-068 (C) Copyright IBM Corp. 1993, 1995
(C) Copyright Mortice Kern Systems, Inc., 1985, 1994.
(C) Copyright Software Development Group, University of Waterloo, 1989.

All Rights Reserved.

U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 - Improve performance by preventing the propagation -
 - of TSO/E or ISPF STEPLIBs                          -
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
$
$ c89 -o ctof ctof.c                 1
$
$ ctof                               2
Enter Celsius temperature:
0                                    3
 0.00 Celsius is 32.00 Fahrenheit
$

 ===>

ESC=¢   1=Help      2=SubCmd    3=HlpRetrn  4=Top       5=Bottom    6=TSO
        7=BackScr   8=Scroll    9=NextSess 10=Refresh  11=FwdRetr  12=Return
```

Note:

1 Use c89 to compile and link the ctof program.

2 Execute the ctof program.

3 Enter a temperature to be converted to Fahrenheit.

8.3 X Window System and OpenEdition MVS

The X Window API allows you to write applications in the OpenEdition MVS environment that can be displayed on X11 servers on a TCP/IP based network, and provides that application with graphics capabilities as defined by the X Window System protocol.

For more information on writing applications that use the X Window API from OpenEdition MVS see *IBM TCP/IP for MVS: Programmer's Reference*.

Appendix A. Refreshing RACF Profiles

When making changes to discrete or generic profiles activated for SETROPTS RACLIST processing, the changes will become effective only when a REFRESH is performed.

Activating the RACLIST processing means that for a general resource class (like the FACILITY class), RACF loads both discrete and generic profiles for the class into a data space. These profiles are available to all authorized users, so eliminating the need for RACF to retrieve a profile each time a request is made to access a resource protected by the profile. What the REFRESH command does is to delete the old data space and to load the discrete and generic profiles for the class into a new data space. This is why a REFRESH has to be done.

When protecting load modules as controlled programs with the PROGRAM class, RACF builds an in-storage profile table with the entries in the PROGRAM class. The table entries describe the programs and who can access them. If changes are done a REFRESH has to be performed so the in-storage table is updated.

If you want to perform the REFRESH for the FACILITY and the PROGRAM classes through the RACF interactive panels, you can follow the next figures.

Select option 5 of the RACF - Services Option Menu.

```
                        RACF - SERVICES OPTION MENU
OPTION ===> 5

SELECT ONE OF THE FOLLOWING:

   1  DATA SET PROFILES

   2  GENERAL RESOURCE PROFILES

   3  GROUP PROFILES AND USER-TO-GROUP CONNECTIONS

   4  USER PROFILES AND YOUR OWN PASSWORD

   5  SYSTEM OPTIONS

  99  EXIT
```

Figure 190. Refresh RACF Profiles (1 of 5)

The REFRESH option is clearly identified as option 6 of the RACF - System Security Options Menu.

```
                    RACF - SYSTEM SECURITY OPTIONS MENU
OPTION ===> 6

Select one of the following:

   1   DISPLAY            Display the current status of options.

   2   AUDIT              Set auditing options.

   3   CLASS OPTIONS      Set class-related options.

   4   PASSWORD           Set password control options.

   5   OTHER OPTIONS      Set other system security options.

   6   REFRESH            Refresh in-storage information.

   7   LANGUAGE           Set default national languages.
```

Figure 191. Refresh RACF Profiles (2 of 5)

For PROGRAM CONTROL TABLES, specify YES if you want to refresh the in-storage program control tables.

This is equivalent to specifying the parameter, WHEN(PROGRAM) with REFRESH on the SETROPTS command.

For PROFILES FOR SPECIFIC CLASSES, specify YES to display a panel with which you can refresh in-storage profiles for specific classes.

```
                           RACF - REFRESH TABLES
COMMAND ===>

To refresh in-storage PROGRAM CONTROL TABLES,
   enter YES     ===> YES


To refresh in-storage PROFILES FOR ALL CLASSES, enter the following.

   GLOBAL        ===>        YES
   GENERIC       ===>        YES


To refresh in-storage PROFILES FOR SPECIFIC CLASSES,
   enter YES     ===> YES
```

Figure 192. Refresh RACF Profiles (3 of 5)

Specify YES if you want to refresh both discrete and generic in-storage profiles for the class specified in the CLASS column.

This is equivalent to specifying RACLIST(class-name) with REFRESH on the SETROPTS command.

```
                          RACF - REFRESH TABLES
COMMAND ===>

Enter class names in the CLASS column.
To specify GLOBAL, GENERIC, or RACLIST, enter YES.
You may specify any combination of GLOBAL, GENERIC, and RACLIST.


 CLASS      GLOBAL  GENERIC  RACLIST       CLASS      GLOBAL  GENERIC RACLIST
 ---------------------------------       ---------------------------------
 FACILITY   ___     ___      YES           ________   ___     ___     ___
 ________   ___     ___      ___           ________   ___     ___     ___
 ________   ___     ___      ___           ________   ___     ___     ___
 ________   ___     ___      ___           ________   ___     ___     ___
 ________   ___     ___      ___           ________   ___     ___     ___
 ________   ___     ___      ___           ________   ___     ___     ___
 ________   ___     ___      ___           ________   ___     ___     ___
```

Figure 193. Refresh RACF Profiles (4 of 5)

At the end, you will receive the REFRESH DONE message.

```
                    RACF - SYSTEM SECURITY OPTIONS MENU          REFRESH DONE
OPTION ===>

Select one of the following:

   1   DISPLAY            Display the current status of options.

   2   AUDIT              Set auditing options.

   3   CLASS OPTIONS      Set class-related options.

   4   PASSWORD           Set password control options.

   5   OTHER OPTIONS      Set other system security options.

   6   REFRESH            Refresh in-storage information.

   7   LANGUAGE           Set default national languages.
```

Figure 194. Refresh RACF Profiles (5 of 5)

Appendix B. Setting the TZ Environment Variable

Environmental options such as the Time Zone (TZ) specification or the use of the Japanese language can be specified using the -e string. Up to 25 -e strings can be specified in a /etc/init.options file.

We considered the TZ variable to be always set in the init.options file.

The format of TZ values recognized by tzset() is as follows:
[**std**][**offset**][**dst**][**offset**]**,rule** where:

std and **dst**

indicate no less than three, nor more than TZNAME_MAX, bytes that are the designation for the standard (std) and summer (dst) time zones. If more than TZNAME_MAX bytes are specified for std or dst, tzset() truncates to TZNAME_MAX bytes. Only std is required; if dst is missing, summer time does not apply in this locale. Uppercase and lowercase letters are explicitly allowed. Any character except a leading colon (:) or digits, the comma (,), the minus (-), the plus (+), and the null character are permitted to appear in these fields. The meaning of these letters and characters is unspecified.

offset

Indicates the value one must add to the local time to arrive at Coordinated Universal Time (UTC). The offset has the form:

[**hh**[**mm ss**]]

The minutes (mm) and seconds (ss) are optional. The hour (hh) is required and may be a single digit. The offset following std is required. If no offset follows dst, summer time is assumed to be 1 hour ahead of standard time. One or more digits may be used; the value is always interpreted as a decimal number. The hour must be between 0 and 24; minutes and seconds, if present, between 0 and 59. The difference between standard time offset and summer time offset must be greater than or equal to 0; the difference may not be greater than 24 hours. Use of values outside these ranges causes tzset() to use LC_TOD category rather than TZ for time conversion information. An offset preceded by a minus (-) indicates a time zone east of the Prime Meridian. A plus (+) preceding offset is optional and indicates the time zone west of the Prime Meridian.

rule

Indicates when to change to and back from summer time. The rule has the form:

[**date/time , date/time**]

where the first date describes when the change from standard to summer time occurs and the second date describes when the change back happens. Each time field describes when, in current local time, the change to the other time is made.

The format of date must be one of the following:

Jn The Julian day n (1-n-365). Leap days are not counted. That is, in all years, including leap years, February 28 is day 59 and March 1 is day 60. It is impossible to explicitly refer to the occasional February 29.

n The zero-based Julian day (0-n-365). Leap days are counted, and it is possible to refer to February 29.

Mm.n.d
The dth day (0-d-6) of week n of month m of the year (1-n-5, and 1-m-12, where week 5 means the last d day in month m, which may occur in either the fourth or the fifth week). Week 1 is the first week in which the dth day occurs. Day zero is Sunday.

The time has the same format as offset except that no leading sign, minus (-) or plus (+), is allowed. The default, if time is not given, is 02:00:00.

If dst is specified and the rule is not specified by TZ or in the LC_TOD category, the default for the summer time start date is M4.1.0 and for the summer time end date is M10.5.0.

If the TZ variable is not set, time conversions behave as if TZ were set to TZ=GMT0 (Greenwich Mean Time).

B.1.1.1 Examples of How to Set the TZ Environment Variable

Having users located in Boston or NY City, the value of the TZ variable is EST5EDT. EST5 is Eastern Standard Time, five hours west of GMT0. EDT is Eastern Daylight Time which is understood to be four (5 minus 1).

Having users located in Germany, the value of the TZ variable is MEZ-1MESZ,086,268. MEZ stands for mitteleuropaeische zeit and is one hour east of GMT0. MESZ is the name of the zone during daylight savings time. The numbers, 086 represents the Julian date when daylight savings time starts, and 268 represents the Julian date when daylight savings time ends.

Having users located in Sydney, Australia, the value of the TZ variable is EET-10EETDT. Users in the western part of Australia, say in Perth, their TZ variable is WAUST-8WAUDT.

Appendix C. Troubleshooting

One of the most daunting problems facing someone who has just installed all the components that make up the OpenEdition MVS product set is what to do when something goes wrong.

C.1 List of Problems Encountered

The following list contains examples of problems we specifically encountered while producing this book.

Errors during OMVS startup

- After installing FMIDs JBB5522, JBB55N2, HOM1130, and JOM13N0 to upgrade from MVS 5.2.0 to 5.2.2. When starting OMVS you may get this message:

```
CSV003I REQUESTED MODULE BPXEV012 NOT FOUND

This is followed by a U4093 RC3F4 abend of BPXPLPKA.
```

  In MVS/ESA V5.1 of OpenEdition MVS the initialization program was shipped in /etc/init. Now in MVS/ESA V5.2.2 it is shipped in /usr/sbin/init. If you are installing OpenEdition MVS on top of a MVS/ESA V5.1 system you need to go in and delete the /etc/init program by hand (using the ISHELL). The order of search for the shell when looking for the initialization program is /etc/init first then /usr/sbin/init. If you don't delete the old /etc/init program by hand, then it will be incorrectly executed. This procedure will be documented in the PSP bucket for MVS/ESA V5.2.2 in the near future.

-

```
IEF403I OMVS - STARTED - TIME=08.41.04
BPXF013I FILE SYSTEM OMVS.ROOT.HFS 371
WAS SUCCESSFULLY MOUNTED.
BPXF203I DOMAIN AF_UNIX WAS SUCCESSFULLY ACTIVATED.
BPXF203I DOMAIN AF_INET WAS SUCCESSFULLY ACTIVATED.
BPXI018I THE /ETC/INIT PROCESS ENDED IN ERROR, EXIT STATUS 00000500
BPXI004I OMVS INITIALIZATION COMPLETE
```

  The ETC/INIT error above is caused by not having files, init.options and rc in the /etc directory. Copy /samples/init.options to /etc/init.options and copy /samples/rc to /etc/rc. Stop and restart OMVS.

Console errors connecting TCP/IP to OpenEdition MVS

-

```
EZY2141E OpenEdition-TCP/IP Connection error for TCPIP   -BPX1SOC ,
FFFFFFFF,0000045A,112B0000
```

  There are no TCP/IP definitions in the BPXPRMxx member of parmlib. See 7.1.1, "Setting Up a Simple TCP/IP Environment" on page 153 and 7.1.2, "Setting Up the Multiple Transport Driver Support" on page 158.

-

```
EDC5112I Resource temporarily unavailable.
EZY2141E OpenEdition-TCP/IP Connection error for TCPIPMVS-BPX1IOC ,
FFFFFFFF,0000009E,12B2005A
EZY2140I OpenEdition-TCP/IP connection established for TCPIPOE
BPXF206I ROUTING INFORMATION FOR TRANSPORT DRIVER TCPIPOE HAS BEEN
INITIALIZED OR UPDATED.
```

There are two transport providers running (TCPIPMVS and TCPIPOE), but only one is defined in the BPXPRMxx member of SYS1.PARMLIB. Only TCPIPOE can be used for OpenEdition MVS purposes. TCPIPMVS can be used for traditional MVS purposes. See Figure 176 on page 161.

Note: TCPIPMVS in message EZY2141E is the RACF user ID not the procedure name.

-

```
EZY2141E OpenEdition-TCP/IP Connection error for STC     -BPX1IOC ,
FFFFFFFF,0000009E,12B2005A
```

Reasons for this message:

- The associated RACF user ID for a TCP/IP started task has no OMVS segment.
- The TCP/IP started task is not defined as SUBFILESYSTEM in the SYS1.PARMLIB BPXPRMxx member.
- The associated RACF user ID for a TCP/IP started task is not the same as is defined in the SUBFILESYSTEM in the SYS1.PARMLIB BPXPRMxx member.

  Note: STC in message EZY2141E is the RACF user ID that the TCP/IP started task is associated with, not the procedure name.

ISPF message after calling the ISHELL program

```
Errno=9Dx A MVS environmental or internal error has occurred;
Reason=0F04013D The process is not known to current kernel instance.
Press Enter to continue.
```

OMVS has probably been restarted during your TSO session. You have to log off and log on again.

TSO message after issuing the OMVS command

```
FSUM2085I No session was started.  OpenMVS was stopped and has been restarted
FSUM2086I Function = sigprocmask, return value = FFFFFFFF, return code
9D, reason code = 0F04013D
 ***
```

OMVS has probably been restarted during your TSO session. You have to log off and log on again.

Console message during APPC startup

```
*ATB052E LOGICAL UNIT APPCOMVS FOR TRANSACTION SCHEDULER ASCH 683
 NOT ACTIVATED IN THE APPC CONFIGURATION. REASON CODE = 5A.
```

There is no VTAM application defined and active for the LU, APPCOMVS. Nevertheless OMVS will work correctly, because it uses LU=OWN conversations under MVS/ESA 5.2.2. VTAM services are not used. Consider the message as informational only. It is working as designed.

Console message during APPC/OMVS startup

```
*BPXP003E OPENMVS INIT PROCESS CANNOT BE STARTED. 382
 AN ERROR OCCURRED DURING APPC PROCESSING. APPC RETURN CODE = 10.
 VERIFY APPC AND APPC SCHEDULER ARE OPERATIVE, OR ENTER
 FORCE OMVS,ARM TO END PROCESSING.
```

We had this problem intermittently when we tried to start OMVS before APPC and ASCH were all the way up. We had to stop and restart APPC and ASCH, then start OMVS to clear this up.

Error using the TSO/E MOUNT command

If the path name specified refers to the name of an already mounted file system the following message will be displayed on the screen:

```
BPXF135E RETURN CODE 00000079, REASON CODE 055B005C.
THE MOUNT FAILED FOR FILESYSTEM OMVS.SLEKKA.ROOT.
 ***
```

The provided return code X'79' indicates that one of the parameters issued with the TSO/E MOUNT command is incorrect. The reason code is made up of 4 bytes. The high-order halfword is the reason code qualifier. The low-order halfword is the reason code description.

If the reason code qualifier bytes are within the range of 0000 to X'20FF', then the error represented by the reason code is defined by OpenMVS. In other cases the returned reason code is not an OpenMVS reason code.

The reason code qualifier of the captured error message is X'055B' and within the byte range specified above indicating an OpenEdition MVS error. The reason code itself X'005C' identifies a problem with the mount point specified in the TSO/E MOUNT command. Finally, the return code X'79' defines the problem as trying to re-mount an HFS data set that is already mounted. For more information see *Application Development Reference: Assembler Callable Services for OpenEdition MVS*.

Errors during OMVS session initialization

Some errors could occur during initialization of the OpenEdition MVS shell by users that have not been defined to their security product correctly, for example:

- No OpenEdition MVS definition.

```
 FSUM2057I No session was started.  This TSO/E user ID does not have access to
OpenMVS.+
 FSUM2058I Function = sigprocmask, return value = FFFFFFFF, return code = 000000
9C, reason code = 0B0C00FB
 ***
```

The user ID invoking the OMVS shell does not have a definition for the OpenEdition MVS environment. Add the necessary OMVS segment.

- Wrong shell program.

```
FSUM2396 No session was started.  The shell program could not be invoked

Function = execve(), program name = 'noprogram',  return value = -1, errno = 129
(X'00000081'), reason code = 057C006C, message = 'EDC5129I No such file or direc
tory.'


 >>>> FSUM2331 The session has ended.  Press <Enter> to end OMVS.
```

Make sure the user has a valid shell program name defined in their OMVS segment in the PROGRAM field.

- Wrong HOME directory.

```
 FSUM2078I No session was started.  The home directory for this TSO/E usr ID do
es not exist or cannot be accessed.+
 FSUM2079I Function = sigprocmask, return value = FFFFFFFF, return code  000000
09, reason code = 0507014D
```

List the user's OMVS segment information and check the OMVS HOME parameter. Check to make sure there is a valid mount point, pointing to a valid HFS data set.

Information messages after INETD startup

```
BPXFO24I (OMVSKERN) Aug 23 14:29:22 inetd 524294 : login/tcp: socket: 742
EDC5112I Resource temporarily unavailable.
BPXFO24I (OMVSKERN) Aug 23 14:29:22 inetd 524294 : echo/tcp: socket: 743
EDC5112I Resource temporarily unavailable.
BPXFO24I (OMVSKERN) Aug 23 14:29:23 inetd 524294 : discard/tcp: 744
socket: EDC5112I Resource temporarily unavailable.
BPXFO24I (OMVSKERN) Aug 23 14:29:23 inetd 524294 : chargen/tcp: 745
socket: EDC5112I Resource temporarily unavailable.
BPXFO24I (OMVSKERN) Aug 23 14:29:23 inetd 524294 : daytime/tcp: 746
socket: EDC5112I Resource temporarily unavailable.
BPXFO24I (OMVSKERN) Aug 23 14:29:23 inetd 524294 : time/tcp: socket: 747
EDC5112I Resource temporarily unavailable.
BPXFO24I (OMVSKERN) Aug 23 14:29:23 inetd 524294 : echo/udp: socket: 748
EDC5112I Resource temporarily unavailable.
BPXFO24I (OMVSKERN) Aug 23 14:29:23 inetd 524294 : discard/udp: 749
socket: EDC5112I Resource temporarily unavailable.
BPXFO24I (OMVSKERN) Aug 23 14:29:24 inetd 524294 : chargen/udp: 750
socket: EDC5112I Resource temporarily unavailable.
BPXFO24I (OMVSKERN) Aug 23 14:29:24 inetd 524294 : daytime/udp: 751
socket: EDC5112I Resource temporarily unavailable.
BPXFO24I (OMVSKERN) Aug 23 14:29:24 inetd 524294 : time/udp: socket: 752
EDC5112I Resource temporarily unavailable.
```

These messages are normal as long as the routing information between OMVS and the TCPIP transport providers has not been exchanged. You should get no further BPXF024I messages after the following message appears:

```
EZY2140I OpenEdition-TCP/IP connection established for TCPIPOE
BPXF206I ROUTING INFORMATION FOR TRANSPORT DRIVER TCPIPOE HAS BEEN
INITIALIZED OR UPDATED.
```

Problem with rlogin

Compare your TCP/IP buffers with what is set in the examples shipped with this book. We had a problem with rlogin and it was caused by not having the TCP/IP TINYDATABUFFERPOOLSIZE value set to a large enough number. We set it to 384.

Permission bit settings for new HFS data sets

When a new HFS data set is allocated and a mount point directory is defined, the permission bits are set to 700 and the owner field contains the user ID that allocated the mount point directory. If this data set was allocated for a general OpenEdition MVS user, the owner field needs to be updated or the general OpenEdition MVS user will receive the following message:

```
ICH408I USER(USER1) GROUP(GID2) FID(......)
INSUFFICIENT AUTHORITY TO OPEN
ACCESS INTENT(-W-) ACCESS ALLOWED(GROUP R-X)
```

Use the chown shell command to change the ownership of the mount point directory so this general user can access it.

TCP/IP hardware setup
If using a 3172 or OSA device, make sure you define an IECIOSxx member in parmlib to turn off the missing interrupt handler.

```
******************************** Top of Data ********
MIH TIME=00:00,DEV=(1F0-1FF)
******************************** Bottom of Data ******
```

TCP/IP V3.1 for MVS cannot handle a device (3172 or OSA) that has been defined to HCD as a DYNAMIC device. TCP/IP V3.1 cannot handle the dynamic UCB. You will get the following error if you define your 3172 or OSA device as DYNAMIC to HCD and try to start TCP/IP V3.1:

```
EZB4465E PCCA3 IS SHUTTING DOWN
```

Beginning with MVS/ESA V5.1 you are able to define 4-digit addresses to hardware devices. TCP/IP V3.1 cannot handle a 3172 or OSA device that has been defined with 4-digit addresses. They must be defined with 3-digit addresses. For example, 1E0-1EF will work, 5100-5116 will not.

Assignment of a GID to a new file in the HFS
The assignment of a GID to a new file works differently under AIX and OpenEdition MVS. AIX takes the GID of the owning user (the user which created the file), OpenEdition MVS takes the GID of the owning directory. This is not a bug. It is part of the POSIX.1 standard. When a new file is created, OpenEdition MVS gives it the GID of the parent directory.

DFSMSdss RESTORE of an HFS data set
When trying to restore an HFS data set that was dumped from another system to our new system, we got the following messages.

```
- RESTORE  INDD(TAPE1) OUTDD(DASD1) TOL(ENQF) -
  DATASET(INCLUDE(SYS1.HFS.ROOT       -
                  ))    -
  STORCLAS(OPENMVS)   -
  CANCELERROR
ADR780I (001)-TDDS (01), THE INPUT DUMP DATA SET BEING PROCESSED IS
                          IN LOGICAL DATA SET FORMAT
ADR380E (001)-ALLOC(04), DATA SET SYS1.HFS.ROOT NOT PROCESSED, 73
ADR415W (001)-TDLOG(01), NO DATA SETS WERE COPIED, DUMPED, OR
                          RESTORED FROM ANY VOLUME
ADR480W (001)-TDLOG(01), THE FOLLOWING DATA SETS WERE NOT PROCESSED
                          FROM THE LOGICALLY FORMATTED DUMP TAPE
                          DUE TO ERRORS:
                          SYS1.HFS.ROOT
ADR013I (001)-CLTSK(01), 95291 14:55:02 TASK COMPLETED WITH
                          RETURN CODE 0008
```

The problem is that our DFSMS ACS routines, shipped with this book, only allow SMS data sets in the OPENMVS storage class that have a high level qualifier of OMVS. In this example we are trying to restore a HFS data set with a high level qualifier of SYS1.

By adding the RENAMEU parameter to rename the SYS1 high level qualifer to OMVS, the restore is successful.

```
  RESTORE  INDD(TAPE1) OUTDD(DASD1) TOL(ENQF) -
  DATASET(INCLUDE(SYS1.HFS.ROOT           -
                 ))    -
  RENAMEU(SYS1.HFS.**,OMVS.HFS.**), -
  STORCLAS(OPENMVS)   -
  CANCELERROR
ADR780I (001)-TDDS (01), THE INPUT DUMP DATA SET BEING PROCESSED
                          IS IN LOGICAL DATA SET FORMAT
ADR711I (001)-NEWDS(01), DATA SET SYS1.HFS.ROOT HAS BEEN
                          ALLOCATED WITH NEWNAME OMVS.HFS.ROOT
                          USING STORCLAS OPENMVS, NO
                          DATACLAS, AND NO MGMTCLAS.
ADR755W (001)-PROTD(01), SOURCE DATA SET SYS1.HFS.ROOT WAS
                          GENERICALLY PROTECTED. THE TARGET DATA SET
                          OMVS.HFS.ROOT IS NOT
                          PROTECTED BY ANY PROFILE
ADR489I (001)-TDLOG(01), DATA SET OMVS.HFS.ROOT WAS RESTORED
ADR454I (001)-TDLOG(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY
                          PROCESSED
                          SYS1.HFS.ROOT
ADR013I (001)-CLTSK(01), 95291 15:51:36 TASK COMPLETED WITH
                         RETURN CODE 0004
```

Another way to fix this is to change the ACS routine to accept a high level qualifier of SYS1.

Error when starting automount

```
# automount
FOMF0107I Processing file /etc/auto.map
FOMF0112I Error issuing PFSCTL: RC=0 ERRNO=121(79) REASON=11B3005A
```

Add the following statement to the BPXPRMxx member of SYS1.PARMLIB.

```
FILESYSTYPE TYPE(AUTOMNT)
            ENTRYPOINT(BPXTAMD)
```

Appendix D. CA-ACF2 Support for OpenEdition MVS

Disclaimer

The following information has been provided by Computer Associates International, Inc. IBM assumes no responsibility for its accuracy or completeness. For more information, customers should contact their Computer Associates representative.

In environments where users move across hardware platforms and operating systems to access numerous applications, security is a major concern. Sites want and need the same control over and accountability for data and resources accessed in an open system as they are used to having in their mainframe environment. CA-ACF2 offers security for such open environments by support OpenEdition MVS and the standards developed for a Portable Operating System Interface (POSIX). Specifically, CA-ACF2 supports these services in an OpenEdition MVS environment:

- Callable Services
- Hierarchical File System (HFS)
- A New SAF Router and Interface
- New user ID (UID) and Group ID (GID) definitions
- New Audit Records for OpenEdition MVS

The following sections explain:

- Logonids needed to install OpenEdition MVS
- Controlling access to OpenEdition MVS
- Controlling access to the HFS
- CA-ACF2 records for OpenEdition MVS

For general information about Profile records, see *CA-ACF2 6.2 MVS Administrator Guide*. For explanations and syntax related to CA-ACF2 operator commands, see *CA-ACF2 System Programmers Guide*.

D.1 Logonids Needed to Install OpenEdition MVS

During the installation of OpenEdition MVS, you must create a logonid for the new OMVS started task and a logonid with OpenEdition MVS superuser authority for one or more security administrators. Follow the steps outlined below to create these logonids.

D.1.1 OMVS Started Task Logonid

First, make sure you are running with STC set in the GSO OPTS record. Then, follow the steps below to create the OMVS started task ID. For detailed information about the administrative tasks of inserting logonids or checking GSO record settings, refer to *CA-ACF2 6.1 MVS Administrator Guide*.

Step 1 Create the OpenEdition MVS kernel started task logonid by issuing the following CA-ACF2 subcommands:

```
SET LID

INSERT OMVS NAME(OpenEdition MVS ID) GROUP(OMVSGRP) STC
```

This example shows logonid OMVS created as a started task (STC is specified) and assigned to a group called OMVSGRP.

Step 2 Define the OpenEdition MVS kernel started task logonid as a superuser by issuing these CA-ACF2 subcommands:

```
SET PROFILE(USER) DIV(OMVS)

INSERT OMVS UID(0)
```

In accordance with OpenEdition MVS requirements, giving logonid OMVS a UID of zero designates it as a superuser.

Step 3 Define a GID for the group to which the logonid belongs by issuing the following CA-ACF2 subcommands:

```
SET PROFILE(GROUP) DIV(OMVS)

INSERT OMVSGRP GID(1)
```

In this example, the group OMVSGRP is given a GID value of one.

Now that the OpenEdition MVS kernel logonid has been defined to CA-ACF2, you can start OpenEdition MVS by issuing the standard operator command for started tasks.

Step 4 The OMVS started task must be given WRITE and ALLOCATE access to the HFS file and to any file that is mounted to the root file system. Be aware that your naming conventions for these mountable files affect how you give access to the OMVS started task. For example, if you mount the data set USER1.HFS.FILE to the /u/user1 mountpoint, you need a data set access rule allowing OMVS access to the data set. However, if you name the data set OMVS.USER1.HFS.FILE, you need only one rule set to allow access to all of your mountable HFS files.

For more information about mountable files, refer to *MVS/ESA Planning: OpenEdition MVS*.

D.1.2 Superuser Administrator Logonid

The OpenEdition MVS Shell and Utilities installation process creates directories in the HFS. To perform the installation steps, the user must have superuser authority. For more information about superusers, see D.2.1.1, “Superusers” on page 212.

To create a superuser administrator logonid and give it the authority it needs, follow these directions:

Step 1 Define the logonid as a superuser by issuing the following CA-ACF2 subcommands:

```
SET PROFILE(USER) DIV(OMVS)

INSERT SYSPROG1 UID(0)
```

Logonid SYSPROG1 is defined as a superuser by setting the UID value to zero.

Step 2 Define the logonid as a member of a group by issuing these CA-ACF2 subcommands:

```
SET LID

CHANGE SYSPROG1 GROUP(SYSPROG)
```

The example shows logonid SYSPROG1 changed so that this user can signon and be validated as a member of group SYSPROG. The members of group SYSPROG are a special subset of users who perform system-related tasks.

Step 3 Assign the group a GID value by issuing these CA-ACF2 subcommands:

```
SET PROFILE(GROUP) DIV(OMVS)

INSERT SYSPROG GID(20)
```

In this example, the SYSPROG group is assigned a GID of 20.

Step 4 Rebuild the USER and GROUP Profile directories as documented in D.5, “Operator Commands for OpenEdition MVS” on page 215.

D.2 Controlling Access to OpenEdition MVS

When a user attempts to enter the OpenEdition MVS shell, CA-ACF2 verifies that they are an OpenEdition MVS user before the system initializes the shell. CA-ACF2 also verifies that the user associated with a program attempting to access OpenEdition MVS resources is an OpenEdition MVS user before allowing the access to the requested resource.

To define a user as an OpenEdition MVS user, you must:

- Define the user to CA-ACF2
- Assign the user to a group
- Assign an OpenEdition MVS UID to the user
- Assign an OpenEdition MVS GID to the group

D.2.1 Defining OpenEdition MVS Users

OpenEdition MVS recognizes users by their assigned UID and GID numbers. UIDs and GIDs can have numeric values of zero to 2,147,483,647. The UID is set for a user by creating a USER Profile record. The new OMVS segment of the USER Profile record defines a user′s UID, the user′s home directory and the initial program that the user will run. The initial program is generally the shell program that the user invokes.

For more information about creating USER Profile records, see D.4, “CA-ACF2 Records for OpenEdition MVS” on page 214.

D.2.1.1 Superusers

A superuser is a special user under OpenEdition MVS. The superuser is a trusted user who can maintain the OpenEdition MVS system and administer security in the HFS. A superuser′s UID has a value of zero. Use caution when assigning users superuser authority. A superuser bypasses all security checks; this means that a superuser can access any file in the file system. This type of authority is similar to that of an unscoped security officer.

D.2.2 Defining OpenEdition MVS Groups

OpenEdition MVS security is based on user and group ownership of files and processes. CA-ACF2 uses the GROUP field of the logonid to assign the user to an OpenEdition MVS group. Users can also specify the group they want to be associated with at system entry time. A new GROUP Profile record is used to assign the GID to the group. For more information on GROUP Profile records, see D.4, “CA-ACF2 Records for OpenEdition MVS” on page 214.

We recommend that you assign a unique GID to each group. If you assign the same GID value to multiple groups, the groups share ownership of and access to the same files. This could cause unreliable results. For example, if you assign multiple groups the same GID and a getgrgid() service request is made, only one group name is returned in response to the request. CA-ACF2 searches its cross-reference tables and returns the first group that matches the GID; it does not return all the groups associated with that GID, nor can it distinguish the specific group for which you intended the request to be made.

D.2.2.1 Supplemental Groups

Under OpenEdition MVS and CA-ACF2, a user is a member of the group defined in the GROUP field of his logonid and a member of any group that he has access to via a resource rule. These groups are called supplemental groups. When group access checks are performed, CA-ACF2 first compares the GID of the file to the GID of the group defined in the logonid. If those GIDs do not match, CA-ACF2 checks to see if the file's GID matches the GID of one of the supplemental groups. If it matches, then CA-ACF2 uses the GROUP permissions to determine the user's access to the file.

For more information on how to set the owner, group and other permissions for a file, see *MVS/ESA Planning: OpenEdition MVS*.

For more information about how to give users access to groups other than the one defined in their logonid, refer to *CA-ACF2 6.1 MVS Administrator Guide*. You do not have to specify the GROUP field as part of the CA-ACF2 UID string for it to be used with OpenEdition MVS.

D.3 Controlling Access to the Hierarchical File System

The hierarchical file system (HFS) is a tree structured file system consisting of directories and files. It resembles the DOS file system, although the direction of the slashes is reversed from DOS.

Security for the file system directories and files is based on a UNIX model for security. Each file and directory is assigned an owning UID and an owning GID. This assignment is defined and saved in the file system, not in the external security product.

Three categories of users can access each directory and file in the HFS. They are:

- The file owner
- The group that owns the file
- All other users defined to OpenEdition MVS

Different access levels can be set for any of these categories. For example, permissions can be defined so that the file owner gets READ and WRITE access, a member of the file's group gets only READ access, and all other users can neither READ nor WRITE to the file.

Under CA-ACF2, you must define a UID for each OpenEdition MVS user and a GID for every group used to access OpenEdition MVS. You must also assign a default group in all OpenEdition MVS user IDs and give these users access to any supplemental groups needed.

For more information about the HFS and setting file permissions, see *MVS/ESA Planning: OpenEdition MVS* and *MVS/ESA OpenEdition MVS User's Guide*.

D.4 CA-ACF2 Records for OpenEdition MVS

OpenEdition MVS UIDs are defined to CA-ACF2 by new USER Profile records in the CA-ACF2 Infostorage database. Specifically, you define the UID information in the OMVS segment of this profile record. The OMVS segment of the USER Profile record contains three fields: UID, HOME and PROGRAM.

D.4.1 USER Profile Records

A UID is a numeric field that accepts values from zero to 2,147,483,647. A UID defined with a value of zero indicates that this is a superuser. For a definition of the term superuser, see *MVS/ESA OpenEdition MVS User's Guide*. This field does not have to be unique, but we recommend that you make it unique; otherwise, individual accountability and control are lost. This field is required.

The HOME field defines the initial directory path name. This is the initial directory used when a user enters the OMVS command or enters the ISHELL. The HOME field accepts from one to 1023 characters. Both upper and lower case characters are allowed. If HOME is not defined, OpenEdition MVS sets the initial directory for the user to the root directory. This field is optional.

The PROGRAM field defines the user's OpenEdition MVS shell program that is the first program started when the OMVS command is entered or when an OpenEdition MVS batch job is started using the BPXBATCH program. The PROGRAM field accepts from one to 1023 characters. Both upper and lower case characters are allowed. If PROGRAM is not entered, OpenEdition MVS gives control to the default shell program. This field is optional.

The example below shows how to define user OMVSUSR as a superuser. Since HOME and PROGRAM are not explicitly specified, the defaults are taken for these fields.

```
SET PROFILE(USER) DIV(OMVS)

INSERT OMVSUSR UID(0)
```

This example shows how to define user OMVSU2 as a regular user. The HOME and PROGRAM fields are defined.

```
SET PROFILE(USER) DIV(OMVS)

INSERT OMVSU2 UID(199) HOME(/u/omvsu2) PROGRAM(/bin/sh)
```

After inserting or changing any USER Profile records, rebuild the USER Profile directory as documented in D.5, "Operator Commands for OpenEdition MVS" on page 215. This directory must be rebuilt before the new or changed logonid attempts to access OpenEdition MVS resources; otherwise, CA-ACF2 will not recognize the change and the access will be denied.

D.4.2 Group Profile Records

OpenEdition MVS groups are defined to CA-ACF2 by new GROUP Profile records in the CA-ACF2 Infostorage database. The new GROUP Profile record contains the OMVS segment which consists of one field: the GID field.

GID is a numeric field that accepts values from zero to 2,147,483,647. This value does not need to be unique, but we recommend that you make the GID unique; otherwise, control over a particular group is lost.

This example shows how to insert an OMVS GROUP Profile record for a group called OMVSGRP and assign it a GID of 20.

```
SET PROFILE(GROUP) DIV(OMVS)

INSERT OMVSGRP GID(20)
```

D.4.2.1 Assigning Users to Groups under CA-ACF2

You assign a user′s default group by setting the GROUP field in that user′s CA-ACF2 logonid. Users can change their group by specifying GROUP(groupname) with their logonid and password when they log on to TSO. Resource rules control the user′s access to groups not specified in their logonids.

The example below shows you how to assign logonid OMVSU2 to group OMVSGRP:

```
SET LID

CHANGE OMVSU2 GROUP(OMVSGRP)
```

After inserting or changing any GROUP Profile records, rebuild the GROUP Profile directory as documented in D.5, “Operator Commands for OpenEdition MVS.” This directory must be rebuilt before the new or changed logonid attempts to access OpenEdition MVS resources; otherwise, CA-ACF2 will not recognize the change and the access will be denied.

D.5 Operator Commands for OpenEdition MVS

Anytime you create, change or delete a USER or GROUP Profile record, you must issue the appropriate operator command to rebuild the directories for the Profile records so that CA-ACF2 recognizes these changes.

To rebuild the directory for USER Profile records, issue this command:

```
F ACF2,REBUILD(USR),CLASS(P)
```

To rebuild the directory for GROUP Profile records, issue this command:

```
F ACF2,REBUILD(GRP),CLASS(P)
```

In addition, during initialization CA-ACF2 builds a cross-reference table to associate UIDs with their CA-ACF2 logonids and GIDs with the CA-ACF2 group name. If you add, change, or delete entries in USER or GROUP Profile records, these tables must be rebuilt so that the associations are accurate and current. Issue this operator command to rebuild these tables after you have rebuilt the directories:

```
F ACF2,OMVS
```

Appendix E. CA-Top Secret Support for OpenEdition MVS

Disclaimer

The following information has been provided by Computer Associates International, Inc. IBM assumes no responsibility for its accuracy or completeness. For more information, customers should contact their Computer Associates representative.

In environments where users move across hardware platforms and operating systems to access numerous applications, security is a major concern. Sites want and need the same control over, and accountability for, data and resources accessed in an open system as they are used to having in their mainframe environment. CA-Top Secret Release 5.0 offers security for such open environments by supporting OpenEdition MVS and the standards developed for a Portable Operating System Interface (POSIX). Specifically, CA-Top Secret supports these services in an OpenEdition MVS environment:

- Callable Services
- Hierarchical File System (HFS)
- A SAF Router and Interface
- Userid (UID) and Groupid (GID) definitions
- Audit Records for OpenEdition MVS

CA-Top Secret also supports the OpenEdition MVS Shell Setup utility. To invoke this utility, copy the REXX exec in member BPXWIRAC in data set CAI.TSSOPMAT to your REXX or CLIST library. Concatenate it to the SYSPROC DD or the SYSEXEC DD statement in your TSO SIGNON PROC. For details on this utility, see *CA-Top Secret Installation and Maintenance Guide*.

This section discusses CA-Top Secret support for OpenEdition MVS. Specifically, it covers these topics:

- ACIDs needed to install OpenEdition MVS
- Controlling access to OpenEdition MVS
- Controlling access to the HFS
- CA-Top Secret records for OpenEdition MVS

For information and syntax of CA-Top Secret command functions, refer to *CA-Top Secret 5.0 MVS Command Functions Guide*.

E.1 ACIDs Needed to Install OpenEdition MVS

During the installation of OpenEdition MVS, you must create an ACID for the OMVS started task and an ACID with OpenEdition MVS superuser authority.

E.1.1 How to Assign the OMVS Started Task ACID

OpenEdition MVS must be assigned an ACID before you can begin to using CA-Top Secret in this environment. Follow the steps below to create the OMVS started task ACID.

Step 1 Create the GROUP ACID to which the started task ACID will be attached by issuing the following TSS command:

```
TSS CREATE(OMVSGRP) TYPE(GROUP) NAME('OMVS GROUP') DEPT(OMVSDEPT)
```

Step 2 Assign a GID (groupid) to the GROUP ACID you created in Step 1. (Every group must have a GID number assigned to it.) A GID can be any number from 1 to 2,147,483,647. Define a GID of 1 for the OMVSGRP group by issuing the following TSS command:

```
TSS ADD(OMVSGRP) GID(1)
```

Step 3 You can now create the ACID to be used for the OMVS started task.

```
TSS CREATE(OMVS) TYPE(USER) NAME('OMVS STC ACID') PASS(NOPW.0)
DEPT(OMVSDEPT) FACILITY(STC.APPC)
```

Step 4 Assign the ACID to OpenEdition MVS in the STC record.

```
TSS ADD(STC) PROCNAME(OMVS) ACID(OMVS)
```

This example shows an ACID created for OMVS as a started task.

Step 5 Assign a UID to the STC ACID created in Step 3. You must define the OpenEdition MVS kernel started task ACID as a superuser by assigning it a UID of 0.

```
TSS ADD(OMVS) UID(0)
```

Note: In accordance with OpenEdition MVS requirements, giving an ACID an UID of zero automatically designates him as a superuser.

Step 6 Assign a default group to the STC ACID.

```
TSS ADD(OMVS) DFLTGRP(OMVSGRP)
```

Step 7 Assign the OMVSGRP to the OMVS ACID by issuing the following TSS command:

```
TSS ADD(OMVS) GROUP(OMVSGRP)
```

Now that the OpenEdition MVS kernel ACID as been defined to CA-Top Secret, you can start OpenEdition MVS by issuing the standard operator command for started tasks.

Step 8 The OMVS started task must be given WRITE and CREATE access to the HFS file and to any file that is mounted to the root file. Be aware that your naming conventions for these mountable files affect how you give access to the OMVS started task. For example, if you mount the data set USER1.HFS.FILE to the /u/user1 mountpoint, you need a data set access rule allowing OMVS access to the data set. However, if you name the data set OMVS.USER1.HFS.FILE, you need only one rule to allow access to all of your mountable HFS files.

For more information about mountable files, see *MVS/ESA Planning: OpenEdition MVS*.

E.1.2 How to Create the Superuser Administrator ACID

The OpenEdition MVS Shell and Utilities installation process creates directories in the HFS. To perform the installation steps, the user must have superuser authority.

A superuser is a special user ACID under OpenEdition MVS. The superuser is a trusted ACID who can maintain the OpenEdition MVS system and administer security in the HFS. It is important to note that assigning superuser authority to an ACID does not give the user any authority within CA-Top Secret, only authority in OpenEdition MVS. A superuser UID has a value of zero. Use caution when assigning ACIDs superuser authority. A superuser bypasses all security checks, which means the superuser can access any file in the file system. This type of authority is similar to that of an MSCA (master, security, control and audit) user.

To create a superuser administrator ACID and give it the authority it needs, follow these directions:

Step 1 Define the ACID as a superuser by issuing the following TSS command:

```
TSS ADD(SYSPROG1) UID(0)
```

ACID SYSPROG1 is defined as a superuser by setting the UID value to zero.

Step 2 Define SYSPROG1 as a member of a group by issuing:

```
TSS ADD(SYSPROG1) GROUP(OMVSGRP) DFLTGRP(OMVSGRP)
```

The example shows ACID SYSPROG1 changed so that this user can sign on and be validated as a member of group OMVSGRP. The ACIDs of group OMVSGRP are a special subset of users who perform system-related tasks.

Step 3 Assign the OMVSGRP group a GID value by issuing:

```
TSS ADD(OMVSGRP) GID(20)
```

The OMVSGRP group is now assigned a GID of 20.

E.1.3 Controlling Access to OpenEdition MVS

When a user attempts to enter the OpenEdition MVS shell, CA-Top Secret verifies that he is an OpenEdition MVS user before the system initializes the shell. CA-Top Secret also verifies that the user associated with a program attempting to access OpenEdition MVS resources is an OpenEdition MVS user before allowing access to the requested resource.

To define an ACID as an OpenEdition MVS user, you must:

- Define the user to CA-Top Secret
- Assign the user to a group
- Assign an OpenEdition MVS UID to the user
- Assign an OpenEdition MVS GID to the group

E.1.3.1 How to Define OpenEdition MVS Users

OpenEdition MVS recognizes ACIDs by their assigned UID and GID numbers. UIDs and GIDs can be any numeric value from zero to 2,147,483,647. The OMVS segment of the ACID defines an ACID′s UID, the user′s home directory, and the initial program that the user will run. The initial program is generally the shell program that the user invokes.

E.1.3.2 How to Define OpenEdition MVS Groups

OpenEdition MVS security is based on user and group ownership of files and processes. CA-Top Secret uses the DFLTGRP and GROUP fields of the ACID record to assign the user to an OpenEdition MVS group.

CA-Top Secret requires that the UID be unique for each ACID (except UID 0) and that the GID be unique for each group ACID. (Only UID 0 can be assigned to more than one ACID).

When group access checks are performed, CA-Top Secret compares the GID of the file to the GID of the default group defined in DFLTGRP. If it does not match, CA-Top Secret then compares the file's GID to the GIDs of all the groups defined to the ACID in the GROUP field. If a match is found, CA-Top Secret uses group permissions to determine the user's access to the file.

For more information on how to set the owner, group, and other permissions for a file, see *MVS/ESA Planning: OpenEdition MVS*.

E.1.3.3 How to Assign Users to Groups under CA-Top Secret

You assign a user's default group by setting the DFLTGRP field in that user's ACID.

This example shows how to assign ACID OMVSU2 to group OMVSGRP and then assigning that group as its default. You can assign up to 256 groups to a user using the GROUP field.

```
TSS ADD(OMVSU2) GROUP(OMVSGRP)

TSS ADD(OMVSU2) DFLTGRP(OMVSGRP)
```

Note: OMVSGRP is not a valid default group until it is also defined in the GROUP list.

E.1.3.4 How to Dynamically Change Your Default Group

You can change your default group by specifying a group when you log on to TSO. CA-Top Secret validates the specified group by checking to see if it is in your GROUP list. If it isn't in your GROUP list, you won't have a default group for the current session.

Note: All group checking for the current session is done against your GROUP list.

E.1.4 Controlling Access to the Hierarchical File System

The hierarchical file system (HFS) is a tree structured file system consisting of directories and files. It resembles the DOS file system, although the slash is used instead of the backslash.

Security for the file system directories and files is based on a UNIX model of security. Each file and directory is assigned an owning UID and an owning GID. This assignment is defined and saved in the file system, not in the external security product.

Three categories of users can access each directory and file in the HFS. They are:

- The file owner
- The group that owns the file
- All other users defined to OpenEdition MVS

Different access levels can be set for any of these three categories. For example, permissions can be defined so that the file owner gets READ and WRITE access, a member of the file's group gets only READ access, and all other users have neither READ or WRITE access to the file.

Under CA-Top Secret, you must define a UID for each OpenEdition MVS user and a GID for every group used to access OpenEdition MVS. You must also assign a default group in all OpenEdition MVS user IDs and give these users access to any supplemental groups that are needed.

For more information about the hierarchical file system and setting file permissions, see *MVS/ESA OpenEdition MVS User's Guide* and *MVS/ESA Planning: OpenEdition MVS*.

E.1.5 CA-Top Secret Records for OpenEdition MVS

CA-Top Secret records for OpenEdition MVS consist of ACID records and group profile records.

E.1.5.1 ACID Records

OpenEdition MVS UIDs are defined to CA-Top Secret by the OMVS segment in the ACID record. The OMVS segment of the ACID record contains three keywords: UID, HOME and OMVSPGM.

UID	Is a numeric keyword that accepts values from zero to 2,147,483,647. A UID defined with a value of zero indicates that this user is a superuser. This keyword must be unique to maintain individual accountability and control. A UID is required for all ACIDs in OpenEdition MVS.
HOME	Defines the initial directory path name. This is the initial directory used when a user enters the OMVS command or enters the ISPF shell. The HOME keyword accepts from one to 1024 characters. Both uppercase and lowercase characters are allowed. If HOME is not defined, OpenEdition MVS sets the initial directory for the user to the root directory. HOME is optional.
OMVSPGM	Defines the user's OpenEdition MVS shell program. This is the first program started when the OMVS command is entered or when an OpenEdition MVS batch job is started using the BPXBATCH program. The OMVSPGM keyword accepts from one to 1024 characters. Both uppercase and lowercase characters are allowed. If OMVSPGM is not entered, OpenEdition MVS gives control to the default shell program. OMVSPGM is optional.

The following example shows how to define user OMVSUSR as a superuser. Since HOME and OMVSPGM aren't explicitly specified, the defaults are taken for these fields.

```
TSS ADD(OMVSUSR) UID(0)
```

The following example shows how to define another user, OMVSU2 as a regular user. The HOME and PROGRAM keywords are also used.

```
TSS ADD(OMVSU2) UID(199) HOME(/u/omvsu2) OMVSPGM(/bin/sh)
```

E.1.5.2 GROUP Profile Records

OpenEdition MVS groups are defined to CA-Top Secret by the GROUP type ACID. The GROUP type ACID contains the OMVS segment which consists of one field: the GID keyword.

GID is a numeric field that accepts values from zero to 2,147,483,647. This value must be unique to maintain control over a particular group.

The following example shows how to create an OMVS GROUP ACID for a group called OMVSGRP and assign it a GID or 20.

```
TSS CREATE(OMVSGRP) TYPE(GROUP) NAME(OMVSGROUP) DEPT(OMVSDEPT)

TSS ADD(OMVSGRP) GID(20)
```

Appendix F. Uploading the Source Files from the Diskette

The diskette distributed with this redbook contains the source code that is described in this book. The diskette contains the following files:

```
 Directory of SG244529

README   TXT              <disclaimer information>

 Directory of BINARY      <data in binary format>

OMVS     JCL   160720     <sample jcl        - OMVS.JCL >
OMVS     NFS   127360     <sample nfs data   - OMVS.NFS >
OMVS     PRC    63280     <sample procedures - OMVS.PROCLIB >
OMVS     PRM    20560     <sample parmlib    - OMVS.PARMLIB >
TCPIP    PRF    12480     <sample profile    - TCPIP.PROFILE.TCPIP >
TCPIP    DTA     2320     <sample data file  - TCPIP.TCPIP.DATA >
TCPIP    HOS     4400     <sample hosts file - TCPIP.HOSTS.LOCAL >
TCPIPMVS PRF    16400     <sample profile    - TCPIPMVS.PROFILE.TCPIP >
TCPIPMVS DTA     9440     <sample data file  - TCPIPMVS.TCPIP.DATA >
TCPIPOE  DTA     7600     <sample data file  - TCPIPOE.TCPIP.DATA >
TCPIPOE  PRF    10160     <sample profile    - TCPIPOE.PROFILE.TCPIP >
CTOF     C       2480     <sample Convert pgm>
TCPS     C      11120     <sample Server pgm>
TCPC     C      11360     <sample Client pgm>
```

F.1 Uploading the Source Files to the MVS Host

You can copy the sample code from the supplied diskette to the HFS a number of different ways, including:

- Using the Network File System feature if you have NFS installed on both your client workstation and the MVS host
- Using the File Transfer Protocol (FTP) facility of TCP/IP when both the workstation and the MVS host have TCP/IP installed
- Using the SEND and RECEIVE commands shipped with many 3270 emulator packages, including Communication Manager for OS/2

The most important thing to remember when uploading the source code is to make sure the binary data in the BINARY directory is not translated. As an example, the steps required to upload the source code using TCP/IP FTP from an OS/2 workstation:

```
[C:\]ftp wtsc59
IBM TCP/IP for OS/2 - FTP Client ver 15:51:28 on Nov 19 1994
Connected to wtsc59.itsc.pok.ibm.com.
220-FTPSERVE IBM MVS V3R1 at WTSC59.ITSC.POK.IBM.COM, 14:00:37 on 12/04/95
220 Connection will close if idle for more than 5 minutes.
Name (wtsc59): wellie3
331 Send password please.
Password: .......
230 WELLIE3 is logged on.
ftp> site blk=32720 lrecl=80 recfm=fb tracks unit=sysallda primary=25
200 Site command was accepted
ftp> binary
200 Representation type is IMAGE.
ftp> put a:\sg244529\binary\omvs.jcl 'wellie3.omvs.jcl'
200 Port request OK.
125 Storing data set WELLIE3.OMVS.JCL
250 Transfer completed successfully.
local: a:\sg244529\binary\omvs.jcl remote: 'wellie3.omvs.jcl'
162960 bytes sent
ftp> put a:\sg244529\binary\omvs.nfs 'wellie3.omvs.nfs'
200 Port request OK.
125 Storing data set WELLIE3.OMVS.NFS
250 Transfer completed successfully.
local: a:\sg244529\binary\omvs.nfs remote: 'wellie3.omvs.nfs'
127360 bytes sent
ftp> put a:\sg244529\binary\omvs.prc 'wellie3.omvs.proclib'
200 Port request OK.
125 Storing data set WELLIE3.OMVS.PROCLIB
250 Transfer completed successfully.
local: a:\sg244529\binary\omvs.prc remote: 'wellie3.omvs.proclib'
63280 bytes sent
ftp> put a:\sg244529\binary\omvs.prm 'wellie3.omvs.parmlib'
200 Port request OK.
125 Storing data set WELLIE3.OMVS.PARMLIB
250 Transfer completed successfully.
local: a:\sg244529\binary\omvs.prm remote: 'wellie3.omvs.parmlib'
21120 bytes sent
```

Figure 195. Transferring Source Files from the Diskette (1 of 2)

```
ftp> put a:\sg244529\binary\tcpip.prf 'wellie3.tcpip.profile.tcpip'
200 Port request OK.
125 Storing data set WELLIE3.TCPIP.PROFILE.TCPIP
250 Transfer completed successfully.
local: a:\sg244529\binary\tcpip.prf remote: 'wellie3.tcpip.profile.tcpip'
12480 bytes sent
ftp> put a:\sg244529\binary\tcpip.dta 'wellie3.tcpip.tcpip.data'
200 Port request OK.
125 Storing data set WELLIE3.TCPIP.TCPIP.DATA
250 Transfer completed successfully.
local: a:\sg244529\binary\tcpip.dta remote: 'wellie3.tcpip.tcpip.data'
2320 bytes sent
ftp> put a:\sg244529\binary\tcpip.hos 'wellie3.tcpip.hosts.local'
200 Port request OK.
125 Storing data set WELLIE3.TCPIP.HOSTS.LOCAL
250 Transfer completed successfully.
local: a:\sg244529\binary\tcpip.hos remote: 'wellie3.tcpip.hosts.local'
4400 bytes sent
ftp> put a:\sg244529\binary\tcpipmvs.prf 'wellie3.tcpipmvs.profile.tcpip'
200 Port request OK.
125 Storing data set WELLIE3.TCPIPMVS.PROFILE.TCPIP
250 Transfer completed successfully.
local: a:\sg244529\binary\tcpipmvs.prf remote: 'wellie3.tcpipmvs.profile.tcpip'
16400 bytes sent
ftp> put a:\sg244529\binary\tcpipmvs.dta 'wellie3.tcpipmvs.tcpip.data'
200 Port request OK.
125 Storing data set WELLIE3.TCPIPMVS.TCPIP.DATA
250 Transfer completed successfully.
local: a:\sg244529\binary\tcpipmvs.dta remote: 'wellie3.tcpipmvs.tcpip.data'
9440 bytes sent
ftp> put a:\sg244529\binary\tcpipoe.dta 'wellie3.tcpipoe.tcpip.data'
200 Port request OK.
125 Storing data set WELLIE3.TCPIPOE.TCPIP.DATA
250 Transfer completed successfully.
local: a:\sg244529\binary\tcpipoe.dta remote: 'wellie3.tcpipoe.tcpip.data'
7600 bytes sent
ftp> put a:\sg244529\binary\tcpipoe.prf 'wellie3.tcpipoe.profile.tcpip'
200 Port request OK.
125 Storing data set WELLIE3.TCPIPOE.PROFILE.TCPIP
250 Transfer completed successfully.
local: a:\sg244529\binary\tcpipoe.prf remote: 'wellie3.tcpipoe.profile.tcpip'
10160 bytes sent
ftp> put a:\sg244529\binary\ctof.c 'wellie3.ctof.c'
200 Port request OK.
125 Storing data set WELLIE3.CTOF.C
250 Transfer completed successfully.
local: a:\sg244529\binary\ctof.c remote: 'wellie3.ctof.c'
2480 bytes sent
ftp> put a:\sg244529\binary\tcps.c 'wellie3.tcps.c'
200 Port request OK.
125 Storing data set WELLIE3.TCPS.C
250 Transfer completed successfully.
local: a:\sg244529\binary\tcps.c remote: 'wellie3.tcps.c'
11120 bytes sent
ftp> put a:\sg244529\binary\tcpc.c 'wellie3.tcpc.c'
200 Port request OK.
125 Storing data set WELLIE3.TCPC.C
250 Transfer completed successfully.
local: a:\sg244529\binary\tcpc.c remote: 'wellie3.tcpc.c'
11360 bytes sent
ftp>
```

Figure 196. Transferring Source Files from the Diskette (2 of 2)

F.2 What to Do with the Source Files?

There are a few extra things that need to be done once all the source files are uploaded to the MVS host. The JCL, NFS, PROCLIB and PARMLIB data sets that are shipped on the diskette are in TSO/E TRANSMIT OUTDATA format. You must use the TSO/E RECEIVE command to rebuild these data sets before you can use them. Enter the TSO/E RECEIVE command with the INDA parameter for

the OMVS.JCL, OMVS.NFS, OMVS.PROCLIB and OMVS.PARMLIB data sets. An example of how to convert the OMVS.JCL data set follows:

```
  Menu  List  Mode  Functions  Utilities  Help
------------------------------------------------------------
                             ISPF Command Shell
Enter TSO or Workstation commands below:

===> RECEIVE INDA('WELLIE3.OMVS.JCL')


 INMR901I Dataset OMVS.JCL from RCONWAY on ????????
 INMR906A Enter restore parameters or 'DELETE' or 'END' +
```

Figure 197. Using RECEIVE with INDA

Run the TSO/E RECEIVE command with the INDA parameter for the OMVS.NFS, OMVS.PROCLIB and OMVS.PARMLIB data sets to complete the conversion.

The CTOF.C, TCPS.C and TCPC.C source files can be copied to a directory in the HFS using the TSO/E OPUT command as shown in Figure 198. Run the OPUT command for each data set.

```
  Menu  List  Mode  Functions  Utilities  Help
------------------------------------------------------------
                             ISPF Command Shell
Enter TSO or Workstation commands below:

===> OPUT 'WELLIE3.CTOF.C' '/u/user1/ctof.c'
```

Figure 198. Using OPUT to Copy the Sample Programs into the HFS

F.3 Renaming the TCP/IP Data Sets

For more information on TCP/IP data set naming conventions when using OpenEdition MVS, see 7.1.1, “Setting Up a Simple TCP/IP Environment” on page 153 and 7.1.2, “Setting Up the Multiple Transport Driver Support” on page 158.

F.4 How to Access Samples from This Book on the Internet

The sample programs and JCL referenced in this document are available in a self-extracting zip file through the Internet.

Internet Access to Programs: If you are connected to the Internet (IBM employees will need to pass through the firewall using a tollbooth server) you can access the zip file by anonymous FTP:

```
ftp ftp.almaden.ibm.com
user: anonymous
password: your e-mail address
dir
cd redbooks/SG244529
binary
mget *.*
```

Index

Special Characters

A

B

C

D

E

F

H

I

L

M

N

O

P

R

S

T

U

V

X

ITSO Technical Bulletin Evaluation RED000

International Technical Support Organization
MVS/ESA SP 5.2.2 OpenEdition MVS
Installation and Customization Starter Kit
December 1995

Publication No. SG24-4529-00

Your feedback is very important to help us maintain the quality of ITSO Bulletins. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction	____		
Organization of the book	____	Grammar/punctuation/spelling	____
Accuracy of the information	____	Ease of reading and understanding	____
Relevance of the information	____	Ease of finding information	____
Completeness of the information	____	Level of technical detail	____
Value of illustrations	____	Print quality	____

Please answer the following questions:

a) If you are an employee of IBM or its subsidiaries:

Do you provide billable services for 20% or more of your time? Yes____ No____

Are you in a Services Organization? Yes____ No____

b) Are you working in the USA? Yes____ No____

c) Was the Bulletin published in time for your needs? Yes____ No____

d) Did this Bulletin meet your needs? Yes____ No____

If no, please explain:

_____

_____

What other topics would you like to see in this Bulletin?

_____

_____

What other Technical Bulletins would you like to see published?

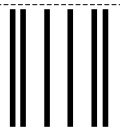
_____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

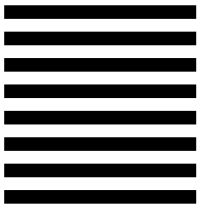
Name	Address
Company or Organization	
Phone No.	

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Organization
Mail Station P099
522 SOUTH ROAD
POUGHKEEPSIE NY
USA 12601-5400

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Printed in U.S.A.