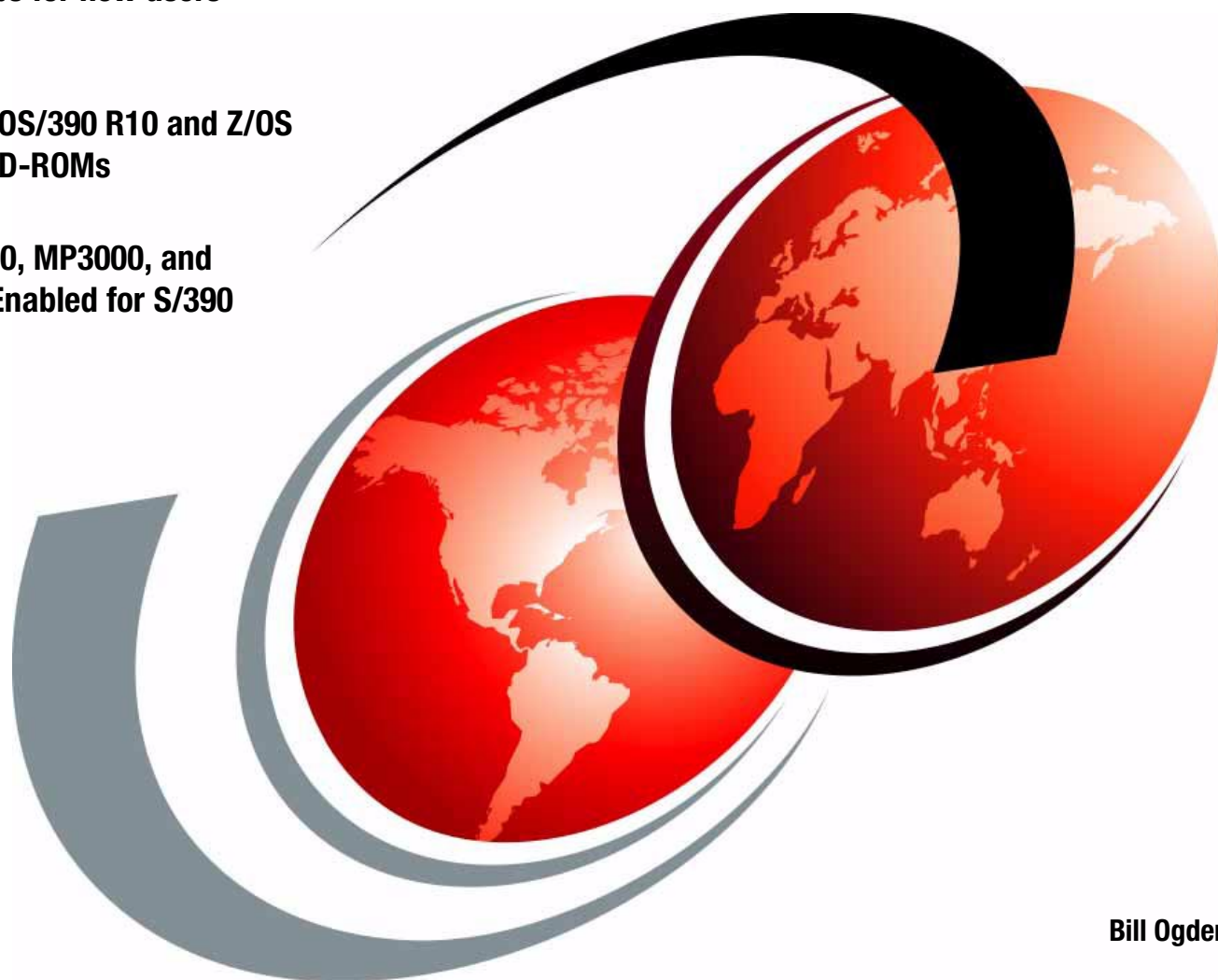


S/390 Partners in Development: OS/390 (and z/OS) New Users Cookbook

Hints & tips for new users

**Based on OS/390 R10 and Z/OS
R1.1 AD CD-ROMs**

**Uses P/390, MP3000, and
Netfinity Enabled for S/390**



Bill Ogden

Redbooks



International Technical Support Organization

**S/390 Partners in Development:
OS/390 (and z/OS)
New Users Cookbook**

November 2001

Take Note! Before using this information and the product it supports, be sure to read the general information in “Special notices” on page 367.

First Edition (November 2001)

This edition applies to the OS/390 Version 2 Release 10 Application Development CD-ROM set and the Z/OS Version 1 Release 1 Application Development CD-ROM set, produced for the IBM S/390 Partners in Development.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
2455 South Road
Poughkeepsie, NY 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.

© Copyright International Business Machines Corporation 2001. All rights reserved.

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

Contents

Contents	iii
Preface	xi
The author	xi
Special notice	xi
IBM trademarks	xii
Comments welcome	xii
Chapter 1. Introduction	1
1.1 Hardware	2
1.2 Web support	4
1.3 P/390 fixpak	4
1.4 General description of AD systems	5
1.4.1 AD system creation	5
1.5 AD CD-ROM installation	6
1.5.1 Complete installation or release migration	7
1.5.2 AD installation for P/390 or Integrated Server	8
1.5.3 AD installation for MP3000	11
1.5.4 AD installation for Netfinity/EFS	15
1.6 I/O configuration for OS/390	15
1.6.1 Changing your I/O configuration	18
1.7 Initial system use	18
1.7.1 P/390 error messages	20
Chapter 2. S/390 V2R10 AD CD-ROM	23
2.1 Specific product content	24
2.2 Volumes distributed on CD-ROMs	26
2.2.1 Disk space required	27
2.3 Parmlibs	27
2.4 IPL parameters	28
2.4.1 Detailed IPL parameter controls	29
2.5 User IDs	30
2.6 IODFs and addresses	33
2.6.1 SYS1.IODF01	33
2.6.2 SYS1.IODF99	34
2.7 Catalogs	35
2.8 Paging and spooling data sets	35
2.9 UNIX System Services file structure	36
2.10 SMP/E data sets	37
2.11 DFSMS state	38
2.12 Minor fixes	39
2.12.1 Timer settings	39
2.12.2 LOGREC full	39
2.12.3 Remove automatic ASCH and APPC startups	39
2.12.4 Reduce IPL pauses	40
2.12.5 C compiler procedures	40
2.12.6 Too many initiators	40
2.12.7 Web server does not start	41
2.12.8 /u has incorrect permissions and is very small	41

2.12.9	IPCS has nonexistent dumps in inventory	41
2.12.10	Unexpected RACF parameters	42
2.12.11	Region size for SMP/E	43
2.12.12	Web server requires RACF login	43
2.12.13	OMVS not trusted	43
2.12.14	NFS server operation	44
2.12.15	Latest P/390 device managers	44
2.12.16	VTAMLST errors and additions	44
2.12.17	ISPFLITE procedure	45
2.12.18	Shutdown scripts	46
2.12.19	UNIX command Ip	46
2.12.20	Error in inetd procedure	46
2.13	Special IPL replies	47
2.14	Shutting down the system	47
2.15	Initial MVS console PFK assignments	48
2.16	ISPF primary menu	48
2.17	Supplied jobs	49
Chapter 3. z/OS 1.1 AD CD-ROM		51
3.1	Specific product content	52
3.2	Volumes distributed on CD-ROMs	54
3.2.1	Disk space required	55
3.3	Parmlibs	55
3.4	IPL parameters	56
3.4.1	Detailed IPL parameter controls	57
3.5	User IDs	59
3.5.1	Group definitions	62
3.6	TSO logon procedures	62
3.7	IODFs and addresses	63
3.7.1	SYS1.IODF01	63
3.7.2	SYS1.IODF99	64
3.8	Catalogs	65
3.9	Paging and spooling data sets	66
3.10	UNIX System Services file structure	66
3.11	HFS mounts in BPXPRMxx	68
3.12	SMP/E data sets	68
3.13	DFSMS state	69
3.14	Minor fixes	70
3.14.1	Different IODF	70
3.14.2	Timer settings	70
3.14.3	Remove automatic ASCH and APPC startups	70
3.14.4	Too many initiators	71
3.14.5	Web server does not start	71
3.14.6	/u has incorrect permissions and is very small	71
3.14.7	IPCS has nonexistent dumps in inventory	72
3.14.8	Unexpected RACF parameters	72
3.14.9	Region size for SMP/E	73
3.14.10	Web server requires RACF login	73
3.14.11	WebSphere not customized	73
3.14.12	OMVS not trusted	73
3.14.13	DB2 mounts for HFS data sets	74
3.14.14	Latest P/390 device managers	74
3.14.15	Error in inetd procedure	74

3.14.16 VTAMLST errors and additions	74
3.14.17 ISPF LITE procedure	75
3.14.18 Shutdown scripts	76
3.14.19 UNIX command Ip	76
3.15 Special IPL replies	77
3.16 Shutting down the system	77
3.17 Initial MVS console PFK assignments	78
3.18 ISPF primary menu	78
3.19 Supplied jobs	79
Chapter 4. Additional system setup	81
4.1 Adding new disk volumes	82
4.2 Adding a user catalog	84
4.2.1 Protecting the catalogs	86
4.3 Defining user groups	86
4.4 Adding new userids	88
4.4.1 Adding TSO account numbers and procedures	91
4.5 How to add an HFS file system	92
4.6 Data set naming conventions and RACF	92
4.7 Authority for administrators	95
4.7.1 SDSF authority	95
4.8 Initial TCP/IP customization	96
4.9 IPL customization	97
4.10 Migration	98
4.10.1 Managing your own migration	99
4.11 OS/390 UNIX System Services considerations	100
Chapter 5. Routine administration and use	101
5.1 How to start the P/390 subsystem	102
5.2 Understanding addresses and device numbers	102
5.3 Understanding IPL parameters	103
5.4 How to provide a manual IPL	105
5.5 What is VTAMAPPL	108
5.6 Basic introduction to IODFs	109
5.7 How to work with PARMLIB	110
5.8 Commonly modified PARMLIB members	113
5.9 How to handle PLPA page data set full	115
5.10 How to modify a RACF user profile	115
5.11 How to delete users	116
5.12 How to reset a password	119
5.13 How to provide emergency userids	121
5.14 RACF execute vs. read permission	122
5.15 How to work with SYS1.UADS	122
5.16 How to RACF protect your data sets	124
5.17 Who can issue RACF data set profile commands	125
5.18 S/390 disk types, models, capacities	126
5.19 How to add additional emulated disk volumes	127
5.20 How to use OS/390 volume use attributes	129
5.21 How to inspect the master catalog	130
5.22 How to define user catalogs	132
5.23 How to migrate your user catalog	134
5.24 How to erase scratch disks	134
5.25 How to change a volume serial number	135

5.26	How to remove VSAM data set components	136
5.27	How to add a system-managed volume to a storage group	136
5.28	How to create and use SMS disks	138
5.28.1	Allocating to SMS volumes	150
5.29	How to select data set names	150
5.30	How to recognize a VSAM catalog	152
5.31	RACF and SAF	152
5.32	Basic security decisions	152
5.33	How to display RACF information	153
5.34	How to back up and unload RACF	157
5.35	How to use a local RACF database	158
5.36	How to protect the master catalog	160
5.37	How to submit jobs under another user's ID	160
5.38	How to grant RACF access to manipulate DASD	161
5.39	How to allocate data sets with HLQ of revoked user	162
5.40	RACF and daemons	162
5.41	How to merge RACF databases	163
5.42	How to allow a user to su to root	163
5.43	How to see detailed JES2 security messages	164
5.44	How to delete a user with a digital certificate	164
5.45	How to set rules for RACF passwords	164
5.46	How to use RACF Global Access Control	165
5.47	How to find the name of the current RACF database	166
5.48	How to define a default OpenEdition userid	166
5.49	What is a RACF user	167
5.50	Understanding the difference between UACC and ID(*)	167
5.51	How to locate HFS files with invalid UID	167
5.52	How to find data set names protected by RACF profile	168
5.53	Modifying RACF control in OpenEdition processes	168
5.54	How to program control libraries and files	168
5.55	How to permit users to view OMVS segments	170
5.56	How to use TSO authorities profiles	171
5.57	Accessing your own data sets	172
5.58	How to collect SMF data	172
5.59	How to remove RMF automatic start	174
5.60	How to increase SPOOL space	174
5.61	How to print & clear SMF data sets	175
5.62	How to clear SYS1.LOGREC	175
5.63	How to change the system's IDs	176
5.64	How to alter TSO timeout logoffs	176
5.65	How to make a logon CLIST	177
5.66	How to change the ISPF menu	179
5.67	Understanding basic SDSF security	181
5.68	How to permit SDSF users to display all jobs	184
5.69	How to change JES2 definitions	184
5.70	How to work with MIH	185
5.71	How to provide more paging space	187
5.72	Controlling virtual memory	188
5.72.1	How to specify a TSO region size	188
5.72.2	Understanding MAXASSIZE	189
5.72.3	How to select the number of page data sets	189
5.73	How to compress a PDS	190
5.74	How to allow more TSO sessions	192

5.75	How to manage dump data sets	192
5.75.1	Preallocated dump data sets	193
5.75.2	Automatically allocated dump data sets	193
5.75.3	Security implications of system dump data sets	194
5.76	How to use job and print classes	194
5.77	How to obtain JOB ENDED messages	197
5.78	How to use TSO Edit.	197
5.79	How to make a trivial IDCAMS job	199
5.80	How to view SYSOUT with ISPF.	199
5.81	How to make a simple HCD change	201
5.82	How to set up an ISPF administrative environment	203
5.83	How to interpret system data set names.	207
5.84	How to backup and restore OS/390	209
5.85	How to continue TSO command lines.	211
5.86	How to copy an ADRDSSU dump.	211
5.87	How to set your TSO prefix.	212
5.88	How to escape from a command or program	212
5.89	How to use VIO	213
5.90	How to add a program to OS/390	213
5.91	How to start a JCL library.	214
5.92	How to use ALLOC and FREE	216
5.93	How to execute a CLIST or REXX script	218
5.94	How to transfer data between OS/2 and OS/390.	219
5.95	How to select BLKSIZE for load libraries	220
5.96	What are BPXAS jobs	220
5.97	How to list system variables	221
5.98	How to write to stdout from the assembler.	221
5.99	How to find your TSO logon ID	222
5.100	How to expand a VTOC size.	222
5.101	How to repair system with TSO down	223
5.102	How to understand condition codes	224
5.103	How to use LLA (Library Lookaside)	225
5.104	How to place SCEERUN and old COBOL libraries.	226
5.105	How to control SuperZAP	226
5.106	How to use ISRDDN	226
5.107	How to edit an alias.	228
5.108	OS/390 Unix System Services terminology	229
5.109	How to clear a 3270 screen in a C program.	230
5.110	How to set and display extended attributes	230
5.111	How to assign a default printer for the lp command	231
5.112	How to execute an authorized program	232
5.113	How to create an MVS load module from OE	233
5.114	How to use the at command	233
5.115	How to use NOHUP to start a daemon	234
5.116	How to use the TSO command from OE.	234
5.117	How to source a script (dot command)	234
5.118	How to restart inetd.	235
5.119	How to reclaim disk space in an HFS	235
5.120	How to determine the size of a loaded program	235
5.121	How to prevent a user from using OE	235
5.122	How to estimate SMF output from OMVS.	235
5.123	How to move an executable from HFS to LPA	236
5.124	AMODE24 programs from OE	237

5.125	How to bounce the Web Server	237
5.126	How to use the sticky bit	237
5.127	Understand CR, LF, NL in ASCII and EBCDIC	238
5.128	MVS data sets that can move to OE via OPUT	238
5.129	The largest file size in HFS	239
5.130	How to begin using dbx	239
5.131	SMP/E	239
5.131.1	More about SMP/E	240
5.131.2	How to display PTF status	241
5.132	How to manage spool space	243
5.133	How to hot start JES2	246
5.134	How to identify a data set user	246
5.135	How to display basic IPL parameters	246
5.136	How to recover the MVS console	247
5.137	Capturing SVC dumps	247
5.138	How to use STARTED task control	249
5.139	Getting started with RMF	251
5.140	Operating the system	258
5.140.1	What the system is doing	258
5.140.2	Cancel a batch job, started task, or TSO user	260
5.140.3	Screen control	260
5.140.4	Query or control I/O devices	261
5.140.5	Displaying IPL information	264
5.140.6	ENQ/DEQ contention	264
5.140.7	Started tasks	265
5.140.8	Sending messages to users	266
5.140.9	JES2 commands	266
5.140.10	SVC dumps	268
5.140.11	RACF	268
5.140.12	How to reply to console messages	268
5.140.13	Common VTAM commands	269
Chapter 6.	More administration and use	271
6.1	How to display all PTFs in a system	272
6.2	Prelinking C programs	272
6.3	How to change the maximum OE core dump size	272
6.4	Oedit and su	272
6.5	How to work with oedit and permission bits	273
6.6	How to create an MVS data set from Open Edition	273
6.7	How to specify tape device types for HCD	273
6.8	Maximum line length for ISPF edit	274
6.9	How to change SQA and CSA sizes	274
6.10	How to adjust telnet sessions to USS	274
6.11	Binder versus linkage editor	274
6.12	How to test for z/Architecture	275
6.13	How to run a shell script at every IPL	275
6.14	How to use an ASCII tar file	275
6.15	Exposures of UACC(READ)	275
6.16	How to work with VTAMLST	276
6.17	How to stop and start VTAM	279
6.18	How to define a new VTAM application or terminal	280
6.19	How to create a VTAM logo	281
6.20	How to connect a simple SDLC 3174 to a P/390	282

6.20.1	How to use the P/390 SNA monitor	287
6.21	How to set up NJE	288
6.22	TCP/IP	292
6.22.1	How to assign IP addresses	293
6.22.2	How to provide basic OS/390 TCP/IP customization	294
6.22.3	RACF and TCP/IP	295
6.22.4	Procedures and data sets	295
6.22.5	TCP/IP profile	296
6.22.6	Other basic customization	302
6.22.7	FTP customization	303
6.22.8	inetd customization	305
6.22.9	Local host name customization	306
6.22.10	TCP/IP services definitions	306
6.22.11	Other setups and customizations	309
6.22.12	Advanced TCP/IP customization	310
6.22.13	How to begin diagnosing TCP/IP problems	311
6.23	How to compile TCP/IP applications	314
6.24	How to set FTP data set parameters	315
6.25	How to FTP from a batch job	315
6.26	How to establish TN3270E connections	316
6.27	How to have several TN3270 ports	316
6.28	How to FTP partitioned data sets	317
6.29	Why some browsers do not work with FTP	318
6.30	How to use nonstandard 3270 screen sizes	318
6.31	How to set up a test TCP/IP	320
6.32	P/390, IBM Integrated Server	321
6.32.1	How to verify P/390 disk integrity	321
6.32.2	Hardware and server setup for TCP/IP	323
6.32.3	How to prepare a P/390 for TCP/IP	323
6.32.4	Configuring selected adapters for OS/2 TCP/IP	328
6.32.5	Configuring PCOM	334
6.32.6	IBM Integrated Server setup	335
6.33	Application StarterPak LAN setup	336
6.34	How to use the CPU ID of a P/390	340
6.35	How to perform AWSTAPE mounts	341
6.36	How many token ring adapters do I need	342
6.37	How many Ethernet adapters do I need	343
6.38	How to display square brackets with PCOM	345
6.39	How to edit a PDS from OS/2	346
6.40	OS/2 does not see my CD-ROM drive	346
6.41	How to disconnect a token ring	346
6.42	Should I use HPFS386 for my P/390	347
6.43	How to determine the level of your P/390 programs	348
6.44	How to remove PCOM's printer startup	348
6.45	How to use PC memory as S/390 expanded storage	348
6.46	How to check LAN3274 connections	349
6.47	Communication line adapters for MP3000	349
6.48	How to access more PUs for LAN3172	350
6.49	3490-F01 device settings	350
6.50	How to improve emulated printer performance	351
6.51	How to install a second EtherJet adapter	351
6.52	Multiple Ethernet adapters for MP3000	352
6.53	How to unlock MPTS	352

6.54 LAN adapter plugging for MP3000	352
6.55 Canonical form of MAC addresses	352
6.56 Understanding mainframe printing	353
6.56.1 AFP data streams	354
6.56.2 How to print AFP data stream (LIST3820) files	355
6.56.3 What else do I get with AFP?	358
6.56.4 Fonts	358
6.56.5 How to deal with different printer resolutions	359
6.56.6 How to consider NLS printing	359
6.56.7 How to print a file In OS/390	360
6.56.8 Printing from CICS	360
6.56.9 How to control line count.	361
6.56.10 Controlling printing	361
6.56.11 What to do when something goes wrong	361
6.56.12 Handling AFP errors and error messages	363
6.57 How to print with a PC laser printer and a P/390	363
Special notices	367
Related publications	369
IBM Redbooks	369
Other resources	369
Referenced Web sites	369
How to get IBM Redbooks	369
IBM Redbooks collections	370
Index	371

Preface

IBM's S/390 Partners in Development (PID) organization provides several prepackaged S/390 operating systems for its members. This document is about one of these, the OS/390 (or z/OS) Application Development CD-ROM system, commonly known as the AD CD-ROM system, or simply the AD system. (Equivalent systems have been produced for VM/ESA and VSE/ESA, but this redbook concerns only the OS/390 (and z/OS) version.) The AD system is a straightforward implementation of OS/390 or z/OS and a selection of products. The CD-ROM portion of the name reflects the fact that these systems are distributed on CD-ROMs. They are intended for use on a variety of entry-level S/390 machines, including P/390s, R/390s, S/390 Integrated Servers, Multiprise 3000s, and, most recently, Netfinity and x/Series 430 Enabled for S/390 machines.

Many PID members, new to OS/390 or z/OS, are using the AD systems to port or implement their products onto this platform. This has produced many queries about basic installation and use of the AD system. Basic operations, understood without thinking by long-time MVS users, can be confusing to new users.

This document discusses the most recent AD releases, including basic planning for system administration. The bulk of the document consists of "How To" articles that may be useful to new users and owners. It does not attempt to provide comprehensive coverage of any topic, but it does address many common queries and situations encountered by new users. Users completely new to S/390 and OS/390 (or z900 and z/OS) will require initial education before using this document. The PID organization offers periodic classes for users completely new to S/390 and OS/390.

Most of the tips and techniques described in this document are not limited to the AD system, and should apply to any smaller OS/390 implementation. However, the specific examples included are based on the AD system.

The author



Bill Ogden was a Systems Engineer for many years, in IBM's World Trade organization. As a Senior Technical Staff Member at the Poughkeepsie ITSO, he specialized in entry-level OS/390 systems and related hardware. Since his formal retirement, he has continued to work part time in this area--participating in product development, writing redbooks, and teaching ITSO workshops

Special notice

This publication is intended to help new users of the OS/390 Application Development CD-ROM systems produced by IBM's S/390 Partners in Development organization. The information in this publication is not intended as the specification of any programming interfaces. See the PUBLICATIONS section of the IBM Programming Announcements for the products mentioned in this document for more information about what publications are considered to be product documentation.

IBM trademarks

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

e (logo)® 	Redbooks
IBM ®	Redbooks Logo 
AD/Cycle	Netfinity
AFP	NetSpool
AIX	Open Class
APPN	OpenEdition
AT	OS/2
BookManager	OS/390
CICS	PAL
CICS/ESA	PrintWay
CICSPIex	QMF
C/370	RAA
CUA	RACF
DB2	RMF
DFSMS/MVS	
DFSMSdss	S/390
DFSMShsm	SecureWay
DFSORT	SOMobjects
DPI	SP
e (logo)	TCS
ECKD	VisualAge
ESCON	VisualLift
FFST	VM/ESA
GDDM	VSE/ESA
IMS/ESA	VTAM
Infoprint	WebSphere
IP PrintWay	XT
Language Environment	RS/6000
Micro Channel	SAA
MQSeries	Domino
Multiprise	Notes

Comments welcome

Your comments are important to us!

We want our IBM Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- ▶ Use the online **Contact us** review redbook form found at:
ibm.com/redbooks
- ▶ Send your comments in an Internet note to:
redbook@us.ibm.com
- ▶ Mail your comments to the address on page ii.



Introduction

This IBM Redbook is intended primarily for owners of P/390s, R/390s, IBM Integrated Servers, Multiprise 3000s, and Netfinity Enabled for S/390s who are relatively new users of OS/390 and who will be using the OS/390 AD CD-ROM releases. “Relatively new” covers a wide range of skills and experience. Consequently, some of the articles in this document will be too elementary and some will be too advanced for a given reader. We apologize in advance, once, for this problem and will not mention it again. Please use what you find relevant and ignore the rest.

The first chapters may be used sequentially for introduction, setup, and planning purposes. The remainder of the document consists of many short “how to” sections, in no particular order. It is intended as a reference document, and is not intended to be read from beginning to end.

This document is a major revision of *P/390, R/390, S/390 Integrated Server: OS/390 New User's Cookbook* (SG24-4757) and completely replaces it.

Many of the examples in this document refer to a specific packaging of OS/390 and z/OS. This is the Application Developers' system distributed on CD-ROMs, and is commonly known as the AD CD-ROM system.¹ This package is available only to members of IBM's S/390 Partners-in-Development (PID) organization. Releases of the AD CD-ROM system are based on standard OS/390 ServerPacs, and are distributed several months after each new release of OS/390. At the time of writing, the current OS/390 AD CD-ROMs are based on OS/390 V2R10 and z/OS R1.1.

Although many articles in this document refer to the AD CD-ROM system, most of the discussions should be usable on any small, current OS/390 or z/OS implementation.

This document uses the term *MVS* more frequently than *OS/390* or *z/OS*. Be aware that the proper term in most cases is *OS/390* or *z/OS*. In particular, you should not infer references to pre-OS/390 time frames when you encounter *MVS* references in this document. *MVS* is one of the components of OS/390, but, as used in this book (and as commonly used elsewhere), the name *MVS* implies the complete environment of OS/390 or z/OS plus various other program products.

¹ There are also AD CD-ROM releases for VSE/ESA and VM/ESA. These are not discussed in this document.

This document is not intended to be an introduction to OS/390 or S/390. It uses common OS/390 and S/390 terminology, acronyms, and abbreviations without preliminary definitions or discussions.

1.1 Hardware

Discussions in this book usually assume the use of these machines:

- ▶ P/390

There have been three P/390 hardware generations. The first used a Micro Channel P/390 adapter card operating at about 4.5 MIPS, normally with 128 MB of S/390 memory,² and was built around a PC Server 500. The second used a PCI P/390 adapter card operating at about 4.5 MIPS with 128 MB of S/390 memory, built around a PC Server 330-ESS. The third used a PCI P/390E adapter card operating at about 7.0 MIPS with 256 MB of S/390 memory, built around a PC Server 330-PB0.

In all cases, the PC Server runs OS/2. OS/2 programs start, stop, and control the P/390 adapter card (where OS/390 runs), and provide emulation for S/390 I/O devices. All I/O is through OS/2 programs (known as *device managers*).

- ▶ R/390

There have been several R/390 versions, corresponding to P/390 machines, but built around RS/6000s instead of PCs. The R/390s use the same P/390 (or P/390E) adapter cards used in the P/390 machines. The last R/390 (using a P/390E adapter) was built around an RS/6000 F50 server.

The RS/6000s run AIX. AIX programs start, stop, and control the P/390 adapter card, and provide emulation for S/390 I/O devices. The control interfaces and S/390 emulated I/O functions are similar to those provided under OS/2 for the P/390 machines.

- ▶ S/390 Integrated Server

The Integrated Server uses a P/390E adapter, operating at about 7.0 MIPS with 256 MB of S/390 memory. It is built in a larger cabinet than the P/390 systems, with higher-performance disks, redundant power supplies, and generally more robust construction.

The Integrated Server is based on a PC, which runs OS/2 and provides exactly the same services³ as the PC Server for a P/390 machine.

- ▶ Multiprise 3000

The MP3000 is built around S/390 processors, and includes a PC functioning as a Support Element (SE). The SE runs under OS/2, and provides the same general functions provided by an SE or HMC on a larger S/390. The SE also provides emulated I/O (EIO) that is similar to that provided by OS/2 for P/390 and Integrated Server systems. In addition, the MP3000 may contain internal 3380/3390-equivalent drives⁴ and many S/390 channels (ESCON and parallel). P/390 emulated I/O processing is used, but no P/390 adapter exists in the MP3000.

- ▶ Netfinity Enabled for S/390

These Netfinity systems (and their big brother, xSeries 430 (formerly NUMA-Q) Enabled for S/390) are based on software emulation of S/390. No P/390 or other S/390 hardware is used. The software emulation is done by the FLEX-ES product from Fundamental Software, Incorporated.

² The P/390 and P/390E adapters contain their own S/390 memory, which is separate from the PC memory used by OS/2.

³ There are a few minor additions to the OS/2-provided functions, relating to SSA control and power controls.

⁴ The 3380/3390 implementation for these primary drives is provided by S/390 SAP functions, and not by OS/2 programming.

FLEX-ES also emulates various S/390 I/O devices. The basic Netfinity operating system is UnixWare (or DYNIX/ptx for xSeries 430). While the results are equivalent, the I/O emulation programs and the administrative controls are quite different than those provided for the P/390-related machines.

Note that the term “P/390” can refer to an adapter card or to the complete system containing the adapter card. It usually means a complete system.

For most of the discussions in this document, the various P/390 machines, the Integrated Server, and R/390 systems are generally lumped together as P/390s.⁵ The Multiprise 3000 is abbreviated to MP3000. The xSeries Netfinity Enabled for S/390 is abbreviated to Netfinity/EFS (Enabled For S/390).

The following two tables help summarize the similarities and differences of these systems.

	Base OS	MIPS	LPAR	ESCON Parallel	CF Link	Parallel Sysplex	S/390 Processor
P/390s	OS/2	4-7	No	Limited	No	No	P/390(E)
R/390s	AIX	4-7	No	Limited	No	No	P/390(E)
Integrated Server	OS/2	7	No	Limited	No	No	P/390E
MP3000	none ^a	50-210+	Yes	Up to 56	No	Internal	G5
Netfinity/EFS	UnixWare	~ 15 ^b	No ^c	Limited ^d	No	No	none

a. The internal Support Element runs under OS/2, and is required to manage the S/390 processor.

b. This is an approximate number, assuming one 833 Mhz PC processor is enabled for S/390.

c. However, multiple independent instances of S/390 systems may emulated at the same time.

d. Not available at time of writing, but expected in near future.

	I/O configuration Definitions	Typical S/390 DASD
P/390s	DEVMAP ^a	3390 emulated via P/390 device manager and OS/2
R/390s	DEVMAP	3390 emulated via R/390 device manager and AIX
Integrated Server	DEVMAP	3390 emulated via P/390 device manager and OS/2
MP3000	IOCDS ^b + DEVMAP ^c	Primary internal disk configured as a 3390
Netfinity x/EFS	Resource Definitions	3390 emulated via FLEX-ES emulation and UnixWare

a. Device Map, managed through an OS/2 (or AIX) application program. All S/390 I/O devices are emulated and must be defined in the DEVMAP.

b. Same IOCDS functions used by larger S/390 machines

c. A DEVMAP entry is required only for emulated I/O devices, not for internal primary disks or channel-attached devices.

All of these systems (including the MP3000) can be connected to SCSI tape drives and have emulation functions to make the SCSI drive appear as a S/390 tape unit such as a 3420, 3480, or 3490. The SCSI drive can be a 4mm unit, a DLT drive, or a 3480/3490-compatible drive.

More information about hardware and setup may be found in the following Redbooks:

- ▶ *Multiprise 3000 Technical Introduction*, SG24-5633
- ▶ *Multiprise 3000 Basic Emulated I/O Definitions*, SG24-5669

⁵ This is not quite fair to the R/390, but space and time do not permit separate coverage for it. In general, discussions of OS/2 setup and techniques do not apply to the R/390.

- ▶ *S/390 PID: Netfinity Enabled for S/390, SG24-6501* (not yet available at time of writing))
- ▶ *NUMA-Q Enabled for S/390, SG24-6215*

1.2 Web support

A Web site is available for Partners in Development members at:

`etpgw02.dfw.ibm.com/adcd.html`

Portions of this site require passwords. Among other contents, there are sections for late-breaking news and updates for the last several AD releases. Installation instructions that were on the CDs in earlier releases can now be found on this Web site.

1.3 P/390 fixpak

Many of the IBM S/390 Partners in Development organization members use P/390-based systems. z/OS is not supported on these machines. Some members have found that, if they install the most recent P/390 fixpak *and* if they have a machine based on a P/390E adapter⁶ (including Integrated Servers) then a z/OS system appears to work well (in 31-bit mode, of course).⁷ *IBM does not officially support z/OS on these machines.* If you have a problem that is uniquely related to z/OS on a P/390, IBM and the PID organization cannot help. The PID group will distribute z/OS AD systems to members with P/390E-based systems⁸ only by request.

The “most recent” fixpak⁹ discussed here is 2.5.2 (for P/390 systems) and 3.2 (for S/390 Integrated Servers). For practical purposes, these two levels can be considered equivalent. This fixpak contains fixes for a number of areas and all P/390 users should upgrade to this level.

In addition to fixes, it contains a new function providing a “system console” (or “Operating System Messages” console) equivalent to the function provided through a Support Element (or HMC) on larger systems. The associated AWSCFCN.DOC file may not completely explain how to start the function. To have the system console start automatically when the P/390 function starts (usually via the IPL icon), edit IPL.CMD in the P390 directory. Look for a section beginning with:

```
/*-----*/
/* Start the Device Mangers which have to be ready before we IPL      */
/*-----*/
```

and find the following line in this section:

```
/* call dmstart 'awsCFCN.exe t /T"Operating System Messages"' dmparms awscfcon */
```

You need to uncomment this line by removing the `/*` and `*/` characters at the beginning and end of the line. After doing this, the next P/390 IPL should start an additional OS/2 window containing the system console function. Linux for S/390, in particular, needs this window for installation and operation.

⁶ Older P/390 systems will not work with z/OS. In general, if you have a P/390-based machine with 256 MB of S/390 storage, it uses a P/390E adapter and the comments here apply to it.

⁷ The author used a P/390E system while developing the material discussed in this redbook.

⁸ Such systems must have been obtained through the PID organization to be eligible for AD system distributions. Other criteria may limit a particular member's eligibility for an AD distribution.

⁹ P/390 fixpaks are obtained from the ftp functions at `p390.ibm.com`.

1.4 General description of AD systems

Counting the current release, there have been 13 releases of the AD system. (Two were for MVS, ten for OS/390, and one for z/OS.) Each release has been different in various ways, but many common characteristics have been carried from release to release. As a general statement, each release has been larger, in terms of disk space, and more program products have been added.

1.4.1 AD system creation

The system is based on a ServerPac, the same starting point used by most OS/390 customers. The ServerPac is received and installed on a normal S/390.¹⁰ The installation and customization processes are completed for the base OS/390 system plus all the additional program products included with the AD system. All products are installed at least through the end of the ServerPac generated jobs. The base system and some of the additional products are further customized to make them more immediately useful.

Additional details are provided in the SYS1.README data set. This is a PDS with a number of members, each addressing different parts of the installation and customization processes done when building the AD CD-ROM. (SYS1.README is on the OS39M1 volume, so you must install this volume and use an OS/390 system to access it in order to read the various members in it.)

The result of the ServerPac (and additional program products) installation and customization is a group of 3390 volumes containing the new system. The complete R10 system, for example, requires twelve 3390 volumes (with a mixture of 3390-3 and 3390-2 units). These volumes are then dumped and restored to a P/390 system (usually an Integrated Server, because these tend to have more disk space than other, older P/390s).

At this point, each 3390 volume is an OS/2 file¹¹ residing on the SCSI (or SSA, depending on the specific machine) disks belonging to OS/2. During S/390 operation, an emulated I/O *device manager* accesses this file and uses it to emulate a “real” 3390 drive. An OS/2 zip program is used to zip (compress) each of these OS/2 files, and the zipped file is written to a CD-ROM. The zip compression process is very effective and a 2.8 GB 3390-3 almost always compresses to less than the 650 MB that will fit on a CD-ROM. Sometimes several zipped volumes will fit on one CD-ROM. The exact layout of the recent releases is described in the next chapters.

The CD-ROMs are then copied and form an AD CD-ROM release. Installation consists of unzipping the files on the CD-ROMs and placing the unzipped files on OS/2 disks. For an R/390, an AIX unzip program is used, but the other details are the same. For an MP3000, the CD-ROMs are unzipped in the Support Element (a PC processor) and they can be used there (as emulated I/O drives) or copied to the primary drives. For a Netfinity/EFS system, unique FLEX-ES programs copy and unzip the files and convert them into the emulated I/O format used by FLEX-ES.

In principle, an AD CD-ROM distribution could be used on any S/390 (or z900) system. For larger S/390s, the challenge would be to read the CD-ROM files, unzip them, and then copy the emulated I/O volumes to “real” 3390s. In practice, the AD systems have not been routinely used with larger S/390 systems thus far.

¹⁰ In principle, this could be done on a P/390. In practice, a larger system is used.

¹¹ 3390-3 volumes use two OS/2 files, due to the maximum size limitation of 2 GB for a single OS/2 file. In the present context, this detail is ignored.

1.5 AD CD-ROM installation

The basic AD CD-ROM installation process is outlined in Figure 1-1. The general process involves unzipping the files on the CD-ROMs. This produces the OS/2 files (in the format used for P/390 emulated I/O) that contain the OS/390 volumes. For P/390 systems, this is sufficient. During P/390 operation, these OS/2 files are accessed by the emulated I/O device managers and appear to OS/390 as normal 3390 volumes.

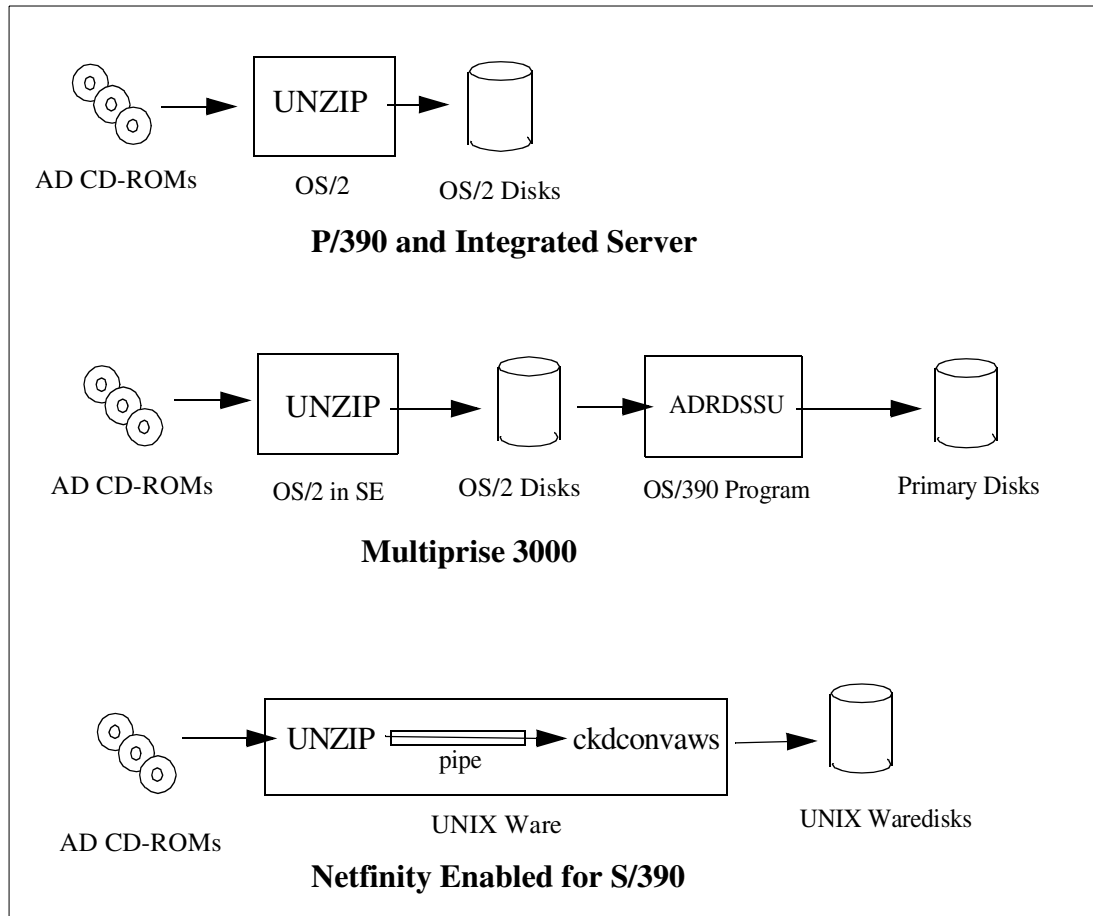


Figure 1-1 AD system installation processes

With an MP3000, the unzipped files on the OS/2 disks¹² can be used directly (via emulated I/O) for running OS/390 or z/OS. This is typically done for a short period during the installation process. There is not enough space on the OS/2 disks for all the OS/390 volumes. Also, emulated I/O from the OS/2 disks is considerably slower than I/O from the MP3000 primary disks. For these reasons, the volumes are normally copied from the OS/2 disks to the primary disks. (They could be copied to channel-attached 3390 drives instead, but we will describe the most common case here.) It is necessary to unzip the CD-ROM files to the OS/2 disks because there is no method to unzip directly from the CD-ROMs to the primary disks.¹³

¹² All MP3000 machines have three disk drives that belong to OS/2. There is approximately 14 GB free space on these drives.

¹³ OS/2 cannot access the primary S/390 disks. OS/390 cannot access the CD-ROM drive or files. The OS/2 disks can be accessed by both OS/2 and OS/390 (via emulated CKD I/O).

1.5.1 Complete installation or release migration

The installation processes described in the rest of this chapter assume you are installing a new OS/390 system—either on a new machine or to completely replace an existing, older OS/390 release. The installation processes (using the `install` command or issuing your own `unzip` commands) do not delete any existing S/390 volumes. With one restriction, you can have several OS/390 releases installed at the same time. With one exception, the 3390 volume serial numbers used for each release are unique and do not conflict with earlier releases.

Typically, P/390-based machines do not have enough disk space to hold multiple OS/390 releases, while MP3000 systems may have enough space in some cases. In either case, you must manage your disk space; the installation processes will not do it for you. Stated another way, the installation processes will not overlay existing OS/2 files (unless an existing file has exactly the same OS/2 name as a new file). In the simplest case, you might delete all the S/390 volumes used by the previous release (thus freeing space for the new release). You would not delete your own, locally-defined S/390 volumes, of course. These should be release independent. The P/390 configurator program (or the emulated I/O configuration panel on the MP3000) is normally used to delete emulated volumes.

The exception and restriction mention above involves the OS39M1 volume. This same volser has been used for several AD releases and is used in OS/390 R10 and z/OS R1.1. This volume contains various VSAM data sets (paging data sets, for example), the master catalog, the RACF data base, SYS1.PROCLIB, SYS1.PARMLIB, SMF data sets, TCP/IP controls, and so forth. From a system viewpoint, most of your local customization affects data sets on this volume. The most important components are the master catalog and the RACF data base.

If you simply replace your old OS39M1 with the new OS39M1 in the latest AD release, you will lose all your local customization. That is, all your local RACF data (including local userids) would be lost and anything in the master catalog would be lost. Nevertheless, an informal, unscientific survey a few years ago indicated that at least half of the AD system community updated their systems by simply replacing this volume with the new version.¹⁴ These system owners would then reapply their local changes to the system data sets, redefine local userids, and recatalog local data sets.¹⁵

If you want to preserve existing catalogs, RACF entries, and so forth, you need to use the AD system *migration* process. In general, it works like this:

- ▶ Keep enough of your existing system volumes to IPL your existing system. You might delete your old DLIB volumes, IMS volumes (if appropriate), CICS volumes (if appropriate), and so forth, to make room for the new system volumes.
- ▶ Install the new key system volumes except for OS39M1.
- ▶ Update your existing DEVMAP to include the three new volumes. You can give them any unused 3390 addresses, such as A90, A91, A92.
- ▶ IPL your old system. Verify that the new volumes are online.
- ▶ Log onto your old system and use ISPF to examine SYS1.MIGRATE.R10 (or SYS1.MIGRATE.R01 for z/OS R1.1).
- ▶ Backup your existing OS39M1 volume.¹⁶

¹⁴ When the survey was made this volume was named SCPMV5, but the principle was the same.

¹⁵ The worst exposure in this method is for local VSAM data sets. You cannot simply recatalog them. If you constructed local user catalogs, as urged several places in this document, then this exposure for VSAM data sets is minimized. Of course, if you have no locally-defined VSAM data sets, then this is not a problem.

¹⁶ There are several ways to do this: OS/390 backup to tape, copy the OS/2 files under another name (if you have enough OS/2 disk space), back up the OS/2 files to tape using the XTAPE program or something similar, and so forth.

- ▶ Read the \$README member in SYS1.MIGRATE carefully. It describes how to run the jobs in the other members to upgrade your existing OS39M1 volume for use with the new release.
- ▶ Run these jobs carefully.
- ▶ IPL your new system volumes (using the old OS39M1, now upgraded to a new level) and run any remaining *post migration* jobs in SYS1.MIGRATE.
- ▶ When you are satisfied with the new system, delete the remaining old system volumes.
- ▶ You might then change the addresses of the new system volumes to match the assumed addresses, such as A80 for the IPL volume, and so forth. This is not required, but is easily done with the P/390 Configurator panels (or the MP3000 IOCDs definitions, or the FLEX-ES system definition file).

The migration process has several limitations and considerations, such as:

- ▶ It assumes you have not altered any data sets on the old system volumes, other than the OS39M1 volume. Alternations on other volumes are simply lost. This includes alternations to the system HFS volume, such as mount points for HFS data on non-system volumes. It is usually simple to redefine the mount points after installing a new system HFS volume, but you need to keep track of what was added to the old volume.
- ▶ The migration process was designed to migrate release *n* to release *n+1*. Attempting to migrate several releases in a single step usually does not work very well.

An alternative to the AD migration process is to establish user catalogs under an existing AD system and be certain all your local data sets—especially VSAM data sets—are cataloged in these catalogs. You can then completely overlay your old AD system with a new one and connect your user catalogs to the new master catalog. You would still need to reestablish all your RACF changes, HFS mount points, PARMLIB changes, and so forth, for the new system. While this may sound difficult, it is fairly practical and is used by a portion of the AD community.

1.5.2 AD installation for P/390 or Integrated Server

A complete *replacement* installation (that is, ignoring any *migration* process) on a P/390 is straightforward and involves these steps:

- ▶ Deciding which volumes to install. This may be a balance between your available disk space and which portions of the AD system you intend to use. For example, if you do not use IMS, you would probably not install the IMS volume. If you are very short on disk space, you might elect not to install the DLIB volumes. The specific volumes distributed for OS/390 R10 and z/OS R1.1 are described in the next chapters.
- ▶ Unzipping the volumes from the CD-ROMs. You would probably unzip enough volumes to create an IPLable, working system; for current AD releases, this requires unzipping four 3390 volumes. You can unzip additional volumes later, for DLIBs, IMS, DB2, and so forth.
- ▶ Providing a DEVMAP that defines the volumes to the P/390 programs. A new DEVMAP is provided on the CD-ROMs, or you might update one of your existing DEVMAPs, or you might create a new one.
- ▶ IPLing the system, verifying operation, and doing minor customization.

For R10, an `install` command (to be executed in an OS/2 window) is provided on the first CD-ROM. It assumes that all unzipped volumes will be installed on a single OS/2 drive (and defaults to the E drive). The command script asks several questions, such as:

- ▶ What is your target OS/2 drive?
- ▶ Do you want to install the DLIB volumes?

- ▶ Do you want to install the DB2 volume?
- ▶ And so forth

The install command is started from an OS/2 window by entering `x:\install` where x is the drive letter for the CD-ROM drive. It unzips the first OS/390 volume (from the first CD-ROM), requests you to mount the second CD-ROM, unzips it, and so forth.

For z/OS R1.1, the first CD contains `ZCOPY.COMD` and a series with `CD1.COMD`, `CD2.COMD`, and so forth. The `CD1.COMD` file issues an `unzip` command to install CD number 1 on your G drive. There is a command for each CD, although the command file for the OS39M1 volume is named `OS39M1.COMD`. The `ZCOPY` command simply issues the series of `CD1`, `CD2`, ... commands (all pointing to your G drive).

An alternative to the `install` or `ZCOPY` commands is to issue `unzip` commands directly. This has the advantage that the unzipped files can be directed to different OS/2 drives, and that you can more directly select which volumes to unzip. All of the CD-ROMs in the R10 AD contain directories named `OS390` and the zipped volumes are in these directories. The R1.1 set uses directory names `ZOS11`. Listing the directory contents provides the name of the zipped file. For example, the first CD-ROM of the R10 release contains the following files in the `OS390` directory:

```

DEVMAP.MVS      (DEVMAP for this MVS)
DEVMAP.NME      (a prototype DEVMAP, not normally used)
ICKDSF.IPL      (stand-alone restore program, in AWSTAPE format)
MIGRATE.DOC     (initial documentation for migration from previous AD release)
OS39RA.ZIP      (the first zipped volume)
README.MVS      (minor documentation)
SADSS.IPL       (stand-alone DASD initialization program, in AWSTAPE format)

```

The root directory of *each* CD-ROM (in the AD set) contains an `unzip` program. To unzip the first volume to your OS/2 E drive, for example, assuming the CD-ROM is drive H, use the command:

```

h:\unzip h:\os390\os39ra.zip *.* -d E:\           (for OS/390 10)
h:\unzip h:\zos11\z1res1.zip *.* -d E:\         (for z/OS 1.1)

```

The same command (except for the zipped file name and, perhaps, the target drive) would be used for each volume to be unzipped.

As described in more detail in the next chapter, you need to unzip four volumes in order to have an IPLable OS/390 R10 system or z/OS R1.1 system. These four volumes are on the first four CD-ROMs. The AD design assumes you will use addresses¹⁷ `A80`, `A81`, `A82`, and `A87` for these volumes. `A80` is the IPL volume, and `A82` contains the operational IODF and `SYS1.IPLPARM` data set. Of course, they could be used at any addresses defined as 3390 devices in the OS/390 IODF, but we will assume they are used at the addresses listed here.

A minimal P/390 DEVMAP (release 10) could look like this:

Addr	Device	Label	Atype	Size	Mgr	FN/P
0560	3480				G	
0700	3278		DSPY		3	/R=OP
0701	3278		DSPY		3	
0A80	3390	OS39RA	CKD	3339C	2	E:\OS390\OS39RA_1.A80
0A81	3390	OS3RAA	CKD	3339C	2	E:\OS390\OS3RAA_1.A81
0A82	3390	OS39M1	CKD	3339C	2	E:\OS390\OS39M1_1.A82
0A87	3390	OS39HA	CKD	2226C	2	E:\OS390\OS39HA.A87

¹⁷ The more correct term may be *device numbers*, although P/390 systems do not use an IOCDS mechanism to translate hardware addresses (chpids, CU, and UA numbers) to device numbers. The MVS community commonly uses *address* (taken from the original S/360 machines) to mean *device number* (as defined in an IOCDS).

```

0E20 3088          A
0E21 3088          A

```

```

2 = AWSCKD 3 = AWS3274 A = LCS3172 G = SCSI3480

```

This document is not intended as an introduction to P/390 systems. Very briefly, this DEVMAP illustrates the following details:

- ▶ The DEVMAP (in approximately the format shown here) is accessed through the *Configurator* icon in the P/390 window on the OS/2 desktop.
- ▶ The 560 address is for the 4mm tape drive that is included with all P/390 (and MP3000) systems. This line is not required to IPL the system, but would normally be included.
- ▶ The 700 address is the MVS master console. All the AD releases use this address for the MVS master console. The /R=OP operand specifies that a local TN3270 session must specify LUNAME=OP. This is a method for forcing a particular S/390 address (700) to a specific TN3270 session.
- ▶ The 701 address provides a TSO terminal through a local TN3270 session.
- ▶ The OS/2 file names for the four 3390 volumes contain names such as OS39RA_1.A80. The A80 suffix is just a name but serves as a reminder of the assumed address of the volume. The _1 suffix in the first part of the file name indicates that the volume is larger than 2 GB. The maximum OS/2 file size is 2 GB. An emulated volume larger than this is automatically split into two OS/2 files. In this example, the OS/2 file names are OS39RA_1.A80 and OS39RA_2.A80. Both these files were in the CD-ROM zipped file. The emulated I/O device managers handle the transition between the two files automatically. The OS39HA volume is less than 2 GB, so it is a single OS/2 file.
- ▶ The E20 and E21 addresses are for use by OS/390 TCP/IP. They are not required to IPL the system and could be omitted from an initial DEVMAP.
- ▶ Initial use of the system, through this DEVMAP, will be through two local (that is, running on the P/390 PC and using OS/2 TCP/IP loopback connections) TN3270 sessions.
- ▶ You can overwrite a few fields in a DEVMAP, such as the OS/2 file names for emulated CKD volumes. Press Enter after changing a field (and while the cursor is still in that field).
- ▶ You may have any number of DEVMAP files on your OS/2 system. The OS/2 command **AWSPROF x:\file.nam** is used to specify the file name of the *current* DEVMAP. A full path name must always be specified with the AWSPROF command. The current DEVMAP is automatically used by the P/390 configurator and device managers. After issuing the AWSPROF command, the DEVMAP may be accessed by clicking the Configurator icon in the P/390 window, or by issuing an **AWSCFG** command in an OS/2 window.¹⁸ *Do not attempt to edit a DEVMAP with a conventional editor; they are not text files and an editor will destroy a DEVMAP.*

Installing the four emulated volumes does not involve *creating* emulated drives (using P/390 Configurator options) or initializing the drives with DFDSS. When the AD system was being built (at IBM) the corresponding emulated drives were created and initialized. The creation parameters and initialization effects were captured when the original OS/2 files were zipped and distributed. Unzipping the files and pointing to them in your DEVMAP is all that is required. If you add local work volumes (such as the WORK01 and WORK02 volumes often used for examples in this redbook), you begin by creating the disks (with the Configurator) and initializing them (usually with ICKDSF under OS/390).

¹⁸ The first Configurator screen asks for a password. The default is a blank field. Very few P/390 owners set passwords for their DEVMAPs.

If you do not have an existing DEVMAP, we suggest you copy DEVMAP.MVS from the first CD-ROM. The `install` command (R10) will automatically copy it to your target drive,¹⁹ or you can copy one from the first CD. This DEVMAP contains definitions for all the volumes in the AD release. You can delete the unused volumes from the DEVMAP or simply leave them in the DEVMAP. A disk definition in a DEVMAP that does not point to an existing OS/2 file is, in effect, a dummy entry and does not hurt anything.

Before IPLing the system in our example, you must ensure that two TN3270 sessions will be ready for use on the P/390 display. Typically, these use the IBM eNetwork Personal Communications TN3270 package (commonly known as PCOM), connected to the loopback address (127.0.0.1), port 7490 (a standard P/390 port number).

After unzipping enough volumes to IPL, verifying that the current DEVMAP points to these volumes, and starting the PCOM sessions, you can IPL the system. Using a normal setup, the parameters are:

```
IPL ADDRESS = 0A80          IPL PARAMETER = 0A82CS
```

The CS component of the IPL parameter triggers a CLPA and JES2 cold start. The next chapter describes the various IPL parameters supplied with release 10.

1.5.3 AD installation for MP3000

As illustrated in Figure 1-1 on page 6, the installation process for an MP3000 is more complex than for a P/390. The basic problem is this:

- ▶ The AD system is distributed on CD-ROMs, but OS/390 (or z/OS) cannot directly access CD-ROMs.
- ▶ The normal goal is to place the AD volumes on the primary disks of the MP3000.²⁰
- ▶ The Support Element (SE) in the MP3000, which is an internal PC running under OS/2, can access the CD-ROM drive but cannot access the primary disks of the MP3000.
- ▶ Both the SE and OS/390 (via emulated I/O definitions) can access the OS/2 disks.²¹

For these reasons, the installation process involves *staging* steps. Volumes are unzipped to the OS/2 drives (using the same OS/2 program the P/390s use), and then (using an OS/390 or z/OS program) the volumes are copied from the OS/2 disks to the primary disks. The copies in the OS/2 disks are then deleted. The second copy is quite fast; installing this way takes just a little longer than a simple P/390 installation.

A standard MP3000 system has approximately 14 GB free space on the OS/2 drives. This space is used for the staging steps just described. The 14 GB is not enough space for all the system volumes in recent releases, so they need to be staged in groups. If an OS/390 system (the *old* system) already is operational on the primary drives, and if this old system is to be retained during the installation of the *new* system, then the old OS/390 system can be used for the second copy steps.²² MP3000 systems usually have sufficient disk space for an old OS/390 and a new OS/390 or z/OS, provided there is not a large amount of user data on the disks. The exact situation will be different for each MP3000 owner.

¹⁹ Most P/390 owners keep their DEVMAPs on the D drive. However, a DEVMAP can be on any drive. The P/390 programs find it when you issue an `AWSPROF` command pointing to it.

²⁰ This is not required. It is possible to run the system with the system volumes resident on emulated drives on the OS/2 disks. However, the primary drives are considerably faster and most MP3000 owners elect to place the system volumes on the primary disks.

²¹ The three Support Element disks (often referenced as the OS/2 disks) are separate from the primary disk drives in an MP3000 system.

²² If you do this, you must deal with two OS39M1 volumes--the old one and the new one.

If, instead, you elect to delete (or overwrite) your existing old OS/390 system on the primary disks, you need to stage enough of your new system to the OS/2 drives so you can IPL the new system and use it to copy itself (that is, the four volumes needed to IPL) from the OS/2 emulated disks to the primary disks. This is the more general process, since it does not depend on the existing contents of the primary disks, and is what we describe here.

The installation steps are:

- ▶ Do whatever is necessary to free at least 11 GB disk space on the OS/2 G drive.
- ▶ Unzip the first four CD-ROMs to the G drive, as described under “AD installation for P/390 or Integrated Server” on page 8.
- ▶ Select an IOCDS (A0 - A3) to create a DEVMAP and IOCDS that contains these volumes plus the primary disks where you will eventually place these four volumes.
- ▶ Perform a power-on reset (POR) to activate the new DEVMAP and IOCDS
- ▶ IPL the new system (on the emulated disks), log onto TSO, and run jobs to initialize four primary drives (if necessary) and copy the four emulated I/O volumes (that you unzipped) to the four primary disks.²³
- ▶ IPL from the primary disks to verify that the new system works. During IPL, there will be error messages about the duplicate volsers on the emulated I/O disks. Reply to these messages, varying the emulated I/O drives offline.
- ▶ Stop the operating system, and then use the Emulated I/O Configuration panel to delete the four emulated I/O volumes. This is necessary to free space on the OS/2 disks so that more volumes can be staged. It might be necessary to *deactivate* the MP3000 (an icon on an SE panel) before you can delete the emulated I/O volumes.
- ▶ Unzip more volumes from the CD-ROMs. The next chapters discuss the contents of the various volumes.
- ▶ Update the DEVMAP to point to these volumes and add whatever drives are necessary to the IOCDS.
- ▶ Perform a POR to activate the new DEVMAP and IOCDS.
- ▶ IPL your new system (on the primary disks). Log onto TSO and submit jobs to initialize additional primary volumes (if necessary) and to copy the second set of volumes from the emulated I/O disks to primary disks.
- ▶ Vary the new volumes (on the primary drives) on line and verify that they look correct.
- ▶ Delete the emulated I/O volumes, using the Emulated I/O Configuration panel.
- ▶ If necessary, stage a third set of volumes in the same manner.

Three groups of staging are needed to install all the volumes. If you do not need all the volumes, then the third group will probably not be needed.

A usable DEVMAP (R10) for the first staging group (containing the four volumes needed to IPL release 10) might look like this:

Addr	Device	Label	Atype	Size	Mgr	FN/P
40	3390	OS39RA	CKD	3339C	2	G:\OS390\OS39RA_1.A80
41	3390	OS3RAA	CKD	3339C	2	G:\OS390\OS3RAA_1.A81
42	3390	OS39M1	CKD	3339C	2	G:\OS390\OS39M1_1.A82
47	3390	OS39HA	CKD	2226C	2	G:\OS390\OS39HA.A87
70	3278		DSPY		3	/R=0P
71	3278		DSPY		3	

²³ This involves copying active system volumes, something that is not normally recommended for integrity reasons.

In this case, for an MP3000, the Addr field in a DEVMAP specifies a Unit Address (UA) that is then used in the IOCDs definitions. For emulated I/O, these unit addresses are arbitrary values between 00 and FE.

A matching IOCDs definition might be:

```
ID MSG1='ITSO A3',MSG2='STAGING SETUP',SYSTEM=(7060,1)
CHPID PATH=(FC),TYPE=EIO
CHPID PATH=(FD),TYPE=DSD
* FIRST SIXTEEN PRIMARY DISKS (LOGICAL VOLUMES) ADDRESS AS AA0 - AAF
CNTLUNIT CUNUMBR=FD00,PATH=(FD),CUADD=00,UNITADD=((00,16)),UNIT=3990-2
IODEVICE CUNUMBR=FD00,ADDRESS=(AA0,16),UNIT=3990,UNITADD=00
* EMULATED I/O DISK VOLUMES
CNTLUNIT CUNUMBR=FC40,PATH=(FC),UNITADD=((40,1)),UNIT=3990-2
IODEVICE CUNUMBR=FC40,ADDRESS=A80,UNIT=3990,UNITADD=40
CNTLUNIT CUNUMBR=FC41,PATH=(FC),UNITADD=((41,1)),UNIT=3990-2
IODEVICE CUNUMBR=FC41,ADDRESS=A81,UNIT=3990,UNITADD=41
CNTLUNIT CUNUMBR=FC42,PATH=(FC),UNITADD=((42,1)),UNIT=3990-2
IODEVICE CUNUMBR=FC42,ADDRESS=A82,UNIT=3990,UNITADD=42
CNTLUNIT CUNUMBR=FC47,PATH=(FC),UNITADD=((47,1)),UNIT=3990-2
IODEVICE CUNUMBR=FC47,ADDRESS=A87,UNIT=3990,UNITADD=47
* TWO EMULATED 3270S FOR THE MASTER CONSOLE AND TSO
CNTLUNIT CUNUMBR=FC70,PATH=(FC),UNITADD=((70,1)),UNIT=3174
IODEVICE CUNUMBR=FC70,ADDRESS=700,UNITADD=70,UNIT=3270,MODEL=X
CNTLUNIT CUNUMBR=FC71,PATH=(FC),UNITADD=((71,1)),UNIT=3174
IODEVICE CUNUMBR=FC71,ADDRESS=701,UNITADD=71,UNIT=3270,MODEL=X
```

Background information for these definitions is described in the Redbooks *Multiprise 3000 Technical Introduction*, SG24-5633 and *Multiprise 3000 Basic Emulated I/O Definitions*, SG24-5669; we do not attempt to describe the general use and syntax of IOCP/IOCDs statements here. There are a few key elements in this example:

- ▶ Sixteen primary disks (logical volumes) are defined. These correspond to Unit Addresses 00 - 0F (as established through the Internal Disk Subsystem panels on the Support Element). The IOCDs definitions here define these volumes as addresses (device numbers) AA0 - AAF. This is partly arbitrary; although we must use addresses that are defined in the IODF of the OS/390 system. The AD system has several ranges of 3990 addresses defined (including A80 - ABF). The predefined IODF addresses are discussed in "I/O configuration for OS/390" on page 15 and in the next chapter. We defined sixteen drives for convenience; we will not use all of them during the AD installation and some of them might not exist.²⁴
- ▶ We defined the four emulated I/O drives, corresponding to the four in the DEVMAP; notice that the Unit Addresses in the IOCDs match the unit addresses in the DEVMAP. Emulated I/O devices are required to have individual CNTLUNIT and IODEVICE statements for each device.
- ▶ We defined two emulated 3270 devices. One will be the OS/390 master console and the other will be a TSO terminal. Both these sessions will use the PCOM TN3270 emulator running on the Support Element.

Once the first four volumes are unzipped, the DEVMAP and IOCDs build, and a POR performed to activate them, the system can be IPLed using:

```
IPL ADDRESS = A80      IPL PARAMETER = 0A82CS
```

After the system is running and you are logged onto TSO, you could initialize four primary volumes with this job:

²⁴ That is, logical volumes may not have been defined for all sixteen Unit Addresses. Logical volumes, for the primary disks, are defined with the Internal Disk Subsystem panels of the Support Element application.

```

//OGDEN1 JOB 1,OGDEN,MSGCLASS=X
// EXEC PGM=ICKDSF,REGION=8000K
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
INIT UNIT(AA0) NOVAL NVFY VOLID(DSKAA0) VTOC(0,1,14)
INIT UNIT(AA1) NOVAL NVFY VOLID(DSKAA1) VTOC(0,1,14)
INIT UNIT(AA2) NOVAL NVFY VOLID(DSKAA2) VTOC(0,1,14)
INIT UNIT(AA7) NOVAL NVFY VOLID(DSKAA7) VTOC(0,1,14)
/*

```

The target drives must be offline before the job is submitted. Again, we stress that you do not need to use these specific addresses (AA0, AA1, AA2, AA7). You do need to use addresses that are defined as 3390 drives in the AD system IODF. The addresses shown here are used consistently throughout this book (and other MP3000 Redbooks). Other documentation sometimes uses 3390 addresses starting with 0300, although these are not defined in the default IODF of the release 10 AD system.

Once the target disks are initialized, you could copy the system volumes with the following job (using the R10 volsers as examples):

```

//OGDEN2 JOB 1,OGDEN,MSGCLASS=X
// EXEC PGM=ADRDSU
//SYSPRINT DD SYSOUT=*
//D1 DD UNIT=3390,DISP=OLD,VOL=SER=OS39RA
//T1 DD UNIT=3390,DISP=OLD,VOL=SER=DSKAA0
//D2 DD UNIT=3390,DISP=OLD,VOL=SER=OS3RAA
//T2 DD UNIT=3390,DISP=OLD,VOL=SER=DSKAA1
//D3 DD UNIT=3390,DISP=OLD,VOL=SER=OS39M1
//T3 DD UNIT=3390,DISP=OLD,VOL=SER=DSKAA2
//D4 DD UNIT=3390,DISP=OLD,VOL=SER=OS39HA
//T4 DD UNIT=3390,DISP=OLD,VOL=SER=DSKAA7
//SYSIN DD *
COPY INDD(D1) OUTDD(T1) ALLDATA(*) ALLEXCP COPYV ADMIN
COPY INDD(D2) OUTDD(T2) ALLDATA(*) ALLEXCP COPYV ADMIN
COPY INDD(D3) OUTDD(T3) ALLDATA(*) ALLEXCP COPYV ADMIN
COPY INDD(D4) OUTDD(T4) ALLDATA(*) ALLEXCP COPYV ADMIN
/*

```

The target volumes will be automatically varied offline after they are copied because they will contain duplicate volsers, duplicating the four emulated I/O volumes that you IPLed.

Without changing the DEVMAP or IOCDS, you can IPL the new volumes with:

```
IPL ADDRESS = AA0      IPL PARAMETER = 0AA2CS
```

The CLPA produced by this IPL parameter is probably not necessary, but it does not hurt anything. At this point, you have an IPLable version of your new AD system resident on the primary disks. You can then delete the emulated I/O volumes and stage another set of AD volumes as outlined earlier.

Using this process, we have found that we can easily install a new AD system on an MP3000 in an afternoon. Your final DEVMAP and IOCDS definitions should include all the emulated I/O devices you plan to use, such as LAN-connected emulated 3270 sessions, OS/390 TCP/IP LAN connections, the 4mm tape drive, and so forth.

1.5.4 AD installation for Netfinity/EFS

The installation process for the OS/390 AD CD-ROM release 10 system, on a Netfinity x/Series EFS machine, is covered in considerable detail in the redbook *Netfinity Enabled for S/390*, SG24-6501 (expected to be available in late 2001), and will not be repeated here.

1.6 I/O configuration for OS/390

By I/O configuration we mean the device addresses (“device numbers”) assumed for the distributed AD system and defined in the IODFs provided with the distributed system. These are reflected in distributed DEVMAPs, documented IOCDs, other documentation, OS/2 file names, and so forth. These addresses can be changed, but it is most convenient to use them as distributed. As a brief overview, the following devices and addresses should be present in order to install and initially use the AD system.

Device	Addresses	Use
3390s	A80 - A9F AA0 - ABF	System volumes, user volumes. Documentation assumes IPL volume is A80
Local 3270X	700	MVS master console
Local 3270s	701 - 71F 900 - 91F	VTAM sessions (TSO, CICS, etc)
CTC (or 3088)	E20 - E21	LAN interface for OS/390 TCP/IP
CTC (or 3088)	E40	LAN interface for SNA

The CTCs are not needed for initial operation, but most users like to activate OS/390 network connections as soon as possible.

How these devices are provided depends on the system being used.

- ▶ For P/390 systems, all devices are emulated I/O devices. The emulated 3390s use OS/2 files. The local 3270s are actually TN3270 connections to OS/2 TCP/IP; the emulated device manager makes these look like channel attached, non-SNA 3270 connections. The CTC devices are emulated through OS/2 LAN connections and appear to OS/390 as 3172 devices.
- ▶ For MP3000 systems:
 - The 3390s could be “real” 3390s connected by ESCON or parallel channels, internal primary drives (not connected to OS/2), or OS/2 emulated I/O drives. In general, we will assume they are internal, primary drives.
 - The local 3270s could be through channel-attached “real” 3174 control units to “real” 3270 terminals, or they could be emulated terminals through OS/2 device managers. In the latter case, the terminals are actually TN3270 sessions connected to OS/2 TCP/IP. In this document, we will assume they are emulated devices.
 - The CTCs could be “real” channel-attached 3172 control units, or emulated through OS/2 device managers. We will assume they are emulated units.
- ▶ For Netfinity/EFS systems, the devices could be “real” units connected through channel adapters manufactured by Fundamental Software, Incorporated²⁵, or emulated units

²⁵ At the time of writing, the FLEX-ES ESCON adapters were not yet available and the FLEX-ES parallel channel adapters did not support external DASD devices. Both restrictions are expected to change.

managed by FLEX-ES emulation programs. In our case, the 3390s are emulated using UNIX files, the local 3270s are emulated using TN3270 connections to a FLEX-ES module, and the CTCs are emulated using another FLEX-ES function that emulates IBM 3172 control units.

As a side note, the Integrated Server supports ESCON and fast parallel channel adapters²⁶, and the other P/390E machines support fast parallel channel adapters.²⁷ These could be used to attach external “real” devices, but this is option is seldom used for initial system setup. (They are typically used for later connections to existing S/390 devices.) The use of channel adapters by P/390-based systems is ignored unless expressly mentioned in a few of the discussions in this document.

Recent AD system releases have had the same general pattern for OS/390 IODF device definitions. In general, these definitions include:

Addresses	Device type
00C	2540 card reader (used for emulated I/O functions)
00E	1403 printer (useful for emulated I/O printing)
120 - 15F	3380 disk drives
240 - 25F	3380 disk drives
260 - 27F	3390 disk drives
560 - 56F	3480 tape drives
580 - 58F	3400 (round tapes) tape drives
700	3270 (for the MVS master console)
701 - 73F	local non-SNA 3277s (for VTAM terminals)
900 - 91F	local non-SNA 3277s (for VTAM terminals)
A80 - ABF	3390 disk drives
E20 - E27	CTC devices (for LAN interfaces)
E40 - E47	CTC devices (for LAN interfaces)

More specific lists are provided in the next chapters. The SYS1.IPLPARM(LOADxx) members distributed with the AD system are usually set up to use an IODF that includes the devices listed here. This is usually SYS1.IODF01. Recent releases also include SYS1.IODF99, and this contains more 3390 and tape definitions. However, the distributed LOADxx members do not use IODF99; you need to modify them if you want to use it.

The intention was that IODF99 should be used for MP3000 systems.²⁸ IODF99 defines more 3390 drives (including addresses starting with 300) and more types of tape drives. If neither of these are important, you can use the default IODF01. All the examples in this document assume the use of IODF01.

The initial AD documentation assumes the use of certain addresses, such as A80 for the IPL volume. For the MP3000, we need two IPL volumes during the staging activity. We elected to make A80 the emulated I/O volume and AA0 the primary volume that will be the “final” IPL volume. We could have just as easily reversed these address definitions. Our choice was arbitrary, but it is used consistently throughout this document (and in other Redbooks). Our commonly used addresses, using release 10 volsers, are:

:

Primary Disk		Emulated I/O		VOLSER	Use
U A	Address	U A	Address		
00	AA0	40	A80	OS39RA	IPL volume, system libraries

²⁶ These are commonly known as the Escimo and Parca adapters.

²⁷ Earlier P/390 machines had an optional “slow” parallel channel adapter that could not be used for external DASD connections.

²⁸ There is no particular reason for using a separate IODF for MP3000 machines. The original intention was to design an improved IODF that could be used for all AD systems, but effort is still underway.

Primary Disk		Emulated I/O		VOLSER	Use
U A	Address	U A	Address		
01	AA1	41	A81	OS3RAA	more system libraries
02	AA2	42	A82	OS39M1	IODF, IPLPARM, VSAM, paging, spool
03	AA3	43	A83	OS39PA	DB2 libraries (including DLIBs)
04	AA4	44	A84	OS3PAA	CICS (including DLIBs)
05	AA5	45	A85	OS39DA	system DLIBs
06	AA6	46	A86	OS3DAA	system DLIBs
07	AA7	47	A87	OS39HA	system HFS data sets
08	AA8	48	A88	OS3DAB	system DLIBs
09	AA9	49	A89	OS3PAB	IMS (including DLIBs)
0A	AAA	4A	A8A	OS3PAC	WebSphere components
0B	AAB	4B	A8B	OS3PAD	WebSphere components
		70	700		AWS3274 emulating local 3270. MVS console
		71	701		AWS3274 emulating local 3270. VTAM/TSO
		72-92	900-91F		LAN3274 emulating local 3270. VTAM/TSO
		10-11	E20-E21		CTC interface for OS/390 TCP/IP
		20	560		4mm tape drive, emulating 3480

This table may be a little confusing. It says, for example, that if volume OS39RA is on an emulated I/O disk (P/390s, or the OS/2 disks of the MP3000), then it is addressed as A80. It uses Unit Address 40 in the DEVMAP and the IOCDs connects this Unit Address (on chpid FC) to device number A80. (P/390s have no Unit Address; their DEVMAP contains the final device number (“address”), which would be A80 in this case.) If volume OS39RA is on an MP3000 primary disk, it is addressed as AA0. In this case the IOCDs would equate Unit Address 00 (on chpid FD) to device number AA0.²⁹

During the MP3000 staging process (when installing an AD system), there will be an OS39RA volume at both addresses (A80 and AA0). You could IPL from either one. Since OS/390 will not permit duplicate volume serial numbers online, it will force you to vary the “other” drive offline.

Examples in later sections of this document might reference the IPL volume as A80 (if the example was tested on a P/390) or AA0 (if the example was tested on an MP3000). Except in special circumstances, it does not matter whether an emulated I/O volume (A80, in this discussion) or a primary volume (AA0) is used.

²⁹ The Unit Addresses for primary disks are assigned by the Internal Disk Subsystem panels on the Support Element. The Unit Addresses in this table are what the author uses on his MP3000, and are listed for completeness and consistency with other parts of this document. Your MP3000 primary disks may be defined differently. For more information about MP3000 primary disks, see the *Multiprise 3000 Technical Introduction* redbook.

1.6.1 Changing your I/O configuration

The I/O configuration of OS/390 running on an MP3000, for example, is set in three places: an IODF, an IOCDS, and a DEVMAP. The configuration for a P/390 is set by an IODF and a DEVMAP. The only way to construct or change an IODF is with the HCD utility. An IOCDS can be built with the HCD utility, or can be built directly through SE panels. A DEVMAP can only be built with P/390 or SE panels.

Experienced OS/390 systems programmers usually build their IODF and IOCDS at the same time, using HCD. (These experienced systems programmers probably never built a DEVMAP, because these do not exist on machines larger than the MP3000.) HCD is a powerful utility, but it is not usually regarded as intuitive and user friendly. Many AD system owners do not have the skill (or time to acquire the skill) to use HCD, and elect to use a different approach for managing I/O configuration changes.

Several IODFs are usually provided with the AD system. (See “IODFs and addresses” on page 33 for specific details about release 10.) These IODFs always define a large I/O configuration, with many more devices than you are likely to have on a small machine. Rather than changing an IODF (which would require HCD skills), many system administrators adjust their DEVMAP (P/390s), IOCDS and DEVMAP (MP3000), or resource definitions (Netfinity/EFS) to match one of the IODFs provided with the AD system.

For example, assume you need to create another 3390 volume. The IODFs distributed with the AD system have a large number of addresses (device numbers) already defined as 3390 devices. If you select one of these already-defined addresses for your new volume (avoiding addresses you are already using, of course), there is no need to change the IODF. By slightly changing the IOCDS (and DEVMAP, if the new volume uses emulated I/O), you can assign the new volume to an address (device number) that already exists in the IODF. To do this, you need to understand how to work with IOCDS and DEVMAP definitions; you will need to do this, in any event.

Other redbooks, referenced earlier, discuss IOCDS and DEVMAP concepts and definitions in some detail. You should check those documents for more information.

1.7 Initial system use

Be certain you have the four volumes required to IPL. For release 10, these are OS39RA, OS3RAA, OS39M1, and OS39HA; for z/OS R1.1 these are Z1RES1, Z1RES2, OS39M1, and Z1USS1. Be certain your DEVMAP (P/390) or your DEVMAP+IOCDS (MP3000) is set correctly. For an MP3000, you need to perform a Power On Reset (POR) function with the correct IOCDS selected. The IPL address (0A80 or 0AA0 in our examples) and IPL parameter (0A82CS or 0AA2CS for your first IPL) can be set in the configurator F4 frame (P/390), or in the LOAD frame in the SE window (MP3000).

The complete IPL process, especially when using parameters that start DB2, CICS, IMS, and so forth, produce many console messages. Some of them may be error messages, and some may even look like serious error messages. Many of these are expected and will not harm the system; they are associated with initial startup, or with choices that were made when building the AD system. There is no easy guide for interpreting OS/390 console messages and knowing what can be ignored and what is cause for alarm.

Initial userids are included with the system. The key ones are:

Userid	Password	Authority	Purpose
P390	P390	SPECIAL, OPERATIONS	System administration
IBMUSER	SYS1	SPECIAL, OPERATIONS, SYS1.UADS	Emergency use if P390 userid or RACF is broken
P390B - P390Z	TEST	normal user	For initial users

The P390 or IBMUSER userid is normally used for all early administration, after installing an AD CD-ROM system. The IBMUSER userid can also be used if RACF is not functional because it (IBMUSER) is also defined in SYS1.UADS.

Additional characteristics of these userids include:

- ▶ IBMUSER (RACF definition)
 - Attributes: SPECIAL, OPERATIONS
 - Significant TSO authorities: OPER, ACCT, MOUNT
 - Password interval: N/A
 - Group: SYS1
 - Size : 4096, Maxsize: 2096128
 - Unit: 3390
 - UID: 0
 - Home: /u/ibmuser
- ▶ IBMUSER (SYS1.UADS definition)
 - Attributes: OPER, ACCT, JCL, MOUNT, RECOVER
 - Password (in UADS) IBMUSER
 - Allowed procedures:
 - IKJACCNT: Size: 4096, Unit: SYSALLDA
 - ISPFPROC: Size: 6072, Unit: (none)
- ▶ P390
 - Attributes: SPECIAL, OPERATIONS, AUDITOR
 - Significant TSO authorities: OPER, ACCT
 - Password interval: N/A
 - Group: SYS1
 - Size: 6072, Maxsize: 2096128
 - Unit: SYSALLDA
 - UID: 0
 - Home: /
- ▶ P390A
 - Attributes: SPECIAL, OPERATIONS, GRPACC
 - Significant TSO authorities: OPER
 - Size: 5999, Maxsize: 6000
 - Unit: 3390
 - No OMVS segment

Userids P390A - P390Z do not have OMVS segments defined.

The AD system provides a large number of IPL options, and these are discussed in the next chapters. If the system volumes are assigned the recommended addresses, initial startup (without DB2, CICS, and so forth) can be done by IPLing address A80, with IPLPARM=0A82CS; for an MP3000 using the primary disks, these values would be AA0 and 0AA2CS. This should load the system and permit you to log on with userid P390 or IBMUSER.

The “CS” portion of the IPLPARM triggers the CLPA function and a cold start for JES2. These need to be done the first time you IPL. Thereafter, if you make no changes to LPA, you can use the parameter 0A8200 (or 0AA200 for primary disks). This will bypass the CLPA and JES2 cold start.

As distributed, the OS39M1 volume is mounted as a STORAGE volume and is the only PUBLIC or STORAGE volume among the distributed volumes. For R10, OS39M1 is approximately 25% full, leaving about 2400 cylinders free. This provides ample scratch space for *initial* system use.

OS39M1 contains data set SYS1.P390.CNTL. You can add members to this data set (or use any existing members, if appropriate) for your initial work, such as the migration jobs described earlier. Many of the members in this data set were used to build the AD system, or earlier releases of the AD system. *Do not assume all the members are relevant to the current release.*

You may notice that there is significant free space on some of the system volumes. We suggest you consider your future plans before using this space. For one thing, anything on the system volumes (other than OS39M1) will be lost if you use the AD migration functions to move to the next release. Also, free space on the system volumes may be needed for PTF or additional program product installation. Provided you understand these considerations, you can use free space on any of the system volumes.

1.7.1 P/390 error messages

The following OS/2 *pop-up* message may appear during IPL:

```
+-----+
|           P/390 Error           |
+-----+
| AWSDEV908E CU=SCSI348: RC=006E |
| opening device driver, DD=SCSI3480 |
+-----+
```

This means that you need to activate the SCSI3480.SYS device driver in CONFIG.SYS. Address 560 (in the DEVMAPs shown earlier) defines a SCSI tape drive. There must be a corresponding definition in the OS/2 CONFIG.SYS. You may need to consult the DOC file (or other P/390 documentation) to help you with the precise CONFIG.SYS statement needed. In the meantime, you can simply delete/cancel this message (using the OS/2 mouse) and continue the OS/390 IPL. You cannot use the OS/390 tape drive associated with the missing SCSI tape drive, but there are no other side effects.

If you receive the following pop-up message:

```
+-----+
|           P/390 Error           |
+-----+
| AWSLCS350S Open error /        |
| AWSLCSDD Device Driver         |
| (Dos open / 110 )              |
+-----+
```

you need to activate the LCS3174.SYS device driver in CONFIG.SYS. The following statement should be in the P/390 section of your CONFIG.SYS. Remove any *rem* characters before the statement.

```
DEVICE=d:\P390\AWSLCSDD.SYS
```

See the LCS3172 DOC file for information about preparing your P/390 for OS/390 TCP/IP. In the meantime, you can delete/cancel this popup message and continue with your IPL.



S/390 V2R10 AD CD-ROM

This chapter discusses specific details of the OS/390 V2R10 AD system. *Chapter 3 is nearly a duplicate of this chapter, but addresses the z/OS R1.1 AD system.* You may want to read this chapter or chapter 3, but you will probably not want to read both.

2.1 Specific product content

The following list contains all the components of the OS/390 release 10 AD system. This list was taken from the Web documentation provided by the PID group in Dallas that currently builds the AD releases. The list is presented here for completeness, but it may not be overly helpful to newer users of OS/390. Many of the listed elements are, in effect, internal parts of OS/390 that you would never work with as separate functions. OS/390 is structured this way for historical reasons. The FMID numbers are used by the maintenance tool, SMP/E, and are sometimes used by systems programmers to indicate exactly which part of the system (and which version of this part) they are discussing.

FMID	DESCRIPTION	FMID	DESCRIPTION
HBB7030	Base Control Program 2.10	HJE7703	JES2 Base
HSWF100	ESCON Director Support	HI01104	I/O Control Program
EMI2220	MICR/OCR	HBD6602	Bulk Data Transfer
JBD6201	Bulk Data Trans file-file	HDZ11F0	DFSMS base V210
HDZ11FH	DFSMS HFS?	JDZ11FB	DFSMS Emu
EER3500	EREP V3R6	HQM4140	High Level Assembler base 1.4
EDU1G01	ICKDSF R16	FDU1G07	ISMF ICKDSF panels
FDU1G08	ISMF ICKDSF Panels	HIF5A02	ISPF base
HBCNCO0	SMP/E SW info base	HTE26D2	TSO/E REXX
JTE26D0	TSO/E EDIT	HFX1112	3270 PC file transfer 1.1.1
HFST101	FFST for MVS 1.2	ETI1106	Terminal I/O Control
H24P111	Host Perf analysis base En	HCS6091	HCD base
JCS6092	HCD HFS	HCM1410	Hardware Config Mgr
HMP1C00	SME/E base	JMP1COB	Planning & Migration Assist
H25I625	Tivoli Frame Endpoint 3.6.1	HCPT2A0	Secure Socket Layer base
HCRY2A0	Open Crypto Serv Facility	HCR7703	ICSF
HLB7703	C Compiler base	JLB7713	C++ Compiler
HLC7703	Open Class Lib source base	HTV7703	C/C++ Open Class Lib base
JTV7713	Applic Support & C Cls Lib	JTV7733	Open Class Library HFS
HLE7703	Language Environment base	JLE77H3	Language Environment HFS
HO8K410	SOM Objects RTE Lib	HDHC170	DCE Application support base
HTKT170	Encina Toolkit Exec base	HDZ11TC	NFS Client
HDZ11TS	NFS Server	HMB3190	DCE base services (OSF DCE 1.1)
HOH2310	DFS (OSF DCE 1.2)	HTCP52A	CS Security Level 2
JTCP5PA	CS High Speed Access Srv	JTCP56A	CS Network Print Facility
JTCP59A	CS HFS	HVT510A	CS SNA
HTCP38X	SecureWay CS X Windows 11.4	HTCP50A	CS IP (TCP/IP)
HEJS120	WebSphere Applic Serv 1.2	HIMW530	HHTP Server base 5.2
HIMN230	TextSearch, NetQuestion Sol	HMR7170	LANRES Host
HQW7500	Lan Server MVS	HOGI300	OSA/SF
HPMG110	USS process manager	HCMG110	USS connection manager
HOT7703	USS Application Serv base USS Shell & Utilities	HSAP2A0	USS integrated call lvl interface USS dbx
HBKM300	READ/MVS 1.3 base	HBKP300	BUILD/MVS 1.2 base
JBKM310	READ/MVS 1.3 English	JBKP310	BUILD/MVS 1.3 English
HBKQ220	BookManager BookServer 2.2	HGD3200	GDDM/MVS base 3.2.0
JGD3219	GDDM/MVS NLS 3.2.0	HMOS280	Infoprint Serv-IP PrintWay base
HNET280	Infoprint Serv - NetSpool base	HOPI280	Infoprint Serv - Print Interface
HJS7703	JES3	JBD6202	Bulk Data Transfer SNA NJE
HRM7703	RMF base	HQX7703	SDFS base
HRF7703	SecureWay - RACF base	HFWL2A0	SecureWay - Firewall base
HRSL2A0	SecureWay - LDAP Serv base	HRSS190	SecureWay - DCE base
HR062A0	SecureWay - Open Crypto plugin	HCKVB00	C/C++ debug tool base
JCKVB03	C/C++ debug tool messages	HSM1G00	DFSORT base R14
JSM1G01	DFSORT/ISPF English	HGD3201	GDDM/MVS/PGF 2.1.3
JMQ414A	High Level Assem toolkit 1.4 VisualLift ADE 1.1.2	HO8K411	SOMobjects development env DCE user data privacy CDMF

HSWK2A0	SecureWay - Network Auth & Pri	HJVA11G	Java 1.01.08
HJVA11H	Java RAWT 1.01.08	HJVA11I	Java Swin 1.01.08
H249100	COBOL for MVS 1.02.00	J249101	COBOL for MVS 1.02.00
H080M10	VisualAge COBOL Millenium EXT	HOU1110	Millenium runtime 1.01.00
HOA5201	VA Java Compiler 2.00.00	HOA5202	VA Java runtime 2.00.00
HMWL131	AD/Cycle PL/I full funct 1.1.1	HWK0130	REXX/370 Compiler 1.03.00
JWK0131	REXX/370 Compiler English	HWJ9130	REXX/370 Library 1.03.00
JWJ9131	REXX/370 Library English	HWJ9133	rex/370 Alt Library English
H01B312	Net Commerce 3.01.01	HPZ8400	TME 10 NetView base 1.03.00
JPZ8404	TME 10 NetView US English	JPZ8407	TME 10 NetView USS English
JPZ8420	TME 10 NetView Procedural base	JPZ8424	TME 10 Netview Proc English
JPZ8440	TME 10 NetView Graphical base	JPZ8454	TME 10 NetView Graphical USS
H0GB310	DITTO/ESA 1.03.00	HMQ2100	MQSeries 2.01.00
HCI5300	CICS TS 1.3 base	JCI5301	COBOL Language parts
JCI5302	PL/I Language parts	JCI5303	C Language parts
JCI530D	IIOP/Java parts	JCI530E	IIOP/Java parts
HLR2140	CICSPlex SM 1.4 (MVS & OS/2)	JLR2142	CICSPlex SM 1.4 (SAS comp)
HBDD110	CICS Applic migration aids	HCC1102	CICS/DDM
HOB5110	CICS REXX runtime facility	HOB7110	CICS REXX development system
HOZ2110	CICS REXX common - CICS/ESA	HOAL200	Tivoli GEM Instrum for CICSPlex
HMK6600	IMS/ESA System Services V6	JMK6601	IMS/ESA Database Manager
JMK6602	IMS/ESA Transaction Manager	JMK6603	IMS/ESA ETO
JMK6604	IMS/ESA Recovery-level tracking	JMK6605	IMS/ESA Database-level tracking
HIR2101	IMS/ESA IRLM 2.1	HDB6610	DB2 base TSO 6.1.0
JDB6602	DB2 JDBC V6 tool	HOG9610	DB2 Performance monitor
JOG961E	DB2 Perf monitor English	JOG961X	DB2 Perf Monitor BUY
H0IH610	DB2 Admin Tool V6	J0IH611	DB2 Admin Tool BUY
H00G610	DB2 Buffer Pool Tool V6	J00G611	DB2 Buffer Pool Tool BUY
H242220	NET.DATA V6	HSQ6610	QMF V6
JSQ6611	QMF V6 BUY	HHP6601	QMF High Perf Option V6
JHP6602	QMF High Perf Opt Compiler	JHP6603	QMF High Perf Opt Manager
JHP6611	QMF High Perf Opt BUY		

We can roughly summarize this list by noting that this AD release includes all the standard components of OS/390 V2R10, many optional components (such as the Security Server (RACF), JES3, compilers, and so forth). It also includes the following software products that are not packaged as part of OS/390:

- ▶ Java for OS/390, release 1.01.08 (5655-A46)
- ▶ COBOL for MVS, release 1.02.00 (5655-A25)
- ▶ Visual Age Java for OS/390 compiler and runtime, release 2 (5655-JAV)
- ▶ SAA AD/Cycle PL/I compiler (full package), release 1.01.01 (5688-235)
- ▶ REXX/370 compiler and libraries, release 1.03.00 (5695-013 and 5695-014)
- ▶ Net.Commerce for OS/390, release 3.01.01 (5697-D32)
- ▶ Many TME 10 NetView elements, release 1.03.00 (5697-B82)
- ▶ DITTO/ESA, release 1.03.00 (5655-103)
- ▶ MQSeries for OS/390, release 2.01.00 (5655-A95)
- ▶ CICS TS, release 1.3 (5655-147) with many optional elements
- ▶ IMS/ESA, version 6 (5655-158), with optional elements
- ▶ DB2, version 6 (5645-DB2), with optional elements (including QMF) and several additional products related to DB2

This list indicates the order numbers, such as 5655-A46, that would be used to order the products separately. The order number is sometimes used in discussions about specific versions of these products.

2.2 Volumes distributed on CD-ROMs

The following table provides a brief overview of the volumes (and their associated CD-ROMs) included with the release 10 AD system.

CD	Volser	Type	Used	Emul	Prim	Content
1	OS39RA	3390-3	87%	A80	AA0	IPL, system libraries
2	OS3RAA	3390-3	50%	A81	AA1	More system libraries
3	OS39M1	3390-3	25%	A82	AA2	IODF, SYS1.IPLPARM, VSAM, etc
4	OS39HA	3390-3	64%	A87	AA7	System HFS data sets
5,3	OS39DA	3390-3	73%	A85	AA5	System DLIBs
6	OS3DAA	3390-3	95%	A86	AA6	More system DLIBs
7	OS3DAB	3390-2	89%	A88	AA8	More system DLIBs
8	OS39PA	3390-2	96%	A83	AA3	DB2, including DLIBs and HFS
9	OS3PAA	3390-2	52%	A84	AA4	TS (CICS), including DLIBs and HFS
10	OS3PAB	3390-2		A89	AA9	IMS, including DLIBs and HFS
11	OS3PAC	3390-3		A8A	AAA	WebSphere components
12	OS3PAD	3390-3		A8B	AAB	WebSphere components
13	(none)					FAQs, IBM Redbooks

A few notes are needed to explain this table:

- ▶ Volume OS39DA is partly on CD 5 and partly on CD 3. This involves one of the rare cases where a zipped volume did not fit on a single CD. This is a 3390-3 volume and, as such, is contained in two OS/2 files. The first file contains the first 2 GB of the volume and the second file contains the remaining .8 GB. The two files would normally be included in a single zip file (as is done for the other 3390-3 volumes), but the single zip file was larger than 650 MB. For this reason, the first file (2 GB) was zipped and placed on CD 5. There was sufficient free space on CD 3 to contain the second zipped file. When these two files are unzipped (from the two CDs), the file names (OS39DA_1.A85 and OS39DA_2.A85) will cause the emulated I/O device manager to use both files to emulate the full 3390-3 volume.
- ▶ Note that there is a mixture of 3390-2 volumes (1.9 GB) and 3390-3 volumes (2.8 GB). The smaller device model was used where possible.
- ▶ There is considerable free space on some of the volumes. If you are not concerned about migration issues (migration to the next AD release, that is), you could use some of this space for your own data sets. We generally recommend against doing this because it almost always leads to migration problems. (“How do I save all my data sets on the old volumes that I need to delete in order to have space to install the next release?”) Also, you should always preserve some free space on system volumes for PTFs and other urgent fixes that should reside on these volumes.
- ▶ The *Emul* and *Prim* columns above refer to the assumed addresses (device numbers) if the volumes are on emulated drives (always the case for P/390s, and an option for MP3000s), or on MP3000 primary drives. Again, the use of these addresses is not required.

- ▶ Only the first four volumes in this list are required to IPL the system. You need not install any of the other volumes unless you need the specific contents of the volume.
- ▶ If you need to access the system DLIBs (to install PTFs, for example), you will probably need to install all three DLIB volumes because there is no definition or documentation about which libraries are on which DLIB volume.

The last CD (number 13) does not contain a S/390 volume. It contains PDFs for several Redbooks, including:

- ▶ SG24-4721: *Accessing OpenEdition from the Internet*
- ▶ SG24-4747: *P/390, R/390, S/390 Integrated Server: OS/390 New User's Cookbook*
- ▶ SG24-4847: *P/390 and R/390 with OS/390: An Introduction*
- ▶ SG24-4894: *The Universal Connectivity Guide to DB2*
- ▶ SG24-5277: *Revealed! CICS Transaction Gateway with More CICS Clients Unmasked*
- ▶ SG24-5633: *Multiprise 3000: Technical Introduction*
- ▶ SG24-5669: *Multiprise 3000: Basic Emulated I/O Definitions*

The last CD also contains a number of files, in HTML format, containing frequently asked questions related to the AD systems. If your CD-ROM drive is H, you can access these files by pointing a browser to `file:///H:/faq/` and selecting the various files, one at a time. About half the files contain index titles and the other half contain the questions and answers.

2.2.1 Disk space required

The AD system no longer includes mini disks¹; all the volumes are in standard 3390 sizes. For release 10, the four volumes needed to IPL require 11.2 GB disk space (in the form of four 3390-3 volumes), or 12 SUs of MP3000 primary space. The system DLIBs require 7.5 GB (8 SUs). The DB2, CICS, and IMS volumes *each* require 1.9 GB (2 SUs each). The two WebSphere volumes require 5.6 GB (6 SUs).

The complete system requires 30 GB disk (or 32 SUs). We expect few users to install all volumes, but, nonetheless, this AD release is considerably larger than early AD releases. In general, each release has been larger than the previous one and this trend will probably continue.

2.3 Parmlibs

In the good old days, most system parameters were in SYS1.PARMLIB, and this was a magic, well-known name. OS/390 allows multiple PARMLIBs and does not require that the names contain the qualifier "PARMLIB", except that SYS1.PARMLIB is always included. OS/390 will concatenate whatever libraries you name (in the LOADxx member you selected via bytes 5 - 6 in your IPL parameter) as parameter libraries and will automatically add SYS1.PARMLIB to the end of the concatenation (if you do not specify it elsewhere in the concatenation). The data set names of the system parameter libraries and the order of concatenation of the libraries are critical information for anyone maintaining or customizing OS/390.

The release 10 AD system contains two system parameter libraries:

```
SYS1.ADCD10.PARMLIB
SYS1.PARMLIB
```

¹ P/390-based systems can define 3380 and 3390 drives with any number of cylinders, to a maximum of 3339 cylinders (for 3390s). Early versions of the AD system used odd-sized volumes to conserve disk space on the P/390 machines. For example, if the contents of a DLIB volume used 1600 cylinders, then the distributed volume would be, say, 1700 cylinders. (The additional cylinders allow for reasonable PTF activity and customization.) The primary disks of the MP3000 do not provide odd-sized volumes. (Emulated disks on the MP3000 do support odd-sized volumes.) With many PID members moving to the MP3000, the group building the AD systems elected to distribute only standard-sized volumes. Doing this also saves time when building a new AD release.

and they are concatenated in the order shown. All of the members customized for the AD system are in the first library. However, default members (used by the AD system but unchanged from the original ServerPac) may be in the second library. You should always check to see if a member is in the first library; if not then look in the second library.²

You can display the current parameter library concatenation list with the MVS operator command **D PARMLIB**.

In the remainder of this redbook, we will use the term PARMLIB to mean the concatenated list of system parameter libraries, without specifying exactly which library contains the member being discussed.

2.4 IPL parameters

The release 10 AD system greatly expanded the number of IPL parameters distributed with the system. In the follow discussion, we assume that the IPL address (for the OS39RA volume) is A80 and that the IPLPARAM points to address A82 (for the OS39M1 volume). Your first IPL would use the parameters:

```
IPL address = A80      IPL Parameter = 0A82CS
```

The following discussion is about the fifth and sixth characters in the IPL Parameter, shown as CS in this example. These two characters are used to access a member in SYS1.IPLPARAM. The two characters are used as a suffix for the word "LOAD". In this example, the OS/390 load process would reference SYS1.IPLPARAM(LOADCS). The LOADCS member has values that control other startup parameters. The general process is described in more detail in "Understanding IPL parameters" on page 103. In this section we will describe the particular parameters (the two characters) that are provided with the release 10 AD system.

The goal of the AD system is to permit you to *cold start* or *warm start* your system in a number of ways. As used here, *cold start* means to perform a CLPA and then to clear the JES2 spool contents.³ *Warm start* means to use the existing contents of the PLPA (established by an earlier cold start), and to retain the existing contents of the JES2 spool.

Several subsystems need to place modules in PLPA. For example, if you want to use DB2, then you need to include several DB2 modules in PLPA. If you do not use DB2, then you probably want to omit these modules from your PLPA, to save space in common storage.

The various load parameters provided with release 10 allow you to use combinations of no DB/OLTP or DB2 or CICS + DB2 or IMS and so forth. For each combination, there is a cold start and a warm start option. You must always IPL with the cold start option; later IPLs can then be warm starts. If you then want a different mixture of subsystems, you must cold start with the appropriate load parameter. Thereafter you can warm start with the matching parameter.

The supplied parameters are:

Cold Start	Warm Start	Purpose.....
CS	00	Basic system. No CICS, IMS, DB2, MQ
DC	DB	Basic system + DB2, CICS (and RRS)
CC	CW	Basic system + CICS (and RRS)

² You could use the ISRDDN function of ISPF instead. This will handle multiple concatenated libraries, in the correct order of concatenation.

³ The various startup combinations also select different SVC lists, different lists of subsystem properties, and so forth, in PARMLIB. The different PLPA libraries are the most obvious differences between the various IPL parameters and we usually refer to them in discussions, but this should not be taken to exclude the other differences in the various IPL parameter controls.

IC	IW	Basic system + IMS (and RRS)
2C	2W	Basic system + DB2 (and RRS)
	DI	Basic system + IMS (and RRS). CLPA + JES2 warm start
AC	AW	Basic system + DB2, CICS, IMS, MQ (and RRS)

Again, remember that you must CLPA (cold start) a given DB2/IMS/CICS/MQ environment before you can later warm start that environment. A CLPA destroys the previous contents of PLPA. (The DI parameter, in this list, is a mixture of CLPA and a JES2 warm start. We are not quite clear how it came to be included in the release 10 AD system, but it illustrates the principle that you can customize any startup arrangement you want.)

The AD designers assume that, if you IPL with a parameter that includes, say, DB2 and CICS modules, then you want to automatically start these subsystems. This is done by linking different VTAMAPPL command lists indirectly to the IPL parameter. There are other ways to manage your system. The author, for example, prefers to have his system capable of running CICS and DB2, but does not want to start these subsystems automatically. The DC and DB parameters load the correct modules, but then automatically start DB2 and CICS. You can change this by making a modified set of IPL parameters or by simply removing the start commands from the relevant VTAMAPPL command lists.

2.4.1 Detailed IPL parameter controls

The LOADxx member selected by the IPL parameter usually indicates which IEASYSxx member (in PARMLIB) to use. The IEASYSxx member then indicates a variety of other PARMLIB members to use. The following table attempts to summarize key elements of various IPL parameters.

LOADxx	CS	00	DC	DB	CC	CW	2C	2W	IC	IW	DI	AC	AW
IEASYSxx	CS CLPA	00	DC CLPA	DB	CC CLPA	CW	2C CLPA	2W	IC CLPA	DI	DI CLPA	AC CLPA	AW
IEASVCxx	00	00	CI CICS	DB	CI CICS	CI CICS	00	00	IM IMS	IM IMS	IM IMS	AL IMS CICS	AL IMS CICS
LPALSTxx	00	00	DB CICS	DB CICS	DB CICS	DB CICS	D2	D2	IM	IM	IM	DB CICS	DB CICS
IEALPAXx	00	00	00	00	00	00	00	00	DI	DI	DI	DI	DI
BPXPRMxx	CS	CS	00 CICS DB2 WEB WAS	00 CICS DB2 WEB WAS	CI CICS WEB WAS	CI CICS WEB WAS	DB DB2 WEB	DB DB2 WEB	00 CICS DB2 WEB WAS	00 CICS DB2 WEB WAS	00 CICS DB2 WEB WAS	00 CICS DB2 WEB WAS	00 CICS DB2 WEB WAS
SCHEDEXx	00	00	DB	DB	DB	DB	DB	DB	DB	DB	DB	DB	DB
PROGxx	00	00	DB DB2 CICS	DB DB2 CICS	CI CICS	CI CICS	D2 DB2	D2 DB2	IM IMS	IM IMS	IM IMS	AL DB2 IMS CICS	AL DB2 IMS CICS

LOADxx	CS	00	DC	DB	CC	CW	2C	2W	IC	IW	DI	AC	AW
COMMNDxx JES2 started S VTAMxxxx	CS COLD APPL	00 WAR M APPL	DC COLD DB	DB WAR M DB	CC COLD CI	CW WAR M CI	2C COLD D2	2W WAR M D2	IC COLD APPL RRS	IW WAR M APPL RRS	DI WAR M APPL RRS	AC COLD ALL	AW WAR M ALL
VTAMxxxx includes			RRS DB2 CICS	RRS DB2 CICS	RRS CICS	RRS CICS	RRS DB2	RRS DB2				RRS DB2 CICS IMS MQ	RRS DB2 CICS IMS MQ

Consider the column with the heading CC. The controls in this column are invoked when your IPL parameter is 0A82CC. It starts by using the LOADCC member of SYS1.IPLPARM. This member invokes IEASYSCC in PARMLIB. The IEASYSCC member specifies CLPA and points to the other members in this list. It points to IEASVCCI; of interest in this list is the inclusion of the CICS SVC. LPALST DB is used, and it contains a CICS library. IEALPA00 is used. BPXPRMCI is used, and it mounts file systems for CICS, the Web server, and WebSphere (WAS). The SCHEDDB member is used. PROGCI is used, and it includes the CICS libraries. COMMNDCC is used; it cold starts JES2 and invokes VTAMCI. The VTAMCI command list includes commands to start RRS and CICS.

This table is not complete—a complete table would be much too large to list—but it quickly identifies the more important differences among the supplied IPL parameters.

Read horizontally, it indicates, for example, that PARMLIB members BPXPRMCS, BPXPRM00, BPXPRMCI, and BPXPRMDB are used. Other BPXPRMxx members in PARMLIB are not used and could be deleted or customized and then used.

You can add your own LOADxx members and use or expand the PARMLIB members in any way that suits your needs.

Most of the examples in this document were produced using a basic system; that is, one without DB2, CICS, IMS, MQ, or WebSphere. For the basic system, we IPLed (once) with parameter 0A82CS and thereafter IPLed with 0A8200. (For the examples run on our MP3000, these parameters were 0AA2CS and 0AA200.)

2.5 User IDs

The release 10 AD system is provided with many userids. We assume not all of them are intended for use.

USERID	Def	GRP	ATTRIBUTES	INTERVAL	TSO	OMVS	Notes:
IRRCERTA			REVOKED	N/A	no	no	Certificate anchor
IRRMULTI			REVOKED	N/A	no	no	Criteria anchor
IRRSITEC			REVOKED	N/A	no	no	Site Anchor
APPC	SYS1			180	no	no	
ASCH	SYS1		REVOKED	180	no	no	
BLSJPRMI	SYS1		REVOKED	180	no	no	
BPXOINIT	SYS1			180	no	UID=0	
CICSA	SYS1		REVOKED	180	no	no	
CICSTART	SYS1		REVOKED	180	no	no	
CSQ1CHIN	SYS1		OPERATIONS	180	no	UID=0	
CSQ1MSTR	SYS1		OPERATIONS	180	no	no	

DB2iRLM	SYS1	REVOKED	180	no	no	
DCEADM	SYS1	REVOKED	180	no	UID=0	
DCEKERN	DCEGRP	REVOKED	180	no	UID=0	
DFS	DFSGRP	REVOKED	180	no	UID=0	
DFSCM	DFSGRP	REVOKED	180	no	UID=0	
DSNIDBM1	SYS1		180	no	no	
DSNIDIST	SYS1		180	no	UID=0	
DSNISPAS	SYS1	GRPACC	180	no	no	JCL,RECOVER
DUMPSRV	SYS1	REVOKED	180	no	no	
EZAFTPAP	OMVSGRP		180	no	UID=0	
FTPD	SYS1		180	no	UID=0	
GTF	SYS1	REVOKED	180	no	no	
IBMUSER	SYS1	SPEC, OPER	N/A	yes	UID=0	ACCT,JCL,MOUNT,OPER,RECOVER
IMS	SYS1		180	yes	no	
IMSA	SYS1		180	no	no	
IMS61CRI	SYS1		180	no	no	
IMS61DL1	SYS1		180	yes	no	
IMS61F11	SYS1		180	yes	no	
IMS61F12	SYS1		180	yes	no	
IMS61F13	SYS1		180	yes	no	
IMS61M11	SYS1		180	yes	no	
IMS61RC1	SYS1		180	yes	no	
IMS61RD1	SYS1		180	yes	no	
IMWEBSRV	SYS1		180	no	UID=0	
INETID	SYS1		180	no	no	
INIT	SYS1	REVOKED	180	no	no	
INTERNALEMPLOYEE			180	no	UID=537	
IRRDPTAB	SYS1	REVOKED	180	no	no	
IVPGNJCL	SYS1		180	yes	no	
JESB	SYS1	REVOKED	180	no	no	
JES2	SYS1	REVOKED	180	no	no	
LLA	SYS1	REVOKED	180	no	no	
NFSC	SYS1		180	no	UID=0	
NFSS	SYS1		180	no	UID=0	
OMPROUTE	OMVSGRP		180	no	UID=0	
OMVS	SYS1	REVOKED	180	no	UID=0	
OMVSKERN	OMVSGRP	REVOKED	180	no	UID=0	
OPEN1	SYS1		180	yes	UID=0	RECOVER,JCL
OPEN2	SYS1		180	yes	UID=0	RECOVER,JCL
OPEN3	SYS1		180	yes	UID=0	RECOVER,JCL
OSNMPD	OMVSGRP		180	no	UID=0	
PORTMAP	SYS1		180	yes	UID=0	
PRIVATE	SPECIAL		180	no	UID=416	
PUBLIC	EXTERNAL		180	no	UID=998	
P390	SYS1	SPEC, OPER		yes	UID=0	ACCT,OPER,JCL
P390A	TEST	SPEC, OP, GRP N/Z		yes	no	OPER,JCL,RECOVER
P390B	TEST		30	yes	no	JCL,RECOVER
....						
P390Z	TEST		30	yes	no	
RACF	SYS1	REVOKED	180	no	no	
RMF	SYS1	REVOKED	180	no	no	
ROUTEDMV	OMVSGRP	REVOKED	180	no	UID=447	
ROUTEDOE	OMVSGRP	REVOKED	180	no	UID=448	
SNMPQE	SYS1		180	no	UID=0	
SYSADM	SYS1		180	yes	no	JCL,RECOVER
SYSLOGD	OMVSGRP		180	no	UID=0	
SYSOP	SYS1		180	yes	no	
SYSOPR	SYS1	SPEC, OPER	180	yes	no	JCL,RECOVER
TCAS	SYS1	REVOKED	180	no	no	
TCPIP	SYS1		180	yes	UID=0	

TCPIPMVS	OMVSGRP	REVOKED	180	no	UID=445	
TCPIPOE	OMVSGRP	REVOKED	180	no	UID=0	
TESTER	TEST		30	yes	no	RECOVER
VLFF	SYS1	REVOKED	180	no	no	
VTAM	SYS1	REVOKED	180	no	UID=0	
WEBADM	IMWEB		180	no	UID=206	
WEBSRV	IMWEB		180	no	UID=0	

There are several interesting points in this list:

- ▶ The column labeled TSO indicates whether the userid is authorized for TSO; that is, does it have a TSO segment in RACF. The column labeled OMVS indicates whether the userid is authorized to use UNIX System Services; if so, the assigned UID is shown.
- ▶ The Notes column indicates additional TSO authorities the userids may have, such as OPER and so forth.
- ▶ A group of invalid userids (lacking RACF user definitions) are defined for many of the TSO authorities. We assume these are left over from the group that built the AD release.
- ▶ The system was built for convenience, not security. Almost all the userids have their default passwords, which are equal to their default group names. For example, you can log onto SYSOPR (which has RACF SPECIAL authority) with password SYS1. Or, you can log in by telnet to Unix System Services (TCP/IP port 1023, once you have OS/390 TCP/IP operational) with userid TCPIP, password SYS1. To better secure the system, you would need to change the passwords of all userids that are not REVOKED.⁴
- ▶ Many userids are marked REVOKED. This prevents any direct usage of these userids. In general, these REVOKED userids are used as started task names and are valid for that purpose.⁵
- ▶ Note the considerable number of userids with UID=0. These mostly represent system components, rather than end-user userids.
- ▶ The specific non-zero UIDs assigned have no numeric significance; you must avoid reusing the same UIDs for your users.
- ▶ Not all the userids are necessary; some are relics of earlier AD systems. Do not assume there is a strong reason for all the userids present, or for the distribution of various authorities.

Several TSO logon procedures are authorized for various users. As distributed, these are:

Procedure	UACC	Users.....
DBSPROC	NONE	P390,DSN1SPAS,SYSADM,SYSOPR,IBMUSER,P390A
IKJACCNT	NONE	IBMUSER,P390,P390A
ISPFLITE	READ	IBMUSER(UPDATE),P390(UPDATE)
ISPFPRCB	NONE	IBMUSER,P390
ISPFPROC	NONE	IBMUSER,P390,TESTER,P390A-Z
OMVSPROC	NONE	IBMUSER,IOEN1-3,P390

In addition, there are several invalid userids permitted to most of these procedures.

For completeness, we list the significant definitions in the RACF FACILITY class. The R, U, and A indicators are for access levels READ, UPDATE, and ALTER:

Profile	UACC	Permitted users (or owner).....
---------	------	---------------------------------

⁴ Userids that do not have a TSO segment or an OMVS segment (and corresponding UID) provide less exposure, but still might be used for running batch jobs.

⁵ A better approach would be to use the RACF *STARTED* class for this purpose, but for continuity reasons the AD developers have not done this (yet).

BPX.DAEMON	NONE	IBMUSER(A),OMVSKERN(R),WEBSRV(R),FTPD(R),TEST1-6(R)
BPX.FILEATTR.APF	NONE	IBMUSER(A),P390(R)
BPX.FILEATTR.PROGCTL	NONE	IBMUSER(R),P390(R),TEST2(R)
BPX.SERVER	NONE	P390(A),WEBSRV(U),IMWEBSRV(U)
BPX.SUPERUSER	NONE	IBMUSER(A)
DCE-----		(several profiles for DCE-related controls)
IRRDPIO0	READ	P390(A)
STGADMIN.ADR.*	NONE	P390(A),SYS1(A)
STGADMIN.IDC.*	NONE	P390(A),SYS1(A)
STGADMIN.IGD.*	NONE	P390(A),SYS1(A)
STGADMIN.IGG.*	NONE	P390(A),SYS1(A)
STGADMIN.*	NONE	P390(A),SYS1(A)

2.6 IODFs and addresses

Three IODFs are included with the release 10 AD system. These are SYS1.IODF00, SYS1.IODF01, and SYS1.IODF99. We refer to these as the 00, 01, and 99 IODFs. Your IPL parameter, via a value in the LOADxx member, selects which IODF is used for a given IPL. No “work” IODFs are included.⁶

As far as we could determine, the only differences between 00 and 01 were differences in esoteric device definitions that added the 3390s starting at address A80. Based on this, SYS1.IODF00 can be considered obsolete and you can ignore it.

2.6.1 SYS1.IODF01

All of the LOADxx members are set to use SYS1.IODF01. This IODF defines the following devices and addresses:

00C	2540R	Card reader. Useful as an emulated device
00E	1403-N1	Line printer. Useful as an emulated device
00F	3203	Line printer. Not clear why this is included.
010	3277	Terminal. Relic of system used to create AD releases
063	3277	Terminal. Relic of system used to create AD releases
120-15F	3380	Disks
240-25F	3380	Disks
260-27F	3390	Disks
560-57F	3480	No COMPACT option is defined on any drive
580-59F	3420-8	“Round” tapes. For AWSTAPE emulated drives.
5A0-5AF	3422	“Round” tapes. For AWSTAPE or OMA emulation
700	3270	Terminal. AD systems use as NIP & OS/390 master
701-73F	3270-X	Terminal. Normally for VTAM (TSO, CICS, etc)
900-91F	3270-X	Terminal. Normally for VTAM (TSO, CICS, etc)
A80-ABF	3390	Disks. Can be for system volumes, etc
E20-E27	CTC	Typically for 3172s that appear as CTC devices
E40-E47	CTC	Typically for 3172s that appear as CTC devices

As far as we know, all the devices in this IODF are defined correctly. The most common change needed to is add the COMPACT feature to one (or more) 3480 definitions. The COMPACT feature is needed to process a compressed tape. For 3480 drives, this is known as an IDRC hardware feature and is enabled by selecting the COMPACT feature in the IODF definition. The COMPACT function is standard for 3490 drives.

⁶ There are two types of IODFs, production and work. The three included with the release 10 AD system are all production IODFs.

You will probably use this IODF for your initial system operation. Many AD users never need to change this IODF and use it for all their system operations. See “Changing your I/O configuration” on page 18 for comments about matching your hardware to this IODF. However, you are not required to use the IODFs that are distributed with the AD system. If you have the necessary skills, you can create any IODF you require by using the HCD utility.

2.6.2 SYS1.IODF99

SYS1.IODF99 is also provided with the AD release. This is intended to be an expanded version of IODF01, with more devices that might be needed for an MP3000 machine. Unfortunately, IODF99 contains a number of errors. Moreover, there is no particular need to use it with an MP3000 unless you want device addresses that match particular documentation you may have.⁷ None of the supplied LOADxx members point to IODF99. Member LOAD99, included in SYS1.IPLPARM but not documented in “Detailed IPL parameter controls” on page 29, is intended to point to IODF99 but points to IODF01 by mistake.

One way to use IODF99 is as follows:

- ▶ Correct LOAD99 to point to IODF99.
- ▶ IPL with parameter 0A8299T (using the correct address for the OS39M1 volume)
- ▶ The T parameter (seventh character in the IPL parameter field) causes the system to issue the SPECIFY SYSTEM PARAMETERS message. The reply should be:

```
REPLY 00,SYSP=CS
```

or any of the other two-character parameters that are possible suffixes for the IEASYSxx name, as listed in 2.4.1, “Detailed IPL parameter controls” on page 29.

This technique provides the same effect obtained from IPLPARM 0A82CS (in the sample here), but with a different starting point (LOAD99 instead of LOADCS). Another way to use IODF99 would be to change all the LOADxx members to refer to 99 instead of 01, or to create a whole new set of LOADxx members containing 99.

Again, we repeat that all the examples in this book were with a mixture of P/390 and MP3000 systems, all using IODF01. *There is no requirement to use IODF99 with MP3000 systems.*

The IODF99 definitions are quite similar to the IODF01 definitions. IODF99 has the following:

00C	2540R	Card reader. Useful as an emulated device.
00E-00F	1403-N1	Line printer. Useful as an emulated device.
010	3277	Terminal. Relic of system used to create AD releases.
063	3277	Terminal. Relic of system used to create AD releases.
120-15F	3380	Disks
1C0-1FF	3390	Disks. Not defined correctly (no OS defined in HCD).
240-25F	3380	Disks. No OS defined in HCD.
260-27F	3390	Disks. No OS defined in HCD.
300-3BF	3390	Disks.
550-55F	3420-8	“Round” tape. No OS defined.
560-56F	3480	No COMPACT option is defined on any drive.
570-57F	3480X	Has COMPACT feature
580-58F	3490	Tape. No OS defined.
590	3490	Tape.
591-59F	3490E	Tape. Invalid definition; no OS defined.
700	3270	Terminal. AD systems use as NIP & OS/390 master
701-73F	3270-X	Terminal. Normally for VTAM (TSO, CICS, etc)

⁷ This is most common with the DaVinci package. Associated documentation assumes you are using addresses in the 300 range for your 3390 drives.

900-91F	3270-X	Terminal. Normally for VTAM (TSO, CICS, etc)
A80-ABF	3390	Disks. Can be for system volumes, etc
E20-E23	CTC	Typically for 3172s that appear as CTC devices
E40-E43	CTC	Typically for 3172s that appear as CTC devices

2.7 Catalogs

The release AD system has six catalogs:

Catalog Name	Volume	Comments.....
CATALOG.OS390.MASTER	OS39M1	Master Catalog, named in LOADxx members
USERCAT.VOS39RA	OS39RA	Large number of aliases point to this one
USERCAT.VOS39PA	OS39PA	DB2, QMF, etc
USERCAT.VOS3PAA	OS3PAA	CICS
USERCAT.VOS3PAB	OS3PAB	IMS
USERCAT.VOS3PAC	OS3PAC	WebSphere, etc

The master catalog contains aliases pointing to the various user catalogs. The aliases associated with each user catalog are:

```

USERCAT.VOS39RA
  AOP, ASU, CMX, ECN, EDF, EPH, EQAW, EUV, HOU, ICQ, IMW, IOA, JAVA, OS39R, ASM, BFS,
  CDS, CSF, DIT, EOY, EWX, CMN312, DTW220, MVS, CEE150, HLA, CBG, GDDM, GIM, GLD, GSK,
  ICA, IGY, IMO, IOE, ISP, SOMMVS, TCP/IP, COK, CEE, ISF, ASUITE, COT121, COG210,
  DIT130, HOU110, HJP200, IEL111, IGY210, CSQ, CSQ1, NFS, NETV130, FAN130
USERCAT.VOS39PA
  ADB610, DG0610, BFT610, DSN610, QMF610
USERCAT.VOS3PAA
  CICSTS13
USERCAT.VOS3PAB
  IMS
USERCAT.OS3PAC
  AJV118, BB0302, EJS302

```

Any data set names whose High Level Qualifier (HLQ) does not match one of these aliases are cataloged in the master catalog. This includes all the data sets created by (or for) userids such as IBMUSER, P390, P390A, and so forth.

As distributed, the master catalog has no RACF protection. A RACF profile exists for USERCAT.* with Universal Access (UACC) set to *READ*. Userids DSN1DBM1 and DSN1MSTR have UPDATE access to this profile. (IBMUSER, P390, and any other userid with RACF SPECIAL and OPERATIONS authority can also alter the user catalogs.) This is opposite to the usual situations in which users have READ access to the master catalog and UPDATE access to at least one user catalog.

2.8 Paging and spooling data sets

The paging and spool data sets for the distributed AD system are all on the OS39M1 volume. This is convenient for packaging and system management. However, it may not be appropriate for performance and growth. Once you have your AD system installed and operational, you can move and/or expand these data sets.

The paging data sets in the distributed system are:

Full Data Set Name	Volume	Space
SYS1.PLPA.PAGE.DATA	OS39M1	1500 tracks

SYS1.COMMON.PAGE.DATA	OS39M1	750 tracks
SYS1.LOCAL.PAGE.DATA	OS39M1	6000 tracks

The PLPA page data set is about 82% full (after IPLing with the CS parameter). The 6000 tracks of user paging space amounts to about 288 MB of virtual memory that requires disk backing. This is sufficient for reasonable numbers of TSO users and batch jobs. It may not be sufficient for combinations of WebSphere, Java, CICS, DB2, and so forth. If additional paging data sets are needed, they should be allocated on user volumes for better performance.

The allocated JES2 spool space is 2340 tracks (156 cylinders). This represents over a million lines of spooled output. It is certainly sufficient for getting started, but most traditional MVS installations can consume any amount of spool space. MP3000 installations may typically allocate a complete 3390-3 volume for spool space.

As a side note, the allocated space for SMF data sets is:

Full Data Set Name	Volume	Space
SYS1.MAN1.DATA	OS39M1	600 tracks
SYS1.MAN2.DATA	OS39M1	150 tracks
SYS1.MAN3.DATA	OS39M1	150 tracks

By default, SMF recording is turned off, so none of this space is needed. The sizes allocated are reasonable for typical SMF recording. If you enable SMF recording, be certain to have jobs ready to empty the SMF data sets when needed.

2.9 UNIX System Services file structure

The UNIX System Services data sets and mount points are organized differently than in earlier AD releases. The general intention is to provide a new level of indirection in file system mounts. This should make it easier to apply maintenance, install and test new releases, and so forth. These changes were introduced in the base ServerPac, and are not unique to the AD system.

The mount points specified in BPXPRMxx use system symbols:

```
&SYSNAME = P390
&SYSPLEX = ADCDPL
&UNIXVER = V2R10M0
```

You can display the current system symbols with the MVS command **D SYMBOLS**. In addition to these, another symbol is defined in BPXPRMxx:

```
VERSION('&UNIXVER.') = V2R10M0
```

Using these symbols, the file system mounts defined in BPXPRMxx include:⁸

Full Data Set Name	Volume	Mount point	Actual mount point	Space
HFS.ADCDPL.ROOT	OS39HA	-----	/	.73 MB
HFS.P390.DEV	OS39HA	/&SYSNAME./dev	/P390/dev	4.4 MB
HFS.P390.ETC	OS39HA	/&SYSNAME./etc	/P390/etc	15.6 MB
HFS.P390.HFS	OS39HA	/&SYSNAME	/P390	.73 MB
HFS.P390.TMP	OS39HA	/&SYSNAME./tmp	/P390/tmp	30.8 MB
HFS.P390.VAR	OS39HA	/&SYSNAME./var	/P390/var	3.77 MB
HFS.USERS	OS39HA	-----	/u	.1 MB
HFS.USR.MAIL	OS39HA	-----	/usr/mail	6.67 MB
HFS.V2R10M0.VERSION.HFS	OS39HA	/\$VERSION	/V2R10M0	801.1 MB

⁸ There are additional mounts not mentioned here, for NetView, CICS, and so forth. These are straightforward and their inclusion here would only complicate the discussion.

HFS.WAS	OS39HA	-----	/was	101.0	MB
HFS.WEB	OS39HA	-----	/web	22.0	MB

Note the size of /u. It is intended only for a mount point for your /u directories--which should be on your volumes, not the system volumes.⁹ While not shown here, you should be aware that specific product volumes, such as OS39PA (for DB2), contain additional HFS data sets (and each is a file system) for that product. Notice that some traditional "first level" directories, such as /etc, are now "second level" directories, such as /P390/etc.

The root directory contains links that make /P390/etc, for example, look like the more traditional /etc. Output from the command `ls -al` (in the root directory) includes:

```

drwxr-xr-x  6 BPXOINIT SYS1  8192 Dec  1 14:01 P390
drwxr-xr-x 21 BPXOINIT SYS1  8192 Oct 20 17:01 V2R10M0
lrwxrwxrwx  1 BPXOINIT SYS1    12 Dec  1 17:01 bin -> $VERSION/bin
lrwxrwxrwx  1 BPXOINIT SYS1    12 Dec  1 17:01 dev -> $SYSNAME/dev
lrwxrwxrwx  1 BPXOINIT SYS1    12 Dec  1 17:01 etc -> $SYSNAME/etc
lrwxrwxrwx  1 BPXOINIT SYS1    12 Dec  1 17:01 lib -> $VERSION/lib
lrwxrwxrwx  1 BPXOINIT SYS1    12 Dec  1 17:01 opt -> $VERSION/opt
lrwxrwxrwx  1 BPXOINIT SYS1    16 Dec  1 17:01 samples -> $VERSION/samples
lrwxrwxrwx  1 BPXOINIT SYS1    12 Dec  1 17:01 usr -> $VERSION/usr
lrwxrwxrwx  1 BPXOINIT SYS1    12 Dec  1 17:01 var -> $SYSNAME/var
drwxr-xr-x  8 BPXOINIT SYS1  8192 Dec  1 15:02 u
drwxr-xr-x  1 BPXOINIT SYS1  8192 Dec  2 17:05 web
drwxr-xr-x  1 BPXOINIT SYS1  8192 Dec  2 17:01 was

```

Notice that all the traditional names, such as /bin, resolve to directories in either P390 or V2R10M0. In principle, the contents of P390 (and all its subdirectories) are locally customized and perhaps release independent, while V1R10M0 contains code and other files that are release dependent.

When first seen, this arrangement appears complicated. For practical purposes, it is seen only when listing the root directory. Routine functions and commands, such as `cd /etc`, work as expected--thanks to the links in the root directory. The long-term purpose of this arrangement is to make software maintenance and migration easier. In the meantime, it requires second glances when directly working in the root directory.

2.10 SMP/E data sets

The AD system contains a full SMP/E environment for all the components shipped with it. For the base system components, the SMP/E data sets are on the DLIB volumes. For DB2, CICS, IMS, and various WebSphere components, the SMP/E data sets are on their respective OS39PA, OS3PAA, OS3PAB, OS3PAC, or OS3PAD volume. In this section we will ignore these SMP/E data sets and discuss only those contained on the DLIB volumes.

The easiest way to start investigating the SMP/E structure is to use the ISPF 3.4 command to list the names of data sets with the same **** .CSI**; that is, with any data set name ending in a qualifier of **CSI**. All SMP/E consolidated inventory data sets have this type of name. It is possible that a non-SMP/E data set may have such a name, but none are present in the distributed R10 AD system.

The CSIs present on the DLIB volumes are:

Data set name	Volume	Product or Comment
MVS.GLOBAL.CSI	OS3DAA	Primary starting point for most SMP/E work
MVS.MVSD.CSI	OS3DAA	CSI for Base system DLIBs

⁹ Alternately, you could define your own /u file system (not on a system volume) and replace (in BPXPRMxx) the **MOUNT** command for /u.

MVS.MVST.CSI	OS3DAA	CSI for operational libraries
CEE150.GLOBAL.CSI	OS3DAA	Language Environment
CEE150.CEE150D.CSI	OS3DAA	
CEE150.CEE150T.CSI	OS3DAA	
CMN312.GLOBAL.CSI	---	On both OS39DA and OS39PA (?)
CMN312.CMN312D.CSI		Net.Commerce
CMN312.CMN312T.CSI	OS39DA	(Target CSI on OS39DA only)
CSQ.CSQD.CSI	OS39DA	MQSeries
CSQ.CSQT.CSI	OS39DA	Uses the MVS Global CSI
DIT130.GLOBAL.CSI	OS3DAA	Ditto
DIT130.DIT130D.CSI	OS3DAA	
DIT130.DIT130T.CSI	OS3DAA	
FAN130.GLOBAL.CSI	OS3DAA	REXX Compiler
FAN130.FAN130D.CSI	OS3DAA	
FAN130.FAN130T.CSI	OS3DAA	
HOU110.GLOBAL.CSI	OS3DAA	Millenium Extensions
HOU110.HOU110D.CSI	OS3DAA	
HOU110.HOU110T.CSI	OS3DAA	
HPJ200.GLOBAL.CSI	OS39DA	Visual Age for Java
HPJ200.HPJ200D.CSI	OS39DA	
HPJ200.HPJ200T.CSI	OS39DA	
IEL111.GLOBAL.CSI	OS3DAA	PL/I Compiler and libraries
IEL111.IEL111D.CSI	OS3DAA	
IEL111.IEL111T.CSI	OS3DAA	
IGY210.GLOBAL.CSI	OS3DAA	COBOL Compiler and libraries
IGY210.IGY210D.CSI	OS3DAA	
IGY210.IGY210T.CSI	OS3DAA	

When using the ISPF panels for SMP/E, the first panel (SMP/E Primary Options Menu) should usually be set to one of the global CSI data sets, such as:

```
SMPCSI DATA SET ==> 'MVS.GLOBAL.CSI'
```

The various SMP/E holding data sets are named in an orderly way and can be found using the ISPF 3.4 function with **** .SMP*** as the Dsname Level. Note that some of the SMPPTS and SMPLTS data sets are quite large.

2.11 DFSMS state

HFS volumes are no longer required to be System Managed volumes. None of the volumes in the release 10 AD system are managed by DFSMS. Most users of the AD system will probably ignore DFSMS.

Two storage groups (DBCLASS and HFSCCLASS) are defined, and two storage classes are defined (same names). Two volsers are known to DFSMS: OS3P8A (group DBCLASS) and OS3H8A (group HFSCCLASS). These volsers are clearly left over from an earlier AD release.

The use of HFS volumes requires that DFSMS be active, even though HFS volumes no longer need to be defined as System Managed volumes. It is active using the standard data sets:

```
SCDS = SYS1.SCDS
ACDS = SYS1.ACDS
COMMDS = SYS1.COMMDS
SYSTEM NAME = P390
```

2.12 Minor fixes

As with any complex system, implemented under time pressures and intended for use under widely varying conditions, there are a number of minor adjustments that may be necessary in specific situations. Several widely-applicable adjustments are described in this section.

2.12.1 Timer settings

The AD system expects to find an ETR timer; this is typically associated with a Parallel Sysplex environment. If you do not have such an environment, you must reply to an IPL message to bypass this condition. To fix the problem, edit PARMLIB member CLOCK00 as follows:

```
OPERATOR NOPROMPT
TIMEZONE W.05.00.00      (or as appropriate if you do not use EST in the U.S.)
ETRMODE NO              (change from YES)
ETRZONE NO              (change from YES)
ETRDELTA 10             (not relevant)
```

2.12.2 LOGREC full

As distributed, SYS1.LOGREC is full. Run this job to clear LOGREC:

```
//OGDENE JOB 1,OGDEN,MSGCLASS=X
//EREP EXEC PGM=IFCEREPI,PARM='CARD'
//SERLOG DD DISP=SHR,DSN=SYS1.LOGREC
//DIRECTWK DD UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//EREPPT DD SYSOUT=*,DCB=BLKSIZE=133
//ACCDEV DD DUMMY
//TOURIST DD SYSOUT=*,DCB=BLKSIZE=133
//SYSIN DD *
        SYSUM
        ACC=Y
        ZERO=Y
/*
```

You can key this job into any convenient JCL library, such as SYS1.P390.CNTL, using any convenient member name, such as EREPZERO.

2.12.3 Remove automatic ASCH and APPC startups

By default, the AD system starts a number of APPC and ASCH address spaces. Very few AD system users need these. You should know if you need these; otherwise it is probably safe to assume that you do not need them. They are started by the VTAMAPPL command list in PARMLIB. In the release 10 AD system, there are five different VTAMAPPL command lists that might be used, depending on the IPL parameter you selected. You can study “Detailed IPL parameter controls” on page 29 to determine which VTAMAPPL command list members you need to change, or you can simply change all five. The five members (in SYS1.ADCD10.PARMLIB) are VTAMALL, VTAMAPPL, VTAMCI, VTAMDB, and VTAMD2. The same change is made to each member. The change *comments out* two lines, as follows:

```
/* S APPC,SUB=MSTR,APPC=1A */
/* S ASCH,SUB=MSTR,APPC=1A */
```

Your change is to place the /* and */ around these two lines in the PARMLIB members. Alternatively, you could simply delete the lines; however, normal MVS practice is to leave the *commented out* lines as a form of documentation.¹⁰

2.12.4 Reduce IPL pauses

You can speed up the IPL process by changing some of the Pause statements in the relevant VTAMALL, VTAMAPPL, CTAMCI, VTAMDB, or VTAMD2 members of PARMLIB. These statements cause the command script to wait the indicated number of seconds before issuing the next command in the list. The times in the distributed list are suitable for a P/390 system. You can reduce the times for an MP3000 or other faster machines.

Be careful with the pause statements. It is necessary for some system components to be completely started before other components can be started. For example, OMVS must be completely started before TCPIP can be started. In this particular case, if OMVS is not completely started, TCPIP may fail. (You can later manually restart it.) In other cases, such as VTAM and TSO (TCAS), bad timing produces an operator message that requires a reply.

We suggest you approach changes by making a series of minor adjustments.

2.12.5 C compiler procedures

The procedure library for the C compiler is not in the standard list of procedure libraries. You can fix the problem by adding one line to the JES2 started task procedure, or by adding JCLLIB statements to all your C compiler jobs. To correct the JES2 procedure, edit member JES2 in SYS1.PROCLIB and add a line in the concatenation, like this:

```
...
//      DD DSN=EOY.SEOYPROC,DISP=SHR
//      DD DSN=CBC.SCBCPRC,DISP=SHR           <=== add this line
//      DD DSN=FAN130.SVSC.PROCLIB,DISP=SHR
...
```

2.12.6 Too many initiators

By default, release 10 AD starts ten initiators. This is probably too many for most AD users, especially those with P/390-based machines. You can change this by editing member JES2PARAM in PARMLIB. Before doing this, you should copy member JES2PARAM to new member JES2BACK. There is a simple method for starting JES2 using the parameters in JES2BACK instead of the parameters in JES2PARAM.¹¹ This is important if someone makes a serious error in JES2PARAM such that JES2 cannot start.

After making the backup copy of JES2PARAM, edit JES2PARAM and find the following lines:

```
INIT(1) NAME=1,           /* Comments ..... */
      CLASS=A,           /* More comments */
      START=YES         /* Yet more comments... */
                          /*
INIT(1) NAME=2,           /* Comments ..... */
      CLASS=AB,          /* More comments */
      START=YES         /* Yet more comments... */
                          /*
INIT(1) NAME=3,           /* Comments ..... */
      CLASS=ABC,         /* More comments */
      START=YES         /* Yet more comments... */
                          /*
.....
```

¹⁰ Unfortunately, not all PARMLIB members accept *commented out* lines; you should examine existing members for examples of where this is possible.

¹¹ See "How to change JES2 definitions" on page 184.

Change some of the START=YES parameters to START=NO. We have found that four initiators is reasonable for a P/390-based machine, but this must be adjusted for local conditions.

2.12.7 Web server does not start

In earlier releases the Web server used the started task name IMWEBSRV. This has been changed to HTTPD1. The name IMWEBSRV is still used in various RACF controls and other places.

If you IPL with the CS or 00 parameters (that is, with IPLPARM=0A82CS or 0A8200, for example), you will find the Web server fails to start properly. The VTAMAPPL command list contains an S HTTPD1, but the program fails a few seconds after starting. The problem is that the BPXPRMCS member in PARMLIB, which is used by the CS and 00 IPLPARM, does not mount the HFS data set needed by the Web server.

If you want to use the Web server with these IPLPARMs, edit member BPXPRMCS in PARMLIB and add the following lines somewhere in the series of mount commands:

```
MOUNT FILESYSTEM('HFS.USERS')
      TYPE(HFS)
      MODE(RDWR)
      MOUNTPOINT('/u')

MOUNT FILESYSTEM('HFS.WEB')           <=== add these lines
      TYPE(HFS)                       <=== add these lines
      MODE(RDWR)                       <=== add these lines
      MOUNTPOINT('/web')               <=== add these lines
```

2.12.8 /u has incorrect permissions and is very small

The directory entry (in UNIX System Services) for /u has permissions set to 700. This permits full access for users with UID=0, but prevents access to any files mounted under /u for all other users. This is probably not what you want. To correct this, log on with a userid id that has UID=0 (use TSO followed by OMVS or ISH, or telnet directly to OMVS) and issue the command from the OMVS shell:

```
# chmod 755 /u
```

Also, note that the allocated space for the /u file system is very small—only two tracks. You can create files in /u, but this will force secondary allocations for HFS.USERS. The secondary allocations are also very small, so this is not a reasonable approach for any significant use of /u. The intention is to force you to allocate other file systems (not on system packs) and mount these under /u, or to create your own /u file system. You will appreciate this when it is time to move to the next system release.

2.12.9 IPCS has nonexistent dumps in inventory

If you use IPCS (with the userid P390, which is the userid most AD system administrators use) you will notice a number of dumps in the inventory. These dumps do not exist on the disks and can be deleted from the inventory. To do this, go to the ISPF panels for IPCS, then to the inventory display. Enter DD for each inventory item to be deleted. In the next panel, enter

```
RECORDS = ALL
DELETE = YES                               (will create error; ignore it)
```

to delete the inventory entries.

2.12.10 Unexpected RACF parameters

The RACF configuration in the AD release 10 system may not be quite what would be expected for a typical, smaller OS/390 system. In particular:

- ▶ The *List of Groups* function is off. This is quite unusual; almost all OS/390 systems have this function on. We do not know why it was not turned on. The List of Groups function in RACF means that all the groups to which a userid belongs will be automatically searched for access permissions. If this is off, as it is in this AD system, then only the *current* group (which is typically the userid's default group) is searched.¹² You can enable the List of Groups function with the TSO command **SETROPTS GRPLIST**. You must have RACF SPECIAL authority to issue this command. You need issue it only once; it is remembered in the RACF data base and has system-wide effect for all users. We recommend you do this for your AD system.

List of Groups processing has a logical, but interesting effect on access permissions for UNIX System Services files. A RACF group may have a GID value. If a user is a member of several groups that have different GIDs (and if the List of Groups function is active) then the user has several active GID values. All of these will be checked when needed to satisfy the group permissions for a UNIX file.

- ▶ The Enhanced Generic Naming (EGN) function of RACF is off. It has been off in all the recent AD releases. The situation is not as clear as it is with List of Groups. Some customers also have EGN off. With EGN off, the RACF data set profile name *XYZ.** includes all data sets whose HLQ is XYZ. With EGN on, the profile name *XYZ.*** is needed to do the same thing. There is more to it than this, of course, but the differences have to do with fine-tuning data set profile names.

You can turn EGN on with the command **SETROPTS EGN**. This will automatically convert the existing data set profile names to the EGN format. This is usually considered to be a one-way conversion; you would not normally convert back to EGN off, although it can be done in some cases. As far as we can tell, most AD system users leave EGN off. Most larger OS/390 customers turn it on.

- ▶ Common classes are not RACLISTed. The TSOAUTH class, for example, is normally RACLISTed, but this has not been done in this AD release. The RACLIST function provides a small performance enhancement; it makes no other significant difference. Many examples in this book assume that TSOAUTH (and several other classes) are RACLISTed and include **SETROPTS RACLIST(TSOAUTH) REFRESH** commands after other RACF commands that alter profiles in the class. This refresh is not necessary (but does no harm) if the class is not RACLISTed. We include the refresh in the examples because almost all larger OS/390 installations will have these classes RACLISTed.

The command **SETROPTS RACLIST(TSOAUTH)**, for example, will RACLIST class TSOAUTH. On a small AD system the performance gain from RACLISTing is insignificant; the author generally leaves his systems as they are in the current AD package.

- ▶ The STARTED class is enabled, but is used only for DCE, Web server, and a few UNIX System Services functions. It is not used for the normal OS/390 started tasks. These are handled by having user profiles that match the started task name. These are readily seen in "User IDs" on page 30.
- ▶ There are no Global Access Checking (GAC) definitions. GAC provides a fast path through RACF checking for selected profiles and is almost always used in larger

¹² A user can select a group in the password panel during TSO logon. No selection means the user's default group is used. The user must already be a member of a particular group before he can select it during logon.

installations. Its only effect is performance related, and the effects on a small AD system would probably not be significant. See the RACF documentation for more information about enabling GAC. A small example is shown in “How to use RACF Global Access Control” on page 165.

- ▶ No default OMVS userid is defined.

2.12.11 Region size for SMP/E

SMP/E jobs, submitted from the ISPF SMP/E panels,abend due to insufficient storage. You can change the default region sizes for batch jobs (in JES2PARMS) or simply add a REGION=40M parameter to the SMP/E job cards that are generated by the panels. (You do not really need a 40 MB region; this number should be taken as a generic “large” region size.)

2.12.12 Web server requires RACF login

The Web server (started as HTTPD1) in the release 10 AD requires a RACF userid/password to use it. That is, when a browser is pointed to it, the browser user is immediately prompted for a userid and password. This is probably not what you want. Be certain you have the /web file system mounted (see “Web server does not start” on page 41), go to OMVS, and edit /web/httpd1/httpd.conf. Find the line (approximately line number 200) that says

```
UserId %%CLIENT%%
```

and change it to

```
UserId PUBLIC
```

This will cause Web users to be associated with RACF userid PUBLIC, which is defined and set up for this purpose, with appropriately limited privileges.

2.12.13 OMVS not *trusted*

The system HFS data sets have the HLQ of “HFS”. There is a RACF data set profile for HFS.* with a UACC of READ. OMVS, in order to provide file services for UNIX System Services users, needs UPDATE or ALTER access to the HFS data sets. In most OS/390 installations this is provided by using a STARTED class profile for OMVS, and by specifying the TRUSTED attribute in this profile. A started task with the TRUSTED attribute bypasses RACF access checks.¹³

The AD system does not use a STARTED class profile for OMVS and has no way to acquire the trusted attribute. Instead, OMVS is permitted (with a RACF **PERMIT** command) to the HFS.* profiles, with UPDATE access.

If you create new HFS data sets (each being a new file system), you have several choices:

- ▶ You can name your new data set with the HLQ of HFS. It will then be protected by the existing HFS.* generic profile, and OMVS can already access these data sets.
- ▶ You can name your new HFS with a different name and not RACF-protect the new HFS. This is not a good idea.
- ▶ You can name your new HFS anything and provide a RACF profile to protect it. You then need to PERMIT OMVS to this profile, with UPDATE access.

¹³ Note that OMVS itself (the kernel) would bypass checks. This is not extended to users of OMVS.

2.12.14 NFS server operation

We had a few problems with NFS operation. In particular, we discovered that the NFS server would loop if we stopped TCPIP. A subsequent command to stop NFSS (**P NFSS**) indicated that a shutdown was in progress, but it never ended the loop. We needed to cancel NFSS (**C NFSS**) in order to stop the loop. Stopping TCPIP is common, because you must stop it to edit the TCPIP profile data set. We could start NFSS again, and it appeared to work.

A common sequence went like this:

```
P TCPIP                (stop TCPIP)
P NFSS                (it indicates it is stopping, but does not)
C NFSS
edit the TCPIP profile, or do whatever else you intended to do with TCPIP down
S TCPIP                (wait until it completely starts)
S NFSS
```

Of course, if you do not need the NFS server, you can delete the start command in the various VTAMAPPL lists in PARMLIB.

2.12.15 Latest P/390 device managers

You should obtain the latest P/390 (or R/390, or Integrated Server) updates from the p390.ibm.com ftp site before attempting to use OS/390 TCP/IP in the release 10 AD system, or later. Changes to OS/390 TCP/IP code required fixes to the LCS3172 device manager.

In general, you should monitor this site for new fix packs and other fixes. A general fix pack, not yet available at the time of writing, is discussed in “P/390 fixpak” on page 4.

2.12.16 VTAMLST errors and additions

The A0600 member of SYS1.LOCAL.VTAMLST contains minor errors. Note that the AD system builders elected to create and concatenate SYS1.LOCAL.VTAMLST before SYS1.VTAMLST in the VTAM started task procedure. This LOCAL data set is not a traditional part of MVS, and you may not find it elsewhere. SYS1.VTAMLST is the assumed location for VTAM control parameters; you must be aware of the concatenation used in the AD system in order to update the correct library.

Also, these two VTAMLST libraries contain some duplicate member names. Be certain to update the correct member.

The next-to-last line in A0600 is missing a comma and a continuation character in column 72. This creates console error messages when VTAM starts. You should change line:

```
VPACING=1                                0212011
```

to

```
VPACING=1,                               C0212011
```

If you are using the ISPF editor, you will not see the sequence numbers in the last eight columns unless you scroll sideways.

The same member, A0600, contains an entry for each possible concurrent TSO session. There are only 10 members defined for this, limiting your system to 10 concurrent TSO sessions. You probably want to increase this. Find the following three lines:

```
*
A06TS0010 APPL AUTH=(NOACQ,NOBLOCK,PASS,NOTCAM,NVPACE,TSO,NOP0),          X
EAS=1,ACBNAME=TS00010
```

Replicate these three lines to make as many definitions as you want for concurrent TSO use. Then change two fields in each set of lines. For example, we wanted to have a maximum of 40 concurrent TSO users, so we replicated 30 copies of these lines (after the 10 copies that already exist). We then update each set to continue the same pattern of numbering. For example, we added:

```

*
A06TS0011 APPL AUTH=(NOACQ,NOBLOCK,PASS,NOTCAM,NVPACE,TSO,NOPO),          X
           EAS=1,ACBNAME=TS00011
*
A06TS0012 APPL AUTH=(NOACQ,NOBLOCK,PASS,NOTCAM,NVPACE,TSO,NOPO),          X
           EAS=1,ACBNAME=TS00012
*
A06TS0013 APPL AUTH=(NOACQ,NOBLOCK,PASS,NOTCAM,NVPACE,TSO,NOPO),          X
           EAS=1,ACBNAME=TS00013
.....

```

2.12.17 ISPF LITE procedure

The ISPF LITE TSO logon procedure is intended to provide a light-weight, fast TSO logon procedure, as contrasted with the ISPF PROC full-function logon procedure. ISPF PROC allocates ISPF libraries for RACF, ISMF, HCF, SMP/E, and other system programmer tools not needed by normal users. We found that ISPF LITE was a little too light, because it did not allocate the libraries needed to use SDSF. We added the following procedure as SYS1.TSOPROC(BILLPROC)¹⁴ and authorized any TSO user to use it:

```

/ /*****
//*
//*          BILLPROC FOR R10
//*
/ /*****
//ISPFPROC EXEC PGM=IKJEFT01,REGION=0M,DYNAMNBR=175,
//          PARM='%ISPFCL'
//SYSLBC DD DISP=SHR,DSN=SYS1.BROADCAST,UNIT=3390,VOL=SER=OS39M1
//SYSPROC DD DISP=SHR,DSN=SYS1.LOCAL.CLIST,UNIT=3390,VOL=SER=OS39RA
//          DD DISP=SHR,DSN=ISP.SISPCLIB,UNIT=3390,VOL=SER=OS39RA
//          DD DISP=SHR,DSN=SYS1.CLIST,UNIT=3390,VOL=SER=OS39M1
//          DD DISP=SHR,DSN=SYS1.SBPXEXEC,UNIT=3390,VOL=SER=OS39RA
//SYSEXEC DD DISP=SHR,DSN=ISP.SISPEXEC,UNIT=3390,VOL=SER=OS39RA
//          DD DISP=SHR,DSN=SYS1.SBPXEXEC,UNIT=3390,VOL=SER=OS39RA
//SYSHELP DD DISP=SHR,DSN=SYS1.HELP,UNIT=3390,VOL=SER=OS39RA
//          DD DISP=SHR,DSN=ISP.SISPHELP,UNIT=3390,VOL=SER=OS39RA
//ISPLIB DD DISP=SHR,DSN=ISP.SISPMENU,UNIT=3390,VOL=SER=OS39RA
//          DD DISP=SHR,DSN=SYS1.SBPXMENU,UNIT=3390,VOL=SER=OS39RA
//          DD DISP=SHR,DSN=ISF.SISFMLIB,UNIT=3390,VOL=SER=OS3RAA
//ISPEXEC DD DISP=SHR,DSN=ISP.SISPEXEC,UNIT=3390,VOL=SER=OS39RA
//          DD DISP=SHR,DSN=SYS1.SBPXEXEC,UNIT=3390,VOL=SER=OS39RA
//ISPLIB DD DISP=SHR,DSN=SYS1.LOCAL.ISPFPNLS,UNIT=3390,
//          VOL=SER=OS39RA
//          DD DISP=SHR,DSN=ISP.SISPPENU,UNIT=3390,VOL=SER=OS39RA
//          DD DISP=SHR,DSN=SYS1.SBPXPENU,UNIT=3390,VOL=SER=OS39RA
//          DD DISP=SHR,DSN=ISF.SISFPLIB,UNIT=3390,VOL=SER=OS3RAA
//ISPSLIB DD DISP=SHR,DSN=ISP.SISPSLIB,UNIT=3390,VOL=SER=OS39RA
//          DD DISP=SHR,DSN=ISP.SISPSENU,UNIT=3390,VOL=SER=OS39RA
//          DD DISP=SHR,DSN=ISF.SISFSLIB,UNIT=3390,VOL=SER=OS3RAA
//ISPTLIB DD DISP=SHR,DSN=ISP.SISPTENU,UNIT=3390,VOL=SER=OS39RA
//          DD DISP=SHR,DSN=SYS1.SBPXTENU,UNIT=3390,VOL=SER=OS39RA

```

¹⁴ There is nothing special about the name BILLPROC, of course. Whatever name you use should be specified in the corresponding RACF command. An alternative is to update the ISPF LITE procedure and not create a new procedure.

```

//          DD  DISP=SHR,DSN=ISF.SISFTLIB,UNIT=3390,VOL=SER=0S3RAA
//ISPCLT1 DD  DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=80,BLKSIZE=800,RECFM=FB)
//ISPCLT2 DD  DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=80,BLKSIZE=800,RECFM=FB)
//ISPLST1 DD  DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA)
//ISPLST2 DD  DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//          DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA)
//SDSFMENU DD  DSN=ISF.SISFPLIB,DISP=SHR

```

After placing this procedure in SYS1.TSOPROC, the following TSO RACF commands will permit all TSO users to use this logon procedure:

```

RDEF TSOPROC BILLPROC UACC(READ)
SETROPTS RACLIST(TSOPROC) REFRESH

```

The **REFRESH** command will indicate an error because TSOPROC is not RACLISTed in this AD release. In most systems it will be RACLISTed, and you should become accustomed to issuing a SETROPTS REFRESH command after changing this type of profile. To use BILLPROC, a user would simply specify this name in the procedure field of the TSO logon password screen.

2.12.18 Shutdown scripts

The SHUTDB script (in PARMLIB) contains an error. It issues a **C IMWEBSRV** command which should be **P HTTPD1**.

The SHUTALL script issues **F CICS,CEMT P SHUT I**. This has two errors: CICS runs under the name CICSA. Also, the default CICSA with the AD system does not have the MVS console interface fully installed. CICS can be stopped with the command **C CICS**. (It might be better to log onto CICS and use the CEMT transaction to shutdown CICS, but the simple cancel is effective.)

All three scripts (SHUTDOWN, SHUTDB, SHUTALL) contain the command **C NSFC**, which, as far as we can tell, is an error. Other than creating an error message, it does no harm.

2.12.19 UNIX command lp

The UNIX System Services command:

```
lp -d,x my.text.file
```

should spool the indicated file to JES2 (class X), where it could be inspected with SDSF and/or printed. The current system spools the file correctly, but when the user (who issued the lp command) attempts to look at the file with SDSF, he receives a *Not Authorized* response. The spooled file is in the JES2 *Hold* queue. An administrator with SPECIAL, OPERATIONS, OPER, and ACCT authority can view the spooled file.

2.12.20 Error in inetd procedure

The inetd program is usually started from /etc/rc. If necessary, it can be started from SYS1.PROCLIB, as a started task. The INETD member in PROCLIB uses BPXBATCH to start inetd the same way it would be started in /etc/rc. The PARM field in the INETD member contains two errors. It should be PARM='PGM /usr/sbin/inetd /etc/inetd.conf'. The distributed version has *usr* spelled as *use* and is missing the slash before *etc*.

2.13 Special IPL replies

You may see some of the following messages during IPL -- especially during your first IPL:

```
IXL011I XES HARDWARE SUPPORT IS NOT INSTALLED. REASON 01
IXC414I CANNOT JOIN SYSPLEX PLEX1 WHICH IS RUNNING IN MONOPLEX MODE:
      CONFIGURATION REQUIREMENT
IXC404I SYSTEM(1) ACTIVE OR IPLING: P390
IXC420D REPLY I TO INITIALIZE SYSPLEX PLEX1, OR R TO REINITIALIZE XCF
```

You should reply **REPLY 00,I** if you receive this message.

After an IPL is finished, you will have the following outstanding message on the MVS console. (After this message works its way to the top of the screen, you can issue the MVS operator command **K E,1** to remove it from the screen.)

```
*03 ISTEXC200 - DYN COMMANDS MAY BE ENTERED
```

The following commands may be entered in reply to the outstanding WTOR (message ISTEXC200) issued by the Dynamic XID Exit. Note that the command will not be processed and another WTOR will not be issued until next time that the exit is called by VTAM.

```
R XX,RELOAD
      (The current Dynamic XID mapping table is reloaded.)

R XX,DYNTAB=<table>
      (The Dynamic XID mapping table <table> becomes the current table.)
```

For further information see SYS1.README(VTAM). Very few people use this message function. When the DYN COMMANDS MAY BE ENTERED message scrolls to the top of the screen, you can remove it with a **K E,1** command.

2.14 Shutting down the system

The R10 system has three shutdown scripts. The proper script should be matched to the IPL parameter you used, as follows:

SCRIPT	IPL Parameter
SHUTDOWN	CS, 00
SHUTDB	DC, DB, CC, CW, 2C, 2W, IC, IW, DI
SHUTALL	AC, AW

In general, it will not hurt to use a more "powerful" shutdown script (such as SHUTALL) with a less "powerful" IPL parameter. The script will issue various stop and cancel commands for programs that might not be running, but (aside from an error message) this will do no harm.

To stop your system, you can issue **S SHUTDOWN** (or **S SHUTDB**, or **S SHUTALL**) from the Master Console. This may, or may not, produce a clean shutdown. You can edit the script in SYS1.ADCD10.PARMLIB(SHUTDOWN). You may want to, for example, reduce the 30 second delay after the message is sent to TSO users.

The shutdown script takes several minutes to execute. When it is finished you should be able to issue a **\$PJES2** command to stop JES2. A **Z EOD** command would complete the shutdown (although most people do not bother). A **QUIESCE** command, although not required, stops all OS activity.

2.15 Initial MVS console PFK assignments

The MVS console PFK assignments distributed with the AD CD-ROM are:

PFK	COMMAND.....	ACTION.....
1	D U,DASD,,000,999	Display DASD volumes (online and offline)
2	D U,,660,12	Display units at addresses 660 - 611
3	K	Clear main part of display ("top area")
4	K A,10	Create a 10-line Display Area ("bottom area")
5	D A,ALL	Display Address Space Information
6	D R,L	Display Outstanding Replies/Errors
7	D PFK,L=A	Display PF Keys
8	K D,F	Scroll Display Area ("bottom area")
9	D TS,L	Display TSO Users
10	D A,L	Display Active Address Spaces
11	D J,L	Display Active Jobs
12	K E,D	Remove Display Area ("bottom area")

These commands may not be completely appropriate for the AD CD-ROM. You can change the commands by editing SYS1.PARMLIB(PFKTAB00). PFKs 13-24 simply repeat the actions of keys 1-12.

2.16 ISPF primary menu

For the basic logon procedure (ISPFPROC) the second panel of the primary ISPF menu, accessed through the M (More) option on the first panel, contains:

1 SMP/E	System Modification Program/Extended
2 ISMF	Integrated Storage Management Facility
3 RACF	Resource Access Control Facility
4 HCD	Hardware Configuration Dialogs
5 SDSF	Spool Display and Search Facility
6 IPCS	Interactive Problem Control Facility
7 DITTO	DITTO/ESA
8 RMF	Resource Measurement Facility
9 DFSORT	Data Facility Sort
10 OMVS	MVS OpenEdition
11 RRS	Resource Recovery Services

Of these options, only SDSF is commonly accessed by end users. SMP/E, ISMF, RACF, HCD, IPCS, and RMF are the provinces of systems programmers. DITTO and the DFSORT panels might be accessed by normal system users, but this would probably be fairly rare. OMVS is usually started from ISPF Option 6. For these reasons, some installations prefer to move SDSF to the first panel.

If you use the DBSPROC logon procedure (and, presumably, IPLed with a parameter to include DB2 modules), the M panel appears as:

1 SMP/E	System Modification Program/Extended
2 ISMF	Integrated Storage Management Facility
3 RACF	Resource Access Control Facility
4 HCD	Hardware Configuration Dialogs
5 SDSF	Spool Display and Search Facility
6 IPCS	Interactive Problem Control Facility
7 DITTO	DITTO/ESA
8 RMF	Resource Measurement Facility
9 DFSORT	Data Facility Sort
10 OMVS	MVS OpenEdition
11 RRS	Resource Recovery Services

11	DB2PM	Database Performance Monitor
12	DB2DC	Database Data Collection
13	DB2	Data Base Product
14	RRS	Resource Recovery Services
15	DB2ADM	Database Admin Tool
16	QMF	Query Management Facility

2.17 Supplied jobs

A number of jobs are provided in SYS1.P390.CNTL. These should be used with care because many of the jobs correspond to older versions of the AD CD-ROM system and may not be exactly what you want for the current release. Nevertheless, this data set provides examples of many jobs you might need.



z/OS 1.1 AD CD-ROM

This chapter discusses specific details of the a/OS 1.1 AD system. *This chapter nearly duplicates chapter 2, changing only the details that are different between the OS/390 R10 and the z/OS R 1.1 AD systems.*

3.1 Specific product content

The following list contains all the components of the z/OS R1.1 AD system. This list was taken from the documentation provided by the PID group in Dallas that currently builds the AD releases. The list is presented here for completeness, but it may not be overly helpful to newer users of OS/390. Many of the listed elements are, in effect, internal parts of OS/390 that you would never work with as separate functions. OS/390 is structured this way for historical reasons. The FMID numbers are used by the maintenance tool, SMP/E, and are sometimes used by systems programmers to indicate exactly which part of the system (and which version of this part) they are discussing.

FMID	DESCRIPTION	FMID	DESCRIPTION
HBB7703	Base Control Program 2.10	HJE7703	JES2 Base
JBB7713	z/OS Dependent Base	HSLM704	z/OS License Manager
HMSI703	MSI Setup - Framework	JMSI713	MSI Plug-in
JMSI723	MSI Plug-in for RACF		
HSWF100	ESCON Director Support	HI01104	I/O Control Program
EMI2220	MICR/OCR	HBD6602	Bulk Data Transfer
JBD6201	Bulk Data Trans file-file	HDZ11F0	DFSMS base V210
HDZ11FH	DFSMS HFS?	JDZ11FB	DFSMS Emu
EER3500	EREP V3R6	HQM4140	High Level Assembler base 1.4
EDU1G01	ICKDSF R16	FDU1G07	ISMF ICKDSF panels
FDU1G08	ISMF ICKDSF Panels	HIF5A02	ISPF base
HBCNC00	SMP/E SW info base	HTE26D2	TSO/E REXX
JTE26D0	TSO/E EDIT	HFX1112	3270 PC file transfer 1.1.1
HFST101	FFST for MVS 1.2	ETI1106	Terminal I/O Control
H24P111	Host Perf analysis base En	HCS6091	HCD base
JCS6092	HCD HFS	HCM1410	Hardware Config Mgr
HMP1C00	SME/E base	JMP1C0B	Planning & Migration Assist
H25I625	Tivoli Frame Endpoint 3.6.1	HCPT2A0	Secure Socket Layer base
HCRY2A0	Open Crypto Serv Facility	HCR7703	ICSF
HLB7703	C Compiler base	JLB7713	C++ Compiler
HLC7703	Open Class Lib source base	HTV7703	C/C++ Open Class Lib base
JTV7713	Applic Support & C Cls Lib	JTV7733	Open Class Library HFS
HLE7703	Language Environment base	JLE77H3	Language Environment HFS
H08K410	SOM Objects RTE Lib	HDHC170	DCE Application support base
HTKT170	Encina Toolkit Exec base	HDZ11TC	NFS Client
HDZ11TS	NFS Server	HMB3190	DCE base services (OSF DCE 1.1)
HOH2310	DFS (OSF DCE 1.2)	HTCP52A	CS Security Level 2
JTCP5PA	CS High Speed Access Srv	JTCP56A	CS Network Print Facility
JTCP59A	CS HFS	HVT510A	CS SNA
HTCP38X	SecureWay CS X Windows 11.4	HTCP50A	CS IP (TCP/IP)
HIMW530	WebSphere Applic Serv	HIMW530	HHTTP Server base 5.2
HIMN230	TextSearch, NetQuestion Sol	HMR7170	LANRES Host
HQW7500	Lan Server MVS	HOGI300	OSA/SF
HPMG110	USS process manager	HCMG110	USS connection manager
HOT7703	USS Application Serv base	HSAP2A0	USS integrated call lvl interface
HBKM300	READ/MVS 1.3 base	HBKP300	BUILD/MVS 1.2 base
JBKM310	READ/MVS 1.3 English	JBKP310	BUILD/MVS 1.3 English
HBKQ220	BookManager BookServer 2.2	HGD3200	GDDM/MVS base 3.2.0
JGD3219	GDDM/MVS NLS 3.2.0	HMOS280	Infoprint Serv-IP PrintWay base
HNET280	Infoprint Serv - NetSpool base	HPOI280	Infoprint Serv - Print Interface
HJS7703	JES3	JBD6202	Bulk Data Transfer SNA NJE
HRM7703	RMF base	HQX7703	SDSF base
HRF7703	SecureWay - RACF base	HFWL2A0	SecureWay - Firewall base
HRSL2A0	SecureWay - LDAP Serv base	HRSS190	SecureWay - DCE base
HR062A0	SecureWay - Open Crypto plugin	HCKVB00	C/C++ debug tool base
JCKVB03	C/C++ debug tool messages	HSM1G00	DFSORT base R14
JSM1G01	DFSORT/ISPF English	HGD3201	GDDM/MVS/PGF 2.1.3

JMQ414A	High Level Assem toolkit 1.4	H08K411	SOMobjects development env
HJVA130	IBM Developer Kit Java2 OS/390		
HSWK2A0	SecureWay - Network Auth & Pri	HJVA11G	Java 1.01.08
HJVA11H	Java RAWT 1.01.08	HJVA11I	Java Swin 1.01.08
H249100	COBOL for MVS 1.02.00	J249101	COBOL for MVS 1.02.00
H080M10	VisualAge COBOL Millennium EXT	HOU1110	Millennium runtime 1.01.00
HOA5201	VA Java Compiler 2.00.00	HOA5202	VA Java runtime 2.00.00
HMWL131	AD/Cycle PL/I full funct 1.1.1	HWK0130	REXX/370 Compiler 1.03.00
JWK0131	REXX/370 Compiler English	HWJ9130	REXX/370 Library 1.03.00
JWJ9131	REXX/370 Library English	HWJ9133	rex/370 Alt Library English
H01B312	Net Commerce 3.01.01	HPZ8400	TME 10 NetView base 1.03.00
JPZ8404	TME 10 NetView US English	JPZ8407	TME 10 NetView USS English
JPZ8420	TME 10 NetView Procedural base	JPZ8424	TME 10 Netview Proc English
JPZ8440	TME 10 NetView Graphical base	JPZ8454	TME 10 NetView Graphical USS
H0GB310	DITTO/ESA 1.03.00	HMQ2100	MQSeries 2.01.00
HCI6100	CICS base for z/OS	JCI6101	COBOL Language parts
JCI6102	PL/I Language parts	JCI6103	C Language parts
JCI610D	IIOP/Java parts		
HCP2100	CICSplex SM base	JCP2102	CICSplex (SAS comp)
HBDD110	CICS Applic migration aids		
HOB5110	CICS REXX runtime facility	HOB7110	CICS REXX development system
HOZ2110	CICS REXX common - CICS/ESA	HOAL200	Tivoli GEM Instrum for CICSplex
HMK7700	IMS/ESA System Services V7	JMK7701	IMS/ESA Database Manager
JMK7702	IMS/ESA Transaction Manager	JMK7703	IMS/ESA ETO
JMK7704	IMS/ESA Recovery-level tracking	JMK7705	IMS/ESA Database-level tracking
JMK7708	IMS JAVA		
HIR2101	IMS/ESA IRLM 2.1	HDB7710	DB2 base TSO 7.1.0
HIY7710	DB2 IMS attach	HIZ7710	DB2 subsystem init
HDB771A	DB2 online help	JDB7712	DB2 JDBC/SQLJ support
JDB6602	DB2 JDBC V6 tool	HOG9710	DB2 Performance monitor
JDB7714	DB2 English panels	JDB7717	DB2 ODBC base
JDB771B	DB2 IAV extenders	JDB771C	DB2 text extender
JDB771X	DB2 XML		
JOG971X	DB2 Perf Monitor BUY		
H0IH210	DB2 Admin Tool	JOIH211	DB2 Admin Tool BUY
HSQ7710	QMF V7		

We can roughly summarize this list by noting that this AD release includes all the standard components of z/OS R1.1, many optional components (such as the Security Server (RACF), JES3, compilers, and so forth). It also includes the following software products that are not packaged as part of OS/390:

- ▶ Java for OS/390, release 1.1.0 (5655-A46)
- ▶ IBM Development kit for OS/390 Java 2 1.1.0 (5655-D35)
- ▶ Millennium Runtime (5697-E07)
- ▶ COBOL for MVS, release 1.02.00 (5655-A25) and associated products
- ▶ Visual Age Java for OS/390 compiler and runtime, release 2 (5655-JAV)
- ▶ SAA AD/Cycle PL/I compiler (full package), release 1.01.01 (5688-235)
- ▶ REXX/370 compiler and libraries, release 1.03.00 (5695-013 and 5695-014)
- ▶ Net.Commerce for OS/390, release 3.01.01 (5697-D32)
- ▶ Many TME 10 NetView elements, release 1.03.00 (5697-B82)
- ▶ DITTO/ESA, release 1.03.00 (5655-103)
- ▶ MQSeries for OS/390, release 2.01.00 (5655-A95)
- ▶ CICS for z/OS (5697-E93) with many optional elements
- ▶ IMS/ESA, version 7 (5655-B01), with optional elements
- ▶ DB2, version 7 (5675-DB2), with optional elements (including QMF) and several additional products related to DB2

This list indicates the order numbers, such as 5655-A46, that would be used to order the products separately. The order number is sometimes used in discussions about specific versions of these products.

3.2 Volumes distributed on CD-ROMs

The following table provides a brief overview of the volumes (and their associated CD-ROMs) included with the z/OS R1.1 AD system.

CD	Volser	Type	Used	Emul	Prim	Content
1	Z1RES1	3390-3	80%	A80	AA0	IPL, system libraries
2	Z1RES2	3390-3	61%	A81	AA1	More system libraries
3	OS39M1	3390-3	36%	A82	AA2	IODF, SYS1.IPLPARM, VSAM, etc
4	Z1USS1	3390-2	71%	A87	AA7	System HFS data sets
5	Z1DIS1	3390-3		A85	AA5	System DLIBs
6	Z1DIS2	3390-3		A86	AA6	More system DLIBs
7	Z1DIS3	3390-2		A88	AA8	More system DLIBs
8	Z1DB21	3390-2		A83	AA3	DB2, including DLIBs and HFS
9	Z1CIC1	3390-2		A84	AA4	TS (CICS), including DLIBs and HFS
10	Z1IMS1	3390-2		A89	AA9	IMS, including DLIBs and HFS
11	Z1WAS1	3390-2		A8A	AAA	WebSphere components
12	Z1WAS2	3390-2		A8B	AAB	WebSphere components
13	(none)					FAQs, redbooks

A few notes are needed to explain this table:

- ▶ Note that there is a mixture of 3390-2 volumes (1.9 GB) and 3390-3 volumes (2.8 GB). The smaller device type was used where possible.
- ▶ There is considerable free space on some of the volumes. If you are not concerned about migration issues (migration to the next AD release, that is), you could use some of this space for your own data sets. We generally recommend against doing this because it almost always leads to migration problems. (“How do I save all my data sets on the old volumes that I need to delete in order to have space to install the next release?”) Also, you should always preserve some free space on system volumes for PTFs and other urgent fixes that should reside on these volumes.
- ▶ The *Emul* and *Prim* columns above refer to the assumed addresses (device numbers) if the volumes are on emulated drives (always the case for P/390s, and an option for MP3000s), or on MP3000 primary drives. Again, the use of these addresses is not required.
- ▶ Only the first four volumes in this list are required to IPL the system. You need not install any of the other volumes unless you need the specific contents of the volume.
- ▶ If you need to access the system DLIBs (to install PTFs, for example), you will probably need to install all three DLIB volumes because there is no definition or documentation about which libraries are on which DLIB volume.

The last CD (number 13) does not contain a z/OS volume. It contains PDFs for several redbooks, including:

- ▶ SG24-4721: *Accessing OpenEdition from the Internet*
- ▶ SG24-4747: *P/390, R/390, S/390 Integrated Server: OS/390 New User's Cookbook*
- ▶ SG24-4847: *P/390 and R/390 with OS/390: An Introduction*
- ▶ SG24-4894: *The Universal Connectivity Guide to DB2*
- ▶ SG24-5277: *Revealed! CICS Transaction Gateway with More CICS Clients Unmasked*
- ▶ SG24-5633: *Multiprise 3000: Technical Introduction*
- ▶ SG24-5669: *Multiprise 3000: Basic Emulated I/O Definitions*

The last CD also contains a number of files, in HTML format, containing frequently asked questions related to the AD systems. If your CD-ROM drive is H, you can access these files by pointing a browser to `file:///H:/faq/` and selecting the various files, one at a time. About half the files contain index titles and the other half contain the questions and answers.

3.2.1 Disk space required

The AD system no longer includes mini disks¹; all the volumes are in standard 3390 sizes. For z/OS R1.1, the four volumes needed to IPL require 10.3 GB disk space (in the form of three 3390-3 volumes and one 3390-2 volume), or 11 SUs of MP3000 primary space. The system DLIBs require 7.5 GB (8 SUs). The DB2, CICS, IMS, and WebSphere volumes *each* require 1.9 GB (2 SUs each).

The complete system requires approximately 27 GB disk (or 29 SUs).

3.3 Parmlibs

In the good old days, most system parameters were in SYS1.PARMLIB, and this was a magic, well-known name. OS/390 allows multiple PARMLIBs and does not require that the names contain the qualifier "PARMLIB", except that SYS1.PARMLIB is always included. z/OS will concatenate whatever libraries you name (in the LOADxx member you selected via bytes 5 - 6 in your IPL parameter) as parameter libraries and will automatically add SYS1.PARMLIB to the end of the concatenation (if you do not specify it elsewhere in the concatenation). The data set names of the system parameter libraries and the order of concatenation of the libraries are critical information for anyone maintaining or customizing z/OS.

The z/OS R1.1 system contains two system parameter libraries:

```
SYS1.ADCD01.PARMLIB
SYS1.PARMLIB
```

and they are concatenated in the order shown. All of the members customized for the AD system are in the first library. However, default members (used by the AD system but unchanged from the original ServerPac) may be in the second library. You should always check to see if a member is in the first library; if not then look in the second library.²

You can display the current parameter library concatenation list with the MVS operator command **D PARMLIB**.

¹ P/390-based systems can define 3380 and 3390 drives with any number of cylinders, to a maximum of 3339 usable cylinders (for 3390s). Early versions of the AD system used odd-sized volumes to conserve disk space on the P/390 machines. For example, if the contents of a DLIB volume used 1600 cylinders, then the distributed volume would be, say, 1700 cylinders. (The additional cylinders allow for reasonable PTF activity and customization.) The primary disks of the MP3000 do not provide odd-sized volumes. (Emulated disks on the MP3000 do support odd-sized volumes.) With many PID members moving to the MP3000, the group building the AD systems elected to distribute only standard-sized volumes. Doing this also saves time when building a new AD release.

² You could use the ISRDDN function of ISPF instead. This handles multiple concatenated libraries, in the correct order of concatenation.

In the remainder of this redbook, we will use the term PARMLIB to mean the concatenated list of system parameter libraries, without specifying exactly which library contains the member being discussed.

3.4 IPL parameters

This release has added more IPL LOADxx members. In the follow discussion, we assume that the IPL address (for the Z1RES1 volume) is A80 and that the IPLPARAM points to address A82 (for the OS39M1 volume). Your first IPL would use the parameters:

```
IPL address = A80      IPL Parameter = 0A82CS
```

The following discussion is about the fifth and sixth characters in the IPL Parameter, shown as CS in this example. These two characters are used to access a member in SYS1.IPLPARAM. The two characters are used as a suffix for the word "LOAD". In this example, the z/OS load process would reference SYS1.IPLPARAM(LOADCS). The LOADCS member has values that control other startup parameters. The general process is described in more detail in "Understanding IPL parameters" on page 103. In this section we will describe the particular parameters (the two characters) that are provided with the z/OS R1.1 system.

The goal of the AD system is to permit you to *cold start* or *warm start* your system in a number of ways. As used here, *cold start* means to perform a CLPA and then to clear the JES2 spool contents.³ *Warm start* means to use the existing contents of the PLPA (established by an earlier cold start), and to retain the existing contents of the JES2 spool.

Several subsystems need to place modules in PLPA. For example, if you want to use DB2, then you need to include several DB2 modules in PLPA. If you do not use DB2, then you probably want to omit these modules from your PLPA, to save space in common storage.

The various load parameters provided with release 10 allow you to use combinations of no DB/OLTP or DB2 or CICS + DB2 or IMS and so forth. For each combination, there is a cold start and a warm start option. You must always IPL with the cold start option; later IPLs can then be warm starts. If you then want a different mixture of subsystems, you must cold start with the appropriate load parameter. Thereafter you can warm start with the matching parameter.

The supplied parameters are:

Cold Start	Warm Start	Purpose.....
CS	00	Basic system. No CICS, IMS, DB2, MQ
DC	DB	Basic system + DB2, CICS (and RRS)
CC	CW	Basic system + CICS (and RRS)
IC	IW	Basic system + IMS (and RRS)
2C	2W	Basic system + DB2 (and RRS)
	DI	Basic system + IMS (and RRS). CLPA + JES2 warm start
AC	AW	Basic system + DB2, CICS, IMS, MQ (and RRS)
BC	BB	Basic system + DB2 + WebSphere

Again, remember that you must CLPA (cold start) a given DB2/IMS/CICS/MQ/WebSphere environment before you can later warm start that environment. A CLPA destroys the previous contents of PLPA. (The DI parameter, in this list, is a mixture of CLPA and a JES2 warm start. We are not quite clear how it came to be included in the release 10 AD system, but it illustrates the principle that you can customize any startup arrangement you want.)

³ The various startup combinations also select different SVC lists, different lists of subsystem properties, and so forth, in PARMLIB. The different PLPA libraries are the most obvious differences between the various IPL parameters and we usually refer to them in discussions, but this should not be taken to exclude the other differences in the various IPL parameter controls.

The AD designers assume that, if you IPL with a parameter that includes, say, DB2 and CICS modules, then you want to automatically start these subsystems. This is done by linking different VTAMAPPL command lists indirectly to the IPL parameter. There are other ways to manage your system. The author, for example, prefers to have his system capable of running CICS and DB2, but does not want to start these subsystems automatically. The DC and DB parameters load the correct modules, but then automatically start DB2 and CICS. You can change this by making a modified set of IPL parameters or by simply removing the start commands from the relevant VTAMAPPL command lists.

3.4.1 Detailed IPL parameter controls

The LOADxx member selected by the IPL parameter usually indicates which IEASYSxx member (in PARMLIB) to use. The IEASYSxx member then indicates a variety of other PARMLIB members to use. The following table attempts to summarize key elements of various IPL parameters.

Cold Start	CS CLPA JES Cold	DC CLPA JES Cold	CC CLPA JES Cold	2C CLPA JES Cold	IC CLPA JES Cold	BC CLPA JES Cold	DI CLPA JES Warm	AC CLPA JES Cold
Warm Start	00	DB	CW	2W	IW	BW		AW
IEASYSxx	CS or 00	DC or DB	CC or CW	2C or 2W	IC or DI	BC or BW	DI	AC or AW
COUPLExx	00	00	DB	DB	DB	DB	DB	DB
COMMNDxx S VTAMxxxx	CS or 00 VTAMAPPL	DC or DB VTAMDB	CC or CW VTAMCI	2C or 2W VTAMD2	IC or DI VTAMAPPL RRS	BC or BW VTAMBB	DI VTAMAPPL RRS	AC or AW VTAMALL
LPALSTxx (PLPA)	00	DB CICS	DB CICS	D2	IM	D2	IM	DB CICS
IEALPAXx (MLPA)	00	00	00	00	DI	00	DI	DI
BPXPRMxx	CS hpj NCommerce Payment	00 CICS DB2 WEB WAS CB390 JAVA NCommerce Payment hpj	00 hpj NCommerce Payment	DB DB2 WEB hpj JAVA NCommerce Payment WAS	00 CICS DB2 WEB WAS	BB CICS DB2 WEB WAS JAVA hpj	00 CICS DB2 WEB WAS	00 CICS DB2 WEB WAS
PROGxx	00	DB DB2 CICS	CI CICS	D2 DB2	IM IMS	BB	IM IMS	AL DB2 IMS CICS
SCHEDxx	00	DB	DB	DB	DB	(DB,BB)	DB	DB
IEASVCxx	00	CI CICS	CI CICS	00	IM IMS		IM IMS	AL IMS CICS

Consider the column with the headings CC and CW. The controls in this column are invoked when your IPL parameter is 0A82CC (for a cold start) or 0A82CW (for a warm start). It starts by using the LOADCC (or CW) member of SYS1.IPLPARM. This member invokes IEASVCC (or IEASVSCW) in PARMLIB. The IEASYSxx member points to IEASVCCI; of interest in this list is the inclusion of the CICS SVC. LPALST DB is used, and it contains a

CICS library. IEALPA00 is used. BPXPRM00 is used, and it mounts file systems for CICS, the Web server, and WebSphere (WAS). The SCHEDDB member is used. PROGCI is used, and it includes the CICS libraries. This table is intended as a quick reference to the key values implied by each IPL parameter selection.

The LOAD99 member is exactly the same as the LOAD00 member except it uses SYS1.IODF99 instead of SYS1.IODF01. The LPALST members D2, 00, and IM appear to be exactly the same.

The following table outlines the functions started by the various VTAMAPPL members pointed to by the COMMNDxx member. (This table is meant for comparing functions. The various start commands are issued in slightly different orders in the actual VTAMAPPL members.) These members (VTAMALL, VTAMAPPL, and so forth) are in SYS1.ADCD01.PARMLIB.

VTAMALL	VTAMAPPL	VTAMBB	VTAMCI	VTAMDB	VTAMD2
S RRS		S RRS	S RRS	S RRS	S RRS
S APPC	S APPC	S APPC	S APPC	S APPC	S APPC
S ASCH	S ASCH	S ASCH	S ASCH	S ASCH	S ASCH
S LLA	S LLA	S LLA	S LLA	S LLA	S LLA
S TSO	S TSO	S TSO	S TSO	S TSO	S TSO
S IRRDPTAB	S IRRDPTAB	S IRRDPTAB	S IRRDPTAB	S IRRDPTAB	S IRRDPTAB
S SDSF	S SDSF	S SDSF	S SDSF	S SDSF	S SDSF
S TCPIP	S TCPIP	S TCPIP	S TCPIP	S TCPIP	S TCPIP
-start DB2		-start DB2	-	-start DB2	-start DB2
		(see text)			
S HTTPD1	S HTTPD1	S HTTPD1	S HTTPD1	S HTTPD1	S HTTPD1
S NFSS	S NFSS	S NFSS	S NFSS	S NFSS	S NFSS
S CICS			S CICS	S CICS	
S IMS71IRL1					
S IMS71CR1					
%CSQ start QMGR					
%CSQ1 start chinit					
%CSQ1 start listener					

The VTAMBB script includes the following commands after starting DB2

```

SETPROG LPA,ADD,MASK=*,DSNAME=BB0401.SBBOLPA
TRACE CT,WTRSTART=BBOWTR
F WLM,MODE=GOAL

```

These tables are not complete--complete tables would be much too large to list--but they quickly identify the more important differences among the supplied IPL parameters.

Read horizontally, it indicates, for example, that PARMLIB members BPXPRMCS, BPXPRM00, BPXPRMCI, BPXPRMDB, and BPXPRMBB are used. Other BPXPRMxx members in PARMLIB are not used and could be deleted or customized and then used.

You can add your own LOADxx members and use or expand the PARMLIB members in any way that suits your needs.

Most of the examples in this document were produced using a basic system; that is, one without DB2, CICS, IMS, MQ, or WebSphere. For the basic system, we IPLed (once) with parameter OA82CS and thereafter IPLed with OA8200. (For the examples run on our MP3000, these parameters were OAA2CS and OAA200.)

3.5 User IDs

The z/OS R1.1 AD system is provided with many userids. We assume not all of them are intended for use.

USERID	Def	GRP	ATTRIBUTES	INTERVAL	TSO	OMVS	Notes:
IRRCERTA			REVOKED	N/A	no	no	Certificate anchor
IRRMULTI			REVOKED	N/A	no	no	Criteria anchor
IRRSITEC			REVOKED	N/A	no	no	Site Anchor
APPC	SYS1			180	no	no	
ASCH	SYS1		REVOKED	180	no	no	
BLSJPRMI	SYS1		REVOKED	180	no	no	
BPXOINIT	SYS1			180	no	UID=0	
CBACRU1	CBCTL1			180	no	UID=2107	
CBACRU2	CBCTL1			180	no	UID=2115	
CBADMIN	CBADMGP			180	no	UID=2103	
CBASRU1	CBASR1			180	no	UID=2110	
CBASRU2	CBASR2			180	no	UID=2116	
CBDMNCR1	CBCTL1			180	no	UID=2111	
CBGUEST	CBCLGP			180	no	UID=2102	
CBINTCR1	CBCTL1			180	no	UID=2114	
CBINTSR1	CBSR1			180	no	UID=2106	
CBIVP	CBIVPGP			180	no	UID=2109	
CBIVP2	CBIVPGP2			180	no	UID=2117	
CBLDAP	CBLDAPGP			180	no	UID=2119	
CBNAMCR1	CBCTL1			180	no	UID=2113	
CBNAMSR1	CBSR1			180	no	UID=2105	
CBSYMCRI	CBCTL1			180	no	UID=2112	
CBSYMSR1	CBSR1			180	no	UID=2104	
CICSA	SYS1		REVOKED	180	no	UID=0	
CICSTART	SYS1		REVOKED	180	no	no	
CSQ1CHIN	SYS1		OPERATIONS	180	no	UID=0	
CSQ1MSTR	SYS1		OPERATIONS	180	no	no	
DB2iRLM	SYS1		REVOKED	180	no	no	
DCEADM	SYS1		REVOKED	180	no	UID=0	
DCEKERN	DCEGRP		REVOKED	180	no	UID=0	
DFS	DFSGRP		REVOKED	180	no	UID=0	
DFSCM	DFSGRP		REVOKED	180	no	UID=0	
DSN1DBM1	SYS1			180	no	no	
DSN1DIST	SYS1			180	no	UID=0	
DSN1MSTR	SYS1			180	no	no	
DSN1SPAS	SYS1		GRPACC	180	no	no	JCL, RECOVER
DUMPSRV	SYS1		REVOKED	180	no	no	
EZAFTPAP	OMVSGRP			180	no	UID=0	
FTPD	SYS1			180	no	UID=0	
GTF	SYS1		REVOKED	180	no	no	

IBMUSER	SYS1	SPEC, OPER	N/A	yes	UID=0	ACCT,JCL,MOUNT,OPER,RECOVER
IMS	SYS1		180	yes	no	
IMSA	SYS1		180	no	no	
IMS61CR1	SYS1		180	no	no	
IMS61DL1	SYS1		180	yes	no	
IMS61F11	SYS1		180	yes	no	
IMS61F12	SYS1		180	yes	no	
IMS61F13	SYS1		180	yes	no	
IMS61M11	SYS1		180	yes	no	
IMS61RC1	SYS1		180	yes	no	
IMS61RD1	SYS1		180	yes	no	
IMS71CR1	SYS1		180	no	no	
IMS71DL1	SYS1		180	no	no	
IMS71F11	SYS1		180	no	no	
IMS71F12	SYS1		180	no	no	
IMS71F13	SYS1		180	no	no	
IMS71M11	SYS1		180	no	no	
IMS71RC1	SYS1		180	no	no	
IMS71RD1	SYS1		180	no	no	
IMS71RL1	SYS1		180	no	no	
IMS71RL2	SYS1		180	no	no	
IMS710	SYS1		180	no	no	
IMWEBSRV	SYS1		180	no	UID=0	
INETID	SYS1		180	no	no	
INIT	SYS1	REVOKED	180	no	no	
INTERNAL	EMPLOYEE		180	no	UID=537	
IRRDPTAB	SYS1	REVOKED	180	no	no	
IVPGNJCL	SYS1		180	yes	no	
JESB	SYS1	REVOKED	180	no	no	
JES2	SYS1	REVOKED	180	no	no	
LARRYWD	SYS1	SPEC,OPER	180	yes	UID=0	
LLA	SYS1	REVOKED	180	no	no	
NFSC	SYS1		180	no	UID=0	
NFSS	SYS1		180	no	UID=0	
OMPROUTE	OMVSGRP		180	no	UID=0	
OMVS	SYS1	REVOKED	180	no	UID=0	
OMVSKERN	OMVSGRP	REVOKED	180	no	UID=0	
OPEN1	SYS1		180	yes	UID=0	RECOVER,JCL
OPEN2	SYS1		180	yes	UID=0	RECOVER,JCL
OPEN3	SYS1		180	yes	UID=0	RECOVER,JCL
OSNMPD	OMVSGRP		180	no	UID=0	
PORTMAP	SYS1		180	yes	UID=0	
PRIVATE	SPECIAL		180	no	UID=416	
PUBLIC	EXTERNAL		180	no	UID=998	
P390	SYS1	SPEC, OPER	N/A	yes	UID=0	ACCT,OPER,JCL
P390A	TEST	SPEC, OPER	N/A	yes	no	OPER,JCL,RECOVER
P390B	TEST		30	yes	no	JCL,RECOVER
....						
P390Z	TEST		30	yes	no	
RACF	SYS1	REVOKED	180	no	no	
RMF	SYS1	REVOKED	180	no	no	
ROUTEDMV	OMVSGRP	REVOKED	180	no	UID=447	
ROUTEDOE	OMVSGRP	REVOKED	180	no	UID=448	
SNMPQE	SYS1		180	no	UID=0	
STCRACF	SYS1		180	no	no	
SYSADM	SYS1		180	yes	no	JCL,RECOVER
SYSLOGD	OMVSGRP		180	no	UID=0	
SYSOP	SYS1		180	yes	no	
SYSOPR	SYS1	SPEC,GRPACC	180	yes	no	JCL,RECOVER
TCAS	SYS1	REVOKED	180	no	no	

TCPIP	SYS1		180	yes	UID=0	
TCPIPMVS	OMVSGRP	REVOKED	180	no	UID=445	
TCPIPOE	OMVSGRP	REVOKED	180	no	UID=0	
TESTER	TEST		30	yes	no	RECOVER
VLF	SYS1	REVOKED	180	no	no	
VTAM	SYS1	REVOKED	180	no	UID=0	
WEBADM	IMWEB		180	no	UID=206	
WEBSRV	IMWEB		180	no	UID=0	

There are several interesting points in this list:

- ▶ The column labeled TSO indicates whether the userid is authorized for TSO; that is, does it have a TSO segment in RACF. The column labeled OMVS indicates whether the userid is authorized to use UNIX System Services; if so, the assigned UID is shown.
- ▶ The Notes column indicates additional TSO authorities the userids may have, such as OPER and so forth.
- ▶ A group of invalid userids (lacking RACF user definitions) are defined for many of the TSO authorities. We assume these are left over from the group that built the AD release.
- ▶ The system was built for convenience, not security. Almost all the userids have their default passwords, which are equal to their default group names. For example, you can log onto SYSOPR (which has RACF SPECIAL authority) with password SYS1. Or, you can log in by telnet to Unix System Services (TCP/IP port 1023, once you have OS/390 TCP/IP operational) with userid TCPIP, password SYS1. To better secure the system, you would need to change the passwords of all userids that are not REVOKED.⁴
- ▶ Many userids are marked REVOKED. This prevents any direct usage of these userids. In general, these REVOKED userids are used as started task names and are valid for that purpose.⁵
- ▶ Note the considerable number of userids with UID=0. These mostly represent system components, rather than end user userids.
- ▶ The specific non-zero UIDs assigned have no numeric significance; you must avoid reusing the same UIDs for your users.
- ▶ Not all the userids are necessary; some are relics of earlier AD systems. Do not assume there is a strong reason for all the userids present, or for the distribution of various authorities.

For completeness, we list the significant definitions in the RACF FACILITY class. The R, U, and A indicators are for access levels READ, UPDATE, and ALTER:

Profile	UACC	Permitted users (or owner).....
BPX.DAEMON	NONE	IBMUSER(A),OMVSKERN(R),WEBSRV(R),FTPD(R),TEST1-6(R)
BPX.FILEATTR.APF	NONE	IBMUSER(A),P390(R)
BPX.FILEATTR.PROGCTL	NONE	IBMUSER(R),P390(R),TEST2(R)
BPX.SERVER	NONE	P390(A),WEBSRV(U),IMWEBSRV(U)
BPX.SUPERUSER	NONE	IBMUSER(A)
DCE-----		(several profiles for DCE-related controls)
IRRDPIO0	READ	P390(A)
MVSADMIN.WLM.POLICY	READ	IBMUSER(U),P390(U),DPACK(U)
STGADMIN.ADR.*	NONE	P390(A),SYS1(A)
STGADMIN.IDC.*	NONE	P390(A),SYS1(A)
STGADMIN.IGD.*	NONE	P390(A),SYS1(A)
STGADMIN.IGG.*	NONE	P390(A),SYS1(A)
STGADMIN.*	NONE	P390(A),SYS1(A)

⁴ Userids that do not have a TSO segment or an OMVS segment (and corresponding UID) provide less exposure, but still might be used for running batch jobs.

⁵ A better approach would be to use the RACF *STARTED* class for this purpose, but for continuity reasons the AD developers have not done this (yet).

3.5.1 Group definitions

The following groups are defined. Names marked with an asterisk have userids connected to them. Many of the group names are defined in order to define a RACF data set profile for data sets whose HLQs match the group names.

ABD610	ADSM	ANF	API	APK	APO	ASM	ASMA
ASMT	ASU	BFS	BFT610	CBC	CDS	CEE	CEE150
CFS	CICDZN	CICS	CICSMPE	CICSTS13	CICTZN	CMN	CMN312
CMX	COG210	COK	COT	CPAC	CSF	DBS	DBSDZN
DBSTZN	DB2*	DB2PM	DCEGRP*	DGO	DIT	DLIB	ECN
EDC	EDF	EFM	EHW	EJS	EMPLOYEE*	EOY	EPH
EQAW	EUV	EWX	EXTERNAL*	EZM	FAN130	FFST	FME
FWGRP	GDDM	GIM	GLD	GLOBAL	GOT	GSK	HCM
HFS	HOU	HPJ	HPJ200	IBM	ICA	ICQ	IEL
IEL111	IGY	IGY210	IMO	IMW	IMWEB*	IOA	IOE
IPO1	ISF	ISP	JAVA	LOCAL	MQM	MQSER	MVS
MVSDZN	MVSNFS	MVSTZN	NETVIEW	NETV130	OMVSGRP*	OX390	PLI
PSF	P390M5	QMF	REXX	SCRIPT	SOMMVS	SPECIAL*	SYSCTLG*
SYSMVIEW	SYS1*	TCICDLB	TEST*	TIVOLI	TME10GEM	TTY	USER
USERCAT	VSAMDSET*						
CBADMGP*	CBAST1*	CBASR2*	CBCFG1*	CBNAMCR1	CBSYMC1	CBINTCR1	CBNAMSR1
CBACRU1	CBASRU1	CBACRU2	CBASRU2	CBLDAP	CBLG1P*	CBCTL1*	CBIVPGP*
CBIVPGP2*	CBLDAPGP*	CBRS1*	DFSGRP*	DSN710*			

We suggest you avoid using any of these group names, except SYS1, for your groups.

3.6 TSO logon procedures

Several TSO logon procedures are authorized for various users. As distributed, these are:

Procedure	UACC	Users.....
DBSPROC	NONE	P390,DSN1SPAS,SYSADM,SYSOPR,IBMUSER,P390A (more)
IKJACCNT	NONE	IBMUSER,P390,P390A,TEST1-6
ISPFLITE	READ	IBMUSER(UPDATE),P390(UPDATE)
ISPFPRCB	NONE	IBMUSER,P390
ISPFPROC	NONE	IBMUSER(A),P390,TESTER,P390A-Z (more)
OMVSPROC	NONE	IBMUSER(A),OPEN1-3,P390,TEST1-6

The (more) noted here refers to a list of userids that belonged to the system builders and that are no longer defined as userids. The purposes of these logon procedures are:

- ▶ ISPFPROC - A full-function procedure that allocates the data sets needed to use the ISPF panels for RACF, SDSF, SMPE, HCD, ISMF, and so forth. This should be considered the standard logon procedure (unless want to used an abbreviated, faster logon for your users who do not require access to administrative or advanced foreground functions).
- ▶ OMVSPROC - This procedure takes the user directly to the OMVS shell. It contains all the administrative allocations that are in the ISPFPROC procedure, but does not contain allocations for TSO foreground use of some of the compilers.
- ▶ DBSPROC - This is an expansion of the ISPFPROC procedure to include DB2 and QMF data sets. It is used if you have IPLed with parameters that include DB2 initialization functions. (The extra facilities are mentioned in "ISPF primary menu" on page 78.)
- ▶ IKJACCNT - This is a "bare bones" procedure that does not allocate any ISPF or administrative data sets. It is intended for emergency use only.
- ▶ ISPFLITE - This is discussed in "ISPFLITE procedure" on page 75.

- ▶ ISPFPRCB - We are not quite certain why this procedure is included. It allocates less than ISPFPROC and more than ISPFLLITE, and uses the catalog for all allocation.

3.7 IODFs and addresses

Three IODFs are included with the z/OS R1.1 AD system. These are SYS1.IODF00, SYS1.IODF01, and SYS1.IODF99. We refer to these as the 00, 01, and 99 IODFs. Your IPL parameter, via a value in the LOADxx member, selects which IODF is used for a given IPL. No “work” IODFs are included.⁶

As far as we could determine, the only differences between 00 and 01 were differences in esoteric device definitions that added the 3390s starting at address A80. Based on this, SYS1.IODF00 can be considered obsolete and you can ignore it. IODF99 contains more devices than IODF00 and some of the definitions are in error.

All the LOADxx members are set to use IODF99. We believe this is an error, and we (carefully) changed most of the LOADxx members to use IODF01. We did this by changing a few members (we used DC and DB first), IPLing with these members and verifying they worked, and then changing more members. *All the comments in this redbook assume you are using IODF01.*

The key difference between IODF99 and IODF01 is the definition of esoteric names DASD and SYSDA. If these do not match your system, you will have allocation errors that are essentially unrecoverable. These are the esoteric definitions:

- ▶ IODF99
 - DASD = 300-31F, 321-3AF
 - SYSDA = 300-31F, 321-3AF
- ▶ IODF01
 - DASD = 120-15F, 240-25F, 260-27F, A80-ABF
 - SYSDA = 120-15F, 240-25F, 260-27F, A80-ABF

We used addresses in the A80-ABF range for all our DASD. If we IPLed with any distributed LOADxx member (using IODF99), we were unable to allocate UNIT=SYSDA devices because we had no DASD at addresses in the 3xx range. For this reason, we changed all the LOADxx members in SYS1.IPLPARM to use IODF01.

The best solution, perhaps for a future AD system, would be to include all the addresses above in the definitions for DASD, SYSDA, and ALLSYSDA.

3.7.1 SYS1.IODF01

This IODF defines the following devices and addresses:

00C	2540R	Card reader. Useful as an emulated device
00E	1403-N1	Line printer. Useful as an emulated device
00F	3203	Line printer. Not clear why this is included.
010	3277	Terminal. Relic of system used to create AD releases
063	3277	Terminal. Relic of system used to create AD releases
120-15F	3380	Disks
240-25F	3380	Disks
260-27F	3390	Disks
560-57F	3480	No COMPACT option is defined on any drive
580-59F	3420-8	“Round” tapes. For AWSTAPE emulated drives.

⁶ There are two types of IODFs, production and work.

5A0-5AF	3422	“Round” tapes. For AWSTAPE or OMA emulation
700	3270	Terminal. AD systems use as NIP & OS/390 master
701-73F	3270-X	Terminal. Normally for VTAM (TSO, CICS, etc)
900-91F	3270-X	Terminal. Normally for VTAM (TSO, CICS, etc)
A80-ABF	3390	Disks. Can be for system volumes, etc
E20-E27	CTC	Typically for 3172s that appear as CTC devices
E40-E47	CTC	Typically for 3172s that appear as CTC devices

As far as we know, all the devices in this IODF are defined correctly. The most common change needed to is add the COMPACT feature to one (or more) 3480 definitions. The COMPACT feature is needed to process a compressed tape. For 3480 drives, this is known as an IDRC hardware feature and is enabled by selecting the COMPACT feature in the IODF definition. The COMPACT function is standard for 3490 drives.

You will probably use this IODF for your initial system operation. Many AD users never need to change this IODF and use it for all their system operations. See “Changing your I/O configuration” on page 18 for comments about matching your hardware to this IODF. However, you are not required to use the IODFs that are distributed with the AD system. If you have the necessary skills, you can create any IODF you require by using the HCD utility.

3.7.2 SYS1.IODF99

SYS1.IODF99 is also provided with the AD release, and is the IODF specified in all the distributed LOADxx members. It is intended to be an expanded version of IODF01, with more devices that might be needed for an MP3000 machine. Unfortunately, IODF99 contains a number of errors and problems. Moreover, there is no particular need to use it with an MP3000 unless you want device addresses that match particular documentation you may have.⁷ Member LOAD99, included in SYS1.IPLPARM but not documented in “Detailed IPL parameter controls” on page 57, points to IODF99.

Assuming you change all the LOADxx members (except LOAD99) to use IODF01, as recommended above, a way to then use IODF99 is as follows:

- ▶ IPL with parameter 0A8299T (using the correct address for the OS39M1 volume)
- ▶ The T parameter (seventh character in the IPL parameter field) causes the system to issue the SPECIFY SYSTEM PARAMETERS message. The reply should be:

```
REPLY 00,SYSP=CS
```

or any of the other two-character parameters that are possible suffixes for the IEASYSxx name, as listed in 3.4.1, “Detailed IPL parameter controls” on page 57.

This technique provides the same effect obtained from IPLPARM 0A82CS (in the sample here), but with a different starting point (LOAD99 instead of LOADCS). Another way to use IODF99 would be to change all the LOADxx members to refer to 99 instead of 01, or to create a whole new set of LOADxx members containing 99.

Again, we repeat that all the examples in this book were with a mixture of P/390 and MP3000 systems, all using IODF01. *There is no requirement to use IODF99 with MP3000 systems.*

The IODF99 definitions are quite similar to the IODF01 definitions. IODF99 has the following:

00C	2540R	Card reader. Useful as an emulated device.
00E-00F	1403-N1	Line printer. Useful as an emulated device.
120-15F	3380	Disks
1C0-1FF	3390	Disks. Not defined correctly (no OS defined in HCD).
240-25F	3380	Disks. No OS defined in HCD.

⁷ This is most common with the DaVinci package. Associated documentation assumes you are using addresses in the 300 range for your 3390 drives.

260-27F	3390	Disks. No OS defined in HCD.
300-3BF	3390	Disks.
500	3490	
550-55F	3420-8	“Round” tape. No OS defined.
560-56F	3480	No COMPACT option is defined on any drive.
570-57F	3480X	Has COMPACT feature
580-58F	3490	Tape. No OS defined.
590	3490	Tape.
591-59F	3490E	Tape. Invalid definition; no OS defined.
700	3270	Terminal. AD systems use as NIP & OS/390 master
701-73F	3270-X	Terminal. Normally for VTAM (TSO, CICS, etc)
900-91F	3270-X	Terminal. Normally for VTAM (TSO, CICS, etc)
A80-AEF	3390	Disks. Can be for system volumes, etc
E20-E23	CTC	Typically for 3172s that appear as CTC devices
E40-E43	CTC	Typically for 3172s that appear as CTC devices

3.8 Catalogs

The release AD system has six catalogs:

Catalog Name	Volume	Comments.....
CATALOG.OS390.MASTER	OS39M1	Master Catalog, named in LOADxx members
USERCAT.VZ1RES1	Z1RES1	Large number of aliases point to this one
USERCAT.VZ1DB21	Z1DB21	DB2, QMF, etc
USERCAT.VZ1CIC1	Z1CIC1	CICS
USERCAT.VZ1IMS1	Z1IMS1	IMS
USERCAT.VZ1WAS1	Z1WAS1	WebSphere, etc

The master catalog contains aliases pointing to the various user catalogs. The aliases associated with each user catalog are:

```

USERCAT.VZ1RES1
  AOP, ASU, CMX, ECN, EDF, EPH, EQAW, EUV, HOU, ICQ, IMW, IOA, BFS, CDS, CSF, EOY,
  EWX, CMN312, MVS, HLA, GDDM, GIM, GLD, GSK, ICA, IMO, IOE, ISP, SOMMVS, TCPIP, CEE,
  ISF, ASUITE, COT121, COG210, DIT130, HOU110, HJP200, IEL111, IGY210, NFS, NETV130,
  FAN130, JVA118, JVA130, CBC, CSQ520, TIVOLI
USERCAT.VZ1DB21
  ADB210, DG0710, DSN710, QMF710
USERCAT.VZ1CIC1
  CICSTS21
USERCAT.VZ1IMS1
  IMS710
USERCAT.VZ1WAS1
  BB0401

```

Any data set names whose High Level Qualifier (HLQ) does not match one of these aliases are cataloged in the master catalog. This includes all the data sets created by (or for) userids such as IBMUSER, P390, P390A, and so forth.

As distributed, the master catalog has no RACF protection. A RACF profile exists for USERCAT.* with Universal Access (UACC) set to *READ*. Userids DSN1DBM1 and DSN1MSTR have UPDATE access to this profile. (IBMUSER, P390, and any other userid with RACF SPECIAL and OPERATIONS authority can also alter the user catalogs.) This is opposite to the usual situations in which users have READ access to the master catalog and UPDATE access to at least one user catalog. However, it is not practical to change the

catalog security situation unless you (1) establish user catalogs (and corresponding aliases) for any local userids you may add, (2) stop using userids such as P390A that are already using the master catalog, and (3) add profiles permitting access to the master catalog for many of the started tasks used by the system.

3.9 Paging and spooling data sets

The paging and spool data sets for the distributed AD system are all on the OS39M1 volume. This is convenient for packaging and system management. However, it may not be appropriate for performance and growth. Once you have your AD system installed and operational, you can move and/or expand these data sets.

The paging data sets in the distributed system are:

Full Data Set Name	Volume	Space
SYS1.PLPA.PAGE.DATA	OS39M1	1500 tracks
SYS1.COMMON.PAGE.DATA	OS39M1	750 tracks
SYS1.LOCAL.PAGE.DATA	OS39M1	6000 tracks

The PLPA page data set is about 82% full (after IPLing with the CS parameter). The 6000 tracks of user paging space amounts to about 288 MB of virtual memory that requires disk backing. This is sufficient for reasonable numbers of TSO users and batch jobs. It may not be sufficient for combinations of WebSphere, Java, CICS, DB2, and so forth. If additional paging data sets are needed, they should be allocated on user volumes for better performance.

The allocated JES2 spool space is 2340 tracks (156 cylinders). This represents over a million lines of spooled output. It is certainly sufficient for getting started, but most traditional MVS installations can consume any amount of spool space. MP3000 installations may typically allocate a complete 3390-3 volume for spool space.

As a side note, the allocated space for SMF data sets is:

Full Data Set Name	Volume	Space
SYS1.MAN1.DATA	OS39M1	600 tracks
SYS1.MAN2.DATA	OS39M1	150 tracks
SYS1.MAN3.DATA	OS39M1	150 tracks

By default, SMF recording is turned off, so none of this space is needed. The sizes allocated are reasonable for typical SMF recording. If you enable SMF recording, be certain to have jobs ready to empty the SMF data sets when needed.

3.10 UNIX System Services file structure

The UNIX System Services data sets and mount points are organized differently than in earlier AD releases. The general intention is to provide a new level of indirection in file system mounts. This should make it easier to apply maintenance, install and test new releases, and so forth. These changes were introduced in the base ServerPac, and are not unique to the AD system.

The mount points specified in BPXPRMxx use system symbols:

&SYSNAME	=	P390
&SYSPLEX	=	ADCDPL
&UNIXVER	=	V1ROM0

You can display the current system symbols with the MVS command **D SYMBOLS**. In addition to these, another symbol is defined in BPXPRMxx:

```
VERSION('&UNIXVER.') = V1R1M0
```

Using these symbols, the file system mounts defined in BPXPRMxx include:⁸

Full Data Set Name	Volume	Mount point	Actual mount point	Tracks
HFS.ADCDPL.ROOT	OS39HA	-----	/	15
HFS.P390.DEV	OS39HA	/&SYSNAME./dev	/P390/dev	90
HFS.P390.ETC	OS39HA	/&SYSNAME./etc	/P390/etc	320
HFS.P390.HFS	OS39HA	/&SYSNAME	/P390	15
HFS.P390.TMP	OS39HA	/&SYSNAME./tmp	/P390/tmp	629
HFS.P390.VAR	OS39HA	/&SYSNAME./var	/P390/var	30
HFS.USERS	OS39HA	-----	/u	2
HFS.USR.MAIL	OS39HA	-----	/usr/mail	75
HFS.V1R1M0.VERSION.HFS	OS39HA	/\$VERSION	/V1R1M0	16545
HFS.WEB	OS39HA	-----	/web	450

Note the size of /u. It is intended only for a mount point for your /u directories—which should be on your volumes, not the system volumes.⁹ While not shown here, you should be aware that specific product volumes, such as Z1DB21 (for DB2), contain additional HFS data sets (and each is a file system) for that product. Notice that some traditional “first level” directories, such as /etc, are now “second level” directories, such as /P390/etc.

The root directory contains links that make /P390/etc, for example, look like the more traditional /etc. Output from the command **ls -a1** (in the root directory) includes:

```
drwxr-xr-x  6 BPXOINIT SYS1  8192 Dec  1 14:01 P390
drwxr-xr-x 21 BPXOINIT SYS1  8192 Oct 20 17:01 V1R1M0
lrwxrwxrwx  1 BPXOINIT SYS1    12 Jun  1 17:01 bin -> $VERSION/bin
lrwxrwxrwx  1 BPXOINIT SYS1    12 Jun  1 17:01 dev -> $SYSNAME/dev
lrwxrwxrwx  1 BPXOINIT SYS1    12 Jun  1 17:01 etc -> $SYSNAME/etc
lrwxrwxrwx  1 BPXOINIT SYS1    12 Jun  1 17:01 lib -> $VERSION/lib
lrwxrwxrwx  1 BPXOINIT SYS1    12 Jun  1 17:01 opt -> $VERSION/opt
lrwxrwxrwx  1 BPXOINIT SYS1    16 Jun  1 17:01 samples -> $VERSION/samples
lrwxrwxrwx  1 BPXOINIT SYS1    12 Jun  1 17:01 usr -> $VERSION/usr
lrwxrwxrwx  1 BPXOINIT SYS1    12 Jun  1 17:01 var -> $SYSNAME/var
lrwxrwxrwx  1 BPXOINIT SYS1    12 Jun  1 17:01 tmp -> $SYSNAME/tmp
drwxr-xr-x  8 BPXOINIT SYS1  8192 Jun  4 15:02 u
drwxr-xr-x  1 BPXOINIT SYS1  8192 Jun  6 17:05 web
```

Notice that all the traditional names, such as /bin, resolve to directories in either P390 or V2R1M0. In principle, the contents of P390 (and all its subdirectories) are locally customized and perhaps release independent, while V1R1M0 contains code and other files that are release dependent.

When first seen, this arrangement appears complicated. For practical purposes, it is seen only when listing the root directory. Routine functions and commands, such as **cd /etc**, work as expected—thanks to the links in the root directory. The long-term purpose of this arrangement is to make software maintenance and migration easier. In the meantime, it requires second glances when directly working in the root directory.

⁸ There are additional mounts not mentioned here, for NetView, CICS, and so forth. These are straightforward and their inclusion here would just complicate the discussion.

⁹ Alternately, you could define your own /u file system (not on a system volume) and replace (in BPXPRMxx) the **MOUNT** command for /u.

3.11 HFS mounts in BPXPRMxx

The following table indicates which BPXPRMxx members (BB, CI, 00, DB, CS) mount which HFS data sets:

BPXPRMxx:						Mount point.....
BB	CI	00	DB	CS	HFS.....	
X	X	X	X	X	HFS.ADCDPL.ROOT	/
X	X	X	X	X	HFS.P390.HFS	/P390
X	X	X	X	X	HFS.V1R1M0.VERSION.HFS	/V1R1M0
X	X	X	X	X	HFS.P390.DEV	/P390/dev
X	X	X	X	X	HFS.P390.VAR	/P390/var
X	X	X	X	X	HFS.P390.ETC	/P390/etc
X	X	X	X	X	HFS.P390.TMP	/P390/tmp
X	X	X	X	X	HFS.USR.MAIL	/P390/mail
X	X	X	X	X	HFS.USERS	/u
X	X	X	X	X	NETV130.OMVS.USR.LPP.NETVIEW	/usr/lpp/netview
X	X	X	X	X	HPJ200.HFS	/usr/lpp/hpj
	X	X	X	X	COT121.SCOFHFS	/usr/lpp/IBM_Payment_Server
	X	X	X	X	COG210.SCOGHFS	/usr/lpp/PaymentGateway
	X	X	X	X	CMN312.SCMNHFS	/usr/lpp/NetCommerce
X		X	X	X	JVA130.HFS	/usr/lpp/java
X	*	*			DSN710.SDSNHFS	/usr/lpp/db2/db2710
X	*	X			DSN710.SDMBHFS	/usr/lpp/db2ext_07_01_00
X	*	*			DSN710.SDESHFS	/usr/lpp/db2tx/db2710
		X	X		DSN710.SDTWHFS	/usr/lpp/netdata
X	X	X	X		HFS.WEB	/web
X	X	X			HFS.WAS	/was
	X	X			CICSTS21.OMVS.USR.LPP.CICSTS	/usr/lpp/cicsts
	X	X			CICSTS21.OMVS.USR.LPP.CICSTS.CICSTS21	/usr/lpp/cicsts/cicsts21
X					BB0401.SBBOHFS	/usr/lpp/WebSphere
X					BB0401.SBBOHFS2	/usr/lpp/WebSphere390

The “*” points in this list indicate errors. See 3.14.13, “DB2 mounts for HFS data sets” on page 74 for corrections. HFS.WAS does not exist and there will be error messages when BPXPRMCI or BPXPRMDB are used; these can be ignored. (BPXPRMCI is not used by any of the startup parameters provided with this AD system.)

In most z/OS usage (and in the AD system) a “00” member in PARMLIB indicates a minimal configuration for whatever member is being used. This is not the case for BPXPRM00; in this case, BPXPRMCS is a minimal member.

3.12 SMP/E data sets

The AD system contains a full SMP/E environment for all the components shipped with it. For the base system components, the SMP/E data sets are on the DLIB volumes. For DB2, CICS, IMS, and various WebSphere components, the SMP/E data sets are on their separate volumes. In this section we will ignore these SMP/E data sets and discuss only those contained on the DLIB volumes.

The easiest way to start investigating the SMP/E structure is to use the ISPF 3.4 command to list the names of data sets with the same **.CSI; that is with any data set name ending in a qualifier of CSI. All SMP/E consolidated inventory data sets have this type of name. It is possible that a non-SMP/E data set may have such a name, but none are present in the distributed AD system.

The CSIs present on the DLIB volumes are:

Data set name	Volume	Product or Comment
MVS.GLOBAL.CSI	Z1DIS2	Primary starting point for most SMP/E work
MVS.MVSD.CSI	Z1DIS2	CSI for Base system DLIBs
MVS.MVST.CSI	Z1DIS2	CSI for operational libraries
CEE150.GLOBAL.CSI	Z1DIS2	Language Environment
CEE150.CEE150D.CSI	Z1DIS2	
CEE150.CEE150T.CSI	Z1DIS2	
CMN312.GLOBAL.CSI	---	On both OS39DA and OS39PA (?)
CMN312.CMN312D.CSI		Net.Commerce
CMN312.CMN312T.CSI	Z1DIS1	(Target CSI on Z1DIS1 only)
CSQ520.CSQ520D.CSI	Z1DIS1	MQSeries
CSQ520.CSQ520T.CSI	Z1DIS1	Uses the MVS Global CSI
DIT130.GLOBAL.CSI	Z1DIS2	Ditto
DIT130.DIT130D.CSI	Z1DIS2	
DIT130.DIT130T.CSI	Z1DIS2	
FAN130.GLOBAL.CSI	Z1DIS2	REXX Compiler
FAN130.FAN130D.CSI	Z1DIS2	
FAN130.FAN130T.CSI	Z1DIS2	
HOU110.GLOBAL.CSI	Z1DIS2	Millenium Extensions
HOU110.HOU110D.CSI	Z1DIS2	
HOU110.HOU110T.CSI	Z1DIS2	
HPJ200.GLOBAL.CSI	Z1DIS1	Visual Age for Java
HPJ200.HPJ200D.CSI	Z1DIS1	
HPJ200.HPJ200T.CSI	Z1DIS1	
IEL111.GLOBAL.CSI	Z1DIS2	PL/I Compiler and libraries
IEL111.IEL111D.CSI	Z1DIS2	
IEL111.IEL111T.CSI	Z1DIS2	
IGY210.GLOBAL.CSI	Z1DIS2	COBOL Compiler and libraries
IGY210.IGY210D.CSI	Z1DIS2	
IGY210.IGY210T.CSI	Z1DIS2	
JAV118.GLOBAL.CSI	Z1DIS1	
JVA118.JVA118D.CSI	Z1DIS1	
JVA118.JVA118T.CSI	Z1DIS1	
JAV130.GLOBAL.CSI	Z1DIS1	
JVA130.JVA130D.CSI	Z1DIS1	
JVA130.JVA130T.CSI	Z1DIS1	
NETV130.GLOBAL.CSI	Z1DIS2	
NETV130.DLIB1.CSI	Z1DIS2	
NETV130.TGT1.CSI	Z1DIS2	

When using the ISPF panels for SMP/E, the first panel (Primary Options Menu) should usually be set to one of the global CSI data sets, such as:

```
SMPCSI DATA SET ==> 'MVS.GLOBAL.CSI'
```

The various SMP/E holding data sets are named in an orderly way and can be found using the ISPF 3.4 function with **** .SMP*** as the Dsname Level. Note that some of the SMPPTS and SMPLTS data sets are quite large.

3.13 DFSMS state

HFS volumes are no longer required to be System Managed volumes. None of the volumes in the z/OS R1.1 AD system are managed by DFSMS. Most users of the AD system will probably ignore DFSMS.

Two storage groups (DBCLASS and HFSCCLASS) are defined, and two storage classes are defined (same names). Two volsers are known to DFSMS, OS3P8A (group DBCLASS) and OS3H8A (group HFSCCLASS). These volsers are clearly left over from an earlier AD release.

The use of HFS volumes requires that DFSMS be active, even though HFS volumes no longer need to be defined as System Managed volumes. It is active using the standard data sets (**D SMS** command):

```
SCDS = SYS1.SCDS
ACDS = SYS1.ACDS
COMMDS = SYS1.COMMDS
SYSTEM NAME = P390
```

3.14 Minor fixes

As with any complex system, implemented under time pressures and intended for use under widely varying conditions, there are a number of minor adjustments that may be necessary in specific situations. Several widely-applicable adjustments are described in this section.

3.14.1 Different IODF

The default IODF (IODF99) may not be well chosen, depending on your DASD addresses. See “IODFs and addresses” on page 63 for details.

3.14.2 Timer settings

The AD system expects to find an ETR timer; this is typically associated with a Parallel Sysplex environment. If you do not have such an environment, you must reply to an IPL message to accept the system time. To bypass this message, edit PARMLIB member CLOCK00 as follows:

```
OPERATOR NOPROMPT
TIMEZONE W.05.00.00      (change from 0.00.00.00)
ETRMODE NO               (change from YES)
ETRZONE NO               (change from YES)
ETRDELTA 10              (not relevant)
```

Remove the SIMETRID line.

3.14.3 Remove automatic ASCH and APPC startups

By default, the AD system starts a number of APPC and ASCH address spaces. Very few AD system users need these. You should know if you need these; otherwise it is probably safe to assume that you do not need them. They are started by the various VTAMAPPL command list members in PARMLIB. In the z/OS R1.1 system, there are six different VTAMAPPL command lists that might be used, depending on the IPL parameter you selected. You can study “Detailed IPL parameter controls” on page 57 to determine which VTAMAPPL command list members you need to change, or you can simply change all six. The six members (in SYS1.ADCD01.PARMLIB) are VTAMALL, VTAMAPPL, VTAMBB, VTAMCI, VTAMDB, and VTAMD2. The same change is made to each member. The change *comments out* two lines, as follows:

```
/* S APPC,SUB=MSTR,APPC=1A */
/* S ASCH,SUB=MSTR,APPC=1A */
```

Your change is to place the */** and **/* around these two lines in the PARMLIB members. Alternatively, you could simply delete the lines; however, normal MVS practice is to leave the *commented out* lines as a form of documentation.¹⁰

¹⁰ Unfortunately, not all PARMLIB members accept *commented out* lines; you should examine existing members for examples of where this is possible.

3.14.4 Too many initiators

By default, the AD system starts ten initiators. This is probably too many for most AD users, especially those with P/390-based machines. You can change this by editing member JES2PARM in PARMLIB. Before doing this, you should copy member JES2PARM to new member JES2BACK. There is a simple method for starting JES2 using the parameters in JES2BACK instead of the parameters in JES2PARM. This is important if someone makes a serious error in JES2PARM such that JES2 cannot start.

After making the backup copy of JES2PARM, edit JES2PARM and find the following lines:

```
INIT(1) NAME=1,          /* Comments ..... */
      CLASS=A,          /* More comments */
      START=YES        /* Yet more comments... */
                          /*
INIT(1) NAME=2,          /* Comments ..... */
      CLASS=AB,         /* More comments */
      START=YES        /* Yet more comments... */
                          /*
INIT(1) NAME=3,          /* Comments ..... */
      CLASS=ABC,        /* More comments */
      START=YES        /* Yet more comments... */
                          /*
.....
```

Change some of the START=YES parameters to START=NO. We have found that four initiators is reasonable for a P/390-based machine, but this must be adjusted for local conditions.

3.14.5 Web server does not start

In earlier releases the Web server used the started task name IMWEBSRV. This has been changed to HTTPD1. The name IMWEBSRV is still used in various RACF controls and other places.

If you IPL with the CS or 00 parameters (that is, with IPLPARM=0A82CS or 0A8200, for example), you will find that /web is just an empty mount point.¹¹ The BPXPRMCS member in PARMLIB, which is used by the CS and 00 IPLPARM, does not mount the HFS data set needed by the Web server. If you want to use the Web server with these IPLPARMs, edit member BPXPRMCS in PARMLIB and add the following lines somewhere in the series of **mount** commands:

```
MOUNT FILESYSTEM('HFS.WEB')      <=== add these lines
      TYPE(HFS)                   <=== add these lines
      MODE(RDWR)                  <=== add these lines
      MOUNTPoint('/web')          <=== add these lines
```

3.14.6 /u has incorrect permissions and is very small

The directory entry (in Unix System Services) for /u has permissions set to 700. This permits full access for users with UID=0, but prevents access to any files mounted under /u for all other users. This is probably not what you want. To correct this, log on with a userid id that has UID=0 (use TSO followed by OMVS or ISH, or telnet directly to OMVS) and issue the command:

```
# chmod 755 /u
```

¹¹ The VTAMAPPL script, used by the CS and 00 IPL parameters, contains a S HTTPD1 command even though the required HFS files are not mounted.

Also, note that the allocated space for the /u file system is very small—only two tracks. You can create files in /u, but this will force secondary allocations for HFS.USERS. The secondary allocations are also very small, so this is not a reasonable approach for any significant use of /u. The intention is to force you to allocate other file systems (not on system packs) and mount these under /u, or to create your own /u file system. You will appreciate this when it is time to move to the next system release.

3.14.7 IPCS has nonexistent dumps in inventory

If you use IPCS (with the userid P390, which is the userid most AD system administrators use) you will notice a number of dumps in the inventory. Some of these dumps do not exist on the disks and the rest are irrelevant; they can be deleted from the inventory. To do this, go to the ISPF panels for IPCS, then to the inventory display. Enter DD for each inventory item to be deleted. In the next panel, enter

```
RECORDS = ALL
DELETE = YES                                (will create error; ignore it)
```

to delete the inventory entries.

3.14.8 Unexpected RACF parameters

The RACF configuration in the AD release 10 system may not be quite what would be expected for a typical, smaller OS/390 system. In particular:

- ▶ The *List of Groups* function is active. This is good, although it is a change from previous AD releases.

List of Groups processing has a logical, but interesting effect on access permissions for UNIX System Services files. A RACF group may have a GID value. If a user is a member of several groups that have different GIDs (and if the List of Groups function is active) then the user has several active GID values. All of these will be checked when needed to satisfy the group permissions for a UNIX file.

- ▶ The Enhanced Generic Naming (EGN) function of RACF is off. It has been off in all the recent AD releases. The situation is not as clear as it is with List of Groups. Some customers also have EGN off. With EGN off, the RACF data set profile name XYZ.* includes all data sets whose HLQ is XYZ. With EGN on, the profile name XYZ.** is needed to do the same thing. There is more to it than this, of course, but the differences have to do with fine-tuning data set profile names.

You can turn EGN on with the command **SETROPTS EGN**. This will automatically convert the existing data set profile names to the EGN format. This is usually considered to be a one-way conversion; you would not normally convert back to EGN off, although it can be done in some cases. As far as we can tell, most AD system users leave EGN off. Most larger OS/390 customers turn it on.

- ▶ Common classes are not RACLISTed. The TSOAUTH class, for example, is normally RACLISTed, but this has not been done in this AD release. The RACLIST function provides a small performance enhancement; it makes no other significant difference. Many examples in this book assume that TSOAUTH (and several other classes) are RACLISTed and include **SETROPTS RACLIST(TSOAUTH) REFRESH** commands after other RACF commands that alter profiles in the class. This refresh is not necessary (but does no harm) if the class is not RACLISTed. We include the refresh in the examples because almost all larger OS/390 installations will have these classes RACLISTed.

The command **SETROPTS RACLIST(TSOAUTH)**, for example, will RACLIST class TSOAUTH.

On a small AD system the performance gain from RACLISTing is insignificant; the author generally leaves his systems as they are in the current AD package.

- ▶ The STARTED class is enabled, but is used only for DCE, Web server, and a few UNIX System Services functions. It is not used for the normal OS/390 started tasks. These are handled by having user profiles that match the started task name. These are readily seen in “User IDs” on page 59.
- ▶ There are no Global Access Checking (GAC) definitions. GAC provides a fast path through RACF checking for selected profiles and is almost always used in larger installations. Its only effect is performance related, and the effects on a small AD system would probably not be significant. See the RACF documentation for more information about enabling GAC.
- ▶ No default OMVS userid is defined.

3.14.9 Region size for SMP/E

SMP/E jobs, submitted from the ISPF SMP/E panels, sometimes abend due to insufficient storage. You can change the default region sizes for batch jobs (in JES2PARMS) or simply add a REGION=40M parameter to the SMP/E job cards that are generated by the panels. (You do not really need a 40 MB region; this number should be taken as a generic “large” region size.)

3.14.10 Web server requires RACF login

The Web server (started as HTTPD1) requires a RACF userid/password to use it. That is, when a browser is pointed to it, the browser user is immediately prompted for a userid and password. This is probably not what you want. Be certain you have the /web file system mounted (see “Web server does not start” on page 71), go to OMVS, and edit /web/httpd1/httpd.conf. Find the line (approximately line number 200) that says

```
UserId %%CLIENT%%
```

and change it to

```
UserId PUBLIC
```

This will cause Web users to be associated with RACF userid PUBLIC, which is defined and set up for this purpose, with appropriately limited privileges.

3.14.11 WebSphere not customized

WebSphere components are distributed on CDs 11 and 12 (volumes Z1WAS1 and Z1WAS2). However, it is not “ready to run.” This has been resolved since the CDs were shipped and information is available on the Web. See “Web support” on page 4.

3.14.12 OMVS not trusted

The system HFS data sets have the HLQ of “HFS”. There is a RACF data set profile for HFS.* with UACC set to *READ*. OMVS, in order to provide file services for UNIX System Services users, needs UPDATE or ALTER access to the HFS data sets. In most OS/390 installations this is provided by using a STARTED class profile for OMVS, and by specifying the TRUSTED attribute in this profile. A started task with the TRUSTED attribute bypasses RACF access checks.¹²

¹² Note that OMVS itself (the kernel) would bypass checks. This is not extended to users of OMVS.

The AD system does not use a STARTED class profile for OMVS and has no way to assign the trusted attribute. Instead, OMVS is permitted (with a RACF **PERMIT** command) to the HFS.* profiles, with *UPDATE* access.

If you create new HFS data sets (each being a new file system), you have several choices:

- ▶ You can name your new data set with the HLQ of HFS. It will then be protected by the existing HFS.* generic profile, and OMVS can already access these data sets.
- ▶ You can name your new HFS with a different name and not RACF-protect the new HFS. This is not a good idea.
- ▶ You can name your new HFS anything and provide a RACF profile to protect it. You then need to PERMIT OMVS to this profile, with UPDATE access.

3.14.13 DB2 mounts for HFS data sets

Two BPXPRMxx members have incorrect **MOUNT** commands (incorrect mount points) for the HFS data sets used by DB2. BPXPRM00 has three errors and BPXPRMDB has two errors. The correct **MOUNT** commands are:

```
MOUNT FILESYSTEM('DSN710.SDSNHFS')
      TYPE(HFS)
      MODE(RDWR)
      MOUNTPOINT('/usr/lpp/db2/db2710')           <=== verify mount point

MOUNT FILESYSTEM('DSN710.SDMBHFS')
      TYPE(HFS)
      MODE(RDWR)
      MOUNTPOINT('/usr/lpp/db2ext_07_01_00')     <=== verify mount point

MOUNT FILESYSTEM('DSN710.SDESHFS')
      TYPE(HFS)
      MODE(RDWR)
      MOUNTPOINT('/usr/lpp/db2tx/db2710')       <=== verify mount point
```

3.14.14 Latest P/390 device managers

You should obtain the latest P/390 (or R/390, or Integrated Server) updates from the p390.ibm.com ftp site before running z/OS. This is discussed in slightly more detail in “P/390 fixpak” on page 4.

3.14.15 Error in inetd procedure

The inetd program is usually started from /etc/rc. If necessary, it can be started from SYS1.PROCLIB, as a started task. The INETD member in PROCLIB uses BPXBATCH to start inetd the same way it would be started in /etc/rc. The PARM field in the INETD member contains two errors. It should be PARM='PGM /usr/sbin/inetd /etc/inetd.conf'. The distributed version has *usr* spelled as *use* and is missing the slash before *etc*.

3.14.16 VTAMLST errors and additions

The A0600 member of SYS1.LOCAL.VTAMLST contains minor errors. Note that the AD system builders elected to create and concatenate SYS1.LOCAL.VTAMLST before SYS1.VTAMLST in the VTAM started task procedure. This LOCAL data set is not a traditional part of MVS, and you may not find it elsewhere. SYS1.VTAMLST is the assumed location for VTAM control parameters; you must be aware of the concatenation used in the AD system in order to correct the correct library.

Also, these two VTAMLST libraries contain some duplicate member names. Be certain to update the correct member.

The next-to-last line in A0600 is missing a comma and a continuation character in column 72. This creates console error messages when VTAM starts. You should change line:

```
VPACING=1                                0212011
```

to

```
VPACING=1,                                C0212011
```

If you are using the ISPF editor, you will not see the sequence numbers in the last eight columns unless you scroll sideways.

The same member, A0600, contains an entry for each possible concurrent TSO session. There are only 10 members defined for this, limiting your system to 10 concurrent TSO sessions. You probably want to increase this. Find the following three lines:

```
*
A06TS0010 APPL AUTH=(NOACQ,NOBLOCK,PASS,NOTCAM,NVPACE,TSO,NOPO),      X
          EAS=1,ACBNAME=TS00010
```

Replicate these three lines to make as many definitions as you want for concurrent TSO use. Then change two fields in each set of lines. For example, we wanted to have a maximum of 40 concurrent TSO users, so we replicated 30 copies of these lines (after the 10 copies that already exist). We then update each set to continue the same pattern of numbering. For example, we added:

```
*
A06TS0011 APPL AUTH=(NOACQ,NOBLOCK,PASS,NOTCAM,NVPACE,TSO,NOPO),      X
          EAS=1,ACBNAME=TS00011
*
A06TS0012 APPL AUTH=(NOACQ,NOBLOCK,PASS,NOTCAM,NVPACE,TSO,NOPO),      X
          EAS=1,ACBNAME=TS00012
*
A06TS0013 APPL AUTH=(NOACQ,NOBLOCK,PASS,NOTCAM,NVPACE,TSO,NOPO),      X
          EAS=1,ACBNAME=TS00013
.....
```

3.14.17 ISPF LITE procedure

The ISPF LITE TSO logon procedure is intended to provide a light-weight, fast TSO logon procedure, as contrasted with the ISPF PROC full-function logon procedure. ISPF PROC allocates ISPF libraries for RACF, ISMF, HCF, SMP/E, and other system programmer tools not needed by normal users. We found that ISPF LITE was a little too light, because it did not allocate the libraries needed to use SDSF under ISPF. We added the following procedure as SYS1.TSOPROC(BILLPROC)¹³ and authorized any TSO user to use it:

```
/******
//*
//*          BILLPROC FOR z/OS
//*
//*****
//ISPFPROC EXEC PGM=IKJEFT01,REGION=0M,DYNAMNBR=175,
//          PARM='%ISPFCL'
//SYSLBC DD DISP=SHR,DSN=SYS1.BROADCAST,UNIT=3390,VOL=SER=OS39M1
//SYSPROC DD DISP=SHR,DSN=SYS1.LOCAL.CLIST,UNIT=3390,VOL=SER=Z1RES1
```

¹³ There is nothing special about the name BILLPROC, of course. Whatever name you use should be specified in the corresponding RACF command. An alternative is to update the ISPF LITE procedure and not create a new procedure.

```

//      DD DISP=SHR,DSN=ISP.SISPCLIB,UNIT=3390,VOL=SER=Z1RES1
//      DD DISP=SHR,DSN=SYS1.CLIST,UNIT=3390,VOL=SER=OS39M1
//      DD DISP=SHR,DSN=SYS1.SBPXEXEC,UNIT=3390,VOL=SER=Z1RES1
//SYSEXEC DD DISP=SHR,DSN=ISP.SISPEXEC,UNIT=3390,VOL=SER=Z1RES1
//      DD DISP=SHR,DSN=SYS1.SBPXEXEC,UNIT=3390,VOL=SER=Z1RES1
//SYSHLP DD DISP=SHR,DSN=SYS1.HELP,UNIT=3390,VOL=SER=Z1RES1
//      DD DISP=SHR,DSN=ISP.SISPHELP,UNIT=3390,VOL=SER=Z1RES1
//ISPMLIB DD DISP=SHR,DSN=ISP.SISPMENU,UNIT=3390,VOL=SER=Z1RES1
//      DD DISP=SHR,DSN=SYS1.SBPXMENU,UNIT=3390,VOL=SER=Z1RES1
//      DD DISP=SHR,DSN=ISF.SISFMLIB,UNIT=3390,VOL=SER=Z1RES2
//ISPEXEC DD DISP=SHR,DSN=ISP.SISPEXEC,UNIT=3390,VOL=SER=Z1RES1
//      DD DISP=SHR,DSN=SYS1.SBPXEXEC,UNIT=3390,VOL=SER=Z1RES1
//ISPPLIB DD DISP=SHR,DSN=SYS1.LOCAL.ISPFPNLS,UNIT=3390,
//      VOL=SER=Z1RES1
//      DD DISP=SHR,DSN=ISP.SISPPENU,UNIT=3390,VOL=SER=Z1RES1
//      DD DISP=SHR,DSN=SYS1.SBPXPENU,UNIT=3390,VOL=SER=Z1RES1
//      DD DISP=SHR,DSN=ISF.SISFPLIB,UNIT=3390,VOL=SER=Z1RES2
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPLIB,UNIT=3390,VOL=SER=Z1RES1
//      DD DISP=SHR,DSN=ISP.SISPSENU,UNIT=3390,VOL=SER=Z1RES1
//      DD DISP=SHR,DSN=ISF.SISFSLIB,UNIT=3390,VOL=SER=Z1RES2
//ISPPLIB DD DISP=SHR,DSN=ISP.SISPPTENU,UNIT=3390,VOL=SER=Z1RES1
//      DD DISP=SHR,DSN=SYS1.SBPXTENU,UNIT=3390,VOL=SER=Z1RES1
//      DD DISP=SHR,DSN=ISF.SISFTLIB,UNIT=3390,VOL=SER=Z1RES2
//ISPCLT1 DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//      DCB=(LRECL=80,BLKSIZE=800,RECFM=FB)
//ISPCLT2 DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//      DCB=(LRECL=80,BLKSIZE=800,RECFM=FB)
//ISPLST1 DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//      DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA)
//ISPLST2 DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
//      DCB=(LRECL=121,BLKSIZE=1210,RECFM=FBA)
//SDSFMENU DD DSN=ISF.SISFPLIB,DISP=SHR

```

After placing this procedure in SYS1.TSOPROC, the following TSO RACF commands will permit all TSO users to use this logon procedure:

```

RDEF TSOPROC BILLPROC UACC(READ)
SETROPTS RACLIST(TSOPROC) REFRESH

```

To use BILLPROC, a user would simply specify this name in the procedure field of the TSO logon password screen.

3.14.18 Shutdown scripts

The SHUTDB script (in PARMLIB) contains an error. It issues a **C IMWEBSRV** command which should be **P HTTPD1**.

The SHUTALL script issues **F CICS,CEMT P SHUT I**. This has two errors: CICS runs under the name CICSA. Also, the default CICS with the AD system does not have the MVS console interface fully installed. CICS can be stopped with the command **C CICS**. (It might be better to log onto CICS and use the CEMT transaction to shutdown CICS, but the simple cancel is effective.)

All three scripts (SHUTDOWN, SHUTDB, SHUTALL) contain the command **C NSFC**, which, as far as we can tell, is an error. Other than creating an error message, it does no harm.

3.14.19 UNIX command lp

The UNIX System Services command:

```
lp -d,x my.text.file
```

should spool the indicated file to JES2 (class X), where it could be inspected with SDSF and/or printed. The current system spools the file correctly, but when the user (who issued the lp command) attempts to look at the file with SDSF, he receives a *Not Authorized* response. The spooled file is in the JES2 *Hold* queue. An administrator with OPER and ACCT authority can view the spooled file.

3.15 Special IPL replies

You may see some of the following messages during IPL -- especially during your first IPL:

```
IXL011I XES HARDWARE SUPPORT IS NOT INSTALLED. REASON 01
IXC414I CANNOT JOIN SYSPLEX PLEX1 WHICH IS RUNNING IN MONOPLEX MODE:
        CONFIGURATION REQUIREMENT
IXC404I SYSTEM(1) ACTIVE OR IPLING: P390
IXC420D REPLY I TO INITIALIZE SYSPLEX PLEX1, OR R TO REINITIALIZE XCF
```

You should reply **REPLY 00,I** if you receive this message.

After an IPL is finished, you will have the following outstanding message on the MVS console. (After this message works its way to the top of the screen, you can issue the MVS operator command **K E,1** to remove it from the screen.)

```
*03 ISTEXC200 - DYN COMMANDS MAY BE ENTERED
```

The following commands may be entered in reply to the outstanding WTOR (message ISTEXC200) issued by the Dynamic XID Exit. Note that the command will not be processed and another WTOR will not be issued until next time that the exit is called by VTAM.

```
R XX,RELOAD
    (The current Dynamic XID mapping table is reloaded.)

R XX,DYNTAB=<table>
    (The Dynamic XID mapping table <table> becomes the current table.)
```

For further information see SYS1.README(VTAM). Very few people use this message function. When the DYN COMMANDS MAY BE ENTERED message scrolls to the top of the screen, you can remove it with a **K E,1** command.

3.16 Shutting down the system

The system has three shutdown scripts. The proper script should be matched to the IPL parameter you used, as follows:

SCRIPT	IPL Parameter
SHUTDOWN	CS, 00
SHUTDB	DC, DB, CC, CW, 2C, 2W, IC, IW, DI, BB, BC
SHUTALL	AC, AW

In general, it will not hurt to use a more "powerful" shutdown script (such as SHUTALL) with a less "powerful" IPL parameter. The script will issue various stop and cancel commands for programs that might not be running, but (aside from an error message) this will do no harm.

To stop your system, you can issue **S SHUTDOWN** (or **S SHUTDB**, or **S SHUTALL**) from the Master Console. This may, or may not, produce a clean shutdown. You can edit the script in `SYS1.ADCD10.PARMLIB(SHUTDOWN)`. You may want to, for example, reduce the 30 second delay after the message is sent to TSO users.

The shutdown script takes several minutes to execute. When it is finished you should be able to issue a **\$PJES2** command to stop JES2. A **Z EOD** command would complete the shutdown (although most people do not bother). A **QUIESCE** command, although not required, stops all OS activity.

3.17 Initial MVS console PFK assignments

The MVS console PFK assignments distributed with the AD CD-ROM are:

PFK	COMMAND.....	ACTION.....
1	D U,DASD,,000,999	Display DASD volumes (online and offline)
2	D U,,660,12	Display units at addresses 660 - 611
3	K	Clear main part of display ("top area")
4	K A,10	Create a 10-line Display Area ("bottom area")
5	D A,ALL	Display Address Space Information
6	D R,L	Display Outstanding Replies/Errors
7	D PFK,L=A	Display PF Keys
8	K D,F	Scroll Display Area ("bottom area")
9	D TS,L	Display TSO Users
10	D A,L	Display Active Address Spaces
11	D J,L	Display Active Jobs
12	K E,D	Remove Display Area ("bottom area")

These commands may not be completely appropriate for the AD CD-ROM. You can change the commands by editing `SYS1.PARMLIB(PFKTAB00)`. PFKs 13-24 simply repeat the actions of keys 1-12.

3.18 ISPF primary menu

For the basic logon procedure (ISPFPROC) the second panel of the primary ISPF menu, accessed through the **M (More)** option on the first panel, contains:

1	SMP/E	System Modification Program/Extended
2	ISMF	Integrated Storage Management Facility
3	RACF	Resource Access Control Facility
4	HCD	Hardware Configuration Dialogs
5	SDSF	Spool Display and Search Facility
6	IPCS	Interactive Problem Control Facility
7	DITTO	DITTO/ESA
8	RMF	Resource Measurement Facility
9	DFSORT	Data Facility Sort
10	OMVS	MVS OpenEdition
11	RRS	Resource Recovery Services

Of these options, only SDSF is commonly accessed by end users. SMP/E, ISMF, RACF, HCD, IPCS, and RMF are the provinces of systems programmers. DITTO and the DFSORT panels might be accessed by normal system users, but this would probably be fairly rare. OMVS is usually started from ISPF Option 6. For these reasons, some installations prefer to move SDSF to the first panel.

If you use the DBSPROC logon procedure (and, presumably, IPLed with a parameter to include DB2 modules), the M panel appears as:

1	SMP/E	System Modification Program/Extended
2	ISMF	Integrated Storage Management Facility
3	RACF	Resource Access Control Facility
4	HCD	Hardware Configuration Dialogs
5	SDSF	Spool Display and Search Facility
6	IPCS	Interactive Problem Control Facility
7	DITTO	DITTO/ESA
8	RMF	Resource Measurement Facility
9	DFSORT	Data Facility Sort
10	OMVS	MVS OpenEdition
11	RRS	Resource Recovery Services
11	DB2PM	Database Performance Monitor
12	DB2DC	Database Data Collection
13	DB2	Data Base Product
14	RRS	Resource Recovery Services
15	DB2ADM	Database Admin Tool
16	QMF	Query Management Facility

3.19 Supplied jobs

A number of jobs are provided in SYS1.P390.CNTL. These should be used with care because many of the jobs correspond to older versions of the AD CD-ROM system and may not be exactly what you want for the current release. Nevertheless, this data set provides examples of many jobs you might need.



Additional system setup

You can install an AD system, as outlined in the first two chapters, and immediately start using it. Userids, disk space, IPL processes, and so forth, are all provided. This is quite acceptable for very small projects that have few long-term or large-size file requirements, have no security concerns, and so forth. Relatively few AD systems have prospects this limited. More practical systems require additional administrative setup after being installed from the initial distribution media.

For the most part, this should be done once—when installing your first AD system—and then carried forward for each release. The ability to do this (carry forward your local customizations, data sets, and so forth) is a primary goal for an OS/390 administrator. Initial planning and customization, *before using the system for productive work*, is important. Once you start using the system, it becomes more difficult to organize some of the customization functions.

The fundamental areas discussed in this chapter include the following:

- ▶ Adding new disk volumes to your configuration
- ▶ Adding a user catalog
- ▶ Defining user groups
- ▶ Adding new userids
- ▶ Data set naming conventions

These areas all interact with each other and need to be considered as a whole. Other basic customization discussed in this chapter includes:

- ▶ Authority for administrators
- ▶ Initial TCP/IP customization
- ▶ Backup options
- ▶ Security changes
- ▶ IPL customization
- ▶ Migration to new AD releases

The examples in this chapter are mostly based on the OS/390 R10 AD CD-ROM version, but should apply to other recent releases with few adjustments.

4.1 Adding new disk volumes

When we discuss new disk volumes, we intend this to cover any of the types of disk volumes that can be added to these systems. These include:

- ▶ Emulated I/O drives for P/390, R/390, S/390 Integrated Server, or MP3000.
- ▶ Channel-attached drives. For the MP3000, these could be through ESCON or parallel channels. For the P/390E-based machines, it could be through ESCON channels.
- ▶ Primary logical volumes in the MP3000.
- ▶ FLEX-ES emulated disks for Netfinity/EFS machines. (ESCON channel attachment for “real” drives is expected for Netfinity/EFS machines, but was not available at the time of writing.)

The drives could be 3380 or 3390 device types, but we will assume they are 3390s. Older drive types (such as 3350s) may be emulated, but are not supported by the current OS/390. The emulated I/O functions can support “odd size” drives, with an arbitrary number of cylinders.¹ Small emulated drives may be appropriate in some cases (such as a 50-cylinder 3390 devoted to RACF or a JES2 checkpoint), but our discussions generally assume that standard size 3390 volumes will be used. This means 3390-1 (.9 GB), 3390-2 (1.8 GB), or 3390-3 (2.8 GB). 3390-9 volumes may also be used where they are supported.

You will probably need to add several disk volumes to your AD system. There are many reasons for doing this:

- ▶ You should use dedicated scratch volume(s). The distributed AD system uses OS39M1 for scratch space. This is not good because (a) there are too many important data sets on this volume, (b) there is only a limited amount of space remaining on the volume, and (c) performance should be better with several scratch volumes.
- ▶ You should define one or more volumes for your data sets: your source code, your data files, your libraries, and so forth. By default, the distributed system will place new data sets you create on OS39M1. This is not suitable for the reasons just mentioned.
- ▶ You will probably want to allocate a larger spool data set for JES2, since the space and associated control parameters on the distributed system is limited. Most OS/390 installations have dedicated volume(s) for JES2 spool. You might place a new (larger) JES2 checkpoint data set on the same volume and the backup checkpoint data set on a system volume, such as OS3RAA or Z1RES2.
- ▶ If you are working with substantial VSAM data sets, you will probably want dedicated volumes for these. Likewise, if you are using DB2 or another database, you will probably want dedicated volumes for this.
- ▶ If you are using UNIX System Services, you will certainly want additional hierarchical file systems. The space available for /u, in the OS/390 release 10 and z/OS 1.1 AD system, is extremely limited, and is intended only for defining mount points.
- ▶ You may want a volume for additional paging data sets. There is room for additional paging data sets on the OS39M1 volume, but this is a “busy” volume and performance considerations may dictate a separate volumes.

In the ideal situation, all the volumes mentioned here (plus others, such as dedicated volumes for RACF, for catalogs, for all paging data sets, and so forth) would be separate volumes on separate channels. This is the case in very large S/390 installations, but is not a practical goal for most AD users.² We cannot recommend a particular goal because different

¹ The maximum is the largest possible 3380 drive, or 3339 usable cylinders for a 3390 drive.

² It can be a reasonable goal, with slight performance advantages, using emulated I/O and odd-sized volumes. The general trend is away from using odd-sized volumes, so the possibilities in this area are not explored here.

installations have very different usage patterns. A group working almost completely in UNIX System Services might need expanded HFS volumes and paging space, but have no use for OS/390 scratch volumes, spool space, or catalogs. Another installation working only with traditional MVS applications might never use additional HFS files, not need any more paging space, but heavily use spool and scratch volumes.

Making some arbitrary choices, a very small system might be expanded by adding three volumes (probably 3390-1s, but they could be larger if appropriate):

- ▶ LOCAL1 containing scratch space (50-75% of volume) and paging space (remainder).
- ▶ LOCAL2 containing scratch space (50% of volume) and spool space (remainder).
- ▶ LOCAL3 containing a user catalog, an HFS data set (for /u), and space for batch and TSO user data sets.

The volsers (LOCALx) are arbitrary names. A volume is made a *scratch volume* by specifying PUBLIC or STORAGE for the mount attribute; this is easiest done in the VATLSTxx member of PARMLIB. For more details:

- ▶ See “How to add additional emulated disk volumes” on page 127 for details of adding volumes.
- ▶ See “How to use OS/390 volume use attributes” on page 129 for setting mount attributes, such as PUBLIC or STORAGE.
- ▶ See “How to increase SPOOL space” on page 174 for changing the spool data set.
- ▶ See “How to provide more paging space” on page 187 for adding paging data sets.
- ▶ See “How to use a local RACF database” on page 158 for moving the RACF data base.
- ▶ See “How to add an HFS file system” on page 92 for adding a new /u file system.

A slightly larger example, appropriate for smaller MP3000 installations, might be:

- ▶ WORK01 containing scratch space.
- ▶ WORK02 containing scratch space.
- ▶ SPOOL1 containing JES2 spool (and leaving the JES2 checkpoint on its original volume).
- ▶ HFS001 containing /u and other local file systems.
- ▶ VSAM01 containing additional paging space, a user catalog, user data sets, and perhaps the RACF data base.
- ▶ USER01 containing TSO and batch data sets.

This has the advantage that the two scratch volumes should contain only scratch data sets and can be completely cleared, when appropriate. Large data bases, whether DB2 or in some other format, would probably have their own volumes.

You will undoubtedly add volumes throughout your use of OS/390. However, you should establish your pattern of disk usage and expansions as soon as possible. Notice that both examples here mention a user catalog. Establishing a user catalog is important for many AD installations, and should be done before you add any local users to your system.

For several of the examples in this chapter we created a volume named LOCAL1 at address AAF. We did this on a P/390, using a 400 cylinder 3390. You could use an MP3000 logical volume, or any other method of obtaining a 3390 volume. We used address AAF because it was already in the AD system IODF as a 3390 (see “IODFs and addresses” on page 33) and we had not used the address for anything else.

4.2 Adding a user catalog

OS/390 catalogs perform several functions, but the most basic is that they permit a user to find a data set by name.³ If a data set is not cataloged, the user must know the data set name, the volser of the volume containing it, and the type of device used. Certain types of data sets, such as VSAM data sets, have more information in the catalog and *must* be cataloged. In practice, all normal OS/390 data sets are cataloged.

An OS/390 system normally has several catalogs.⁴ One of these is selected as the master catalog at IPL time, by being named in the LOADxx member selected by the IPL parameter. (See “Understanding IPL parameters” on page 103 for more details.) The AD system contains several catalogs; see “Catalogs” on page 35 for the names of these. When a user creates a new data set, with TSO or a batch job, the data set is normally automatically cataloged.

The key point being discussed here is *which catalog* is used when your users create new data sets.

The name “user catalog” may be slightly misleading. You would not normally have a user catalog for each user. Any catalog that is not the master catalog is a *user catalog*. In principle, any catalog can be the master catalog; the master catalog is simply a catalog that is specified as the master catalog during IPL. In practice, the master catalog usually has certain required entries, so you cannot arbitrarily switch which catalog is the master. If you have multiple OS/390 systems, you can have the master catalog on one system be a user catalog to the other systems. (Or you can share a master catalog between the systems; handling catalogs for multiple OS/390s is beyond the scope of this discussion.)

Every OS/390 data set has a High Level Qualifier (HLQ) as the first part of the data set name. (See “Data set naming conventions and RACF” on page 92 for more about this.) For example, for data set ABC.DEF.GHJ, the HLQ is ABC. When asked to catalog this data set, OS/390 will look in the master catalog for an *alias* named ABC. If this alias exists, it will point to another catalog (a *user catalog*) and that is the catalog that will be used for all data sets with ABC as the HLQ. If an alias does not exist, the catalog entry will be made in the master catalog. This has several side effects:

- ▶ Users must be allowed *update* access to the master catalog if their data sets are to be cataloged there. The master catalog is critical to system operation. In general, it is not advisable to allow all users to have *update* access to it. (By default, the AD system permits all users to have unlimited access to the master catalog. This is convenient for getting started, but not a good long-term plan for most installations.)
- ▶ If practically everything is cataloged in the master catalog, it becomes large and slow. It also is a critical point of failure if something corrupts or destroys it.
- ▶ Once an entry exists in the master catalog with a given HLQ, you cannot add an *alias* for that HLQ. For example, in the default AD system, if TSO user P390B logs onto the system and creates a data set using normal TSO/ISPF facilities,⁵ that data set will be cataloged in the master catalog. You cannot then add an alias for P390B unless you uncatalog or delete all the existing P390B data sets. This is why you should think about user catalogs *before* you have users on the system. (If it is too late to think about userid P390; this one already has several cataloged data sets in the distributed AD system.)

³ The OS/390 catalogs do not apply to UNIX System Services files. USS files are found through normal UNIX directory functions. The HFS data sets, that contain the USS file systems, are normally cataloged.

⁴ This discussion ignores obsolete forms of MVS catalogs, known as CVOLs and VSAM catalogs.

⁵ Using the AD system, the first use of ISPF by a userid will automatically create an ISPF profile data set and this will be cataloged.

To continue with the examples in this chapter, we created a user catalog on volume LOCAL1. We intend to have all our locally-created users and projects catalog their data sets in this catalog. We created the catalog with this job:

```
//CATCREAT JOB 1,OGDEN,MSGCLASS=X
// EXEC PGM=IDCAMS
//SYSPRINT DD DYDOUT=*
//SYSIN DD *
  DEFINE USERCATALOG (NAME(CATALOG.SYSLAB) ICFCATALOG -
    CYLINDERS(5 2) VOLUME(LOCAL1) )
/*
```

The specified size, five cylinders primary space and two cylinders secondary allocation, should be ample for typical AD installations. The catalog name, CATALOG.SYSLAB, is arbitrary, but it is a good idea to create a name that indicates the purpose of the data set (and a catalog is a VSAM data set). The ICFCATALOG parameter indicates the type of catalog; there are other, obsolete, types. Always use type ICFCATALOG.

This catalog will not be used until we add aliases to the master catalog, pointing to this user catalog. (There are other ways to access a user catalog, such as JOBCAT JCL statements, but these are rarely used.) An example of a job that adds an alias is:

```
//CATALIAS JOB 1,OGDEN,MSGCLASS=X
// EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DEFINE ALIAS (NAME(BILL) RELATE(CATALOG.SYSLAB))
/*
```

After running this job,⁶ any data sets with the HLQ “BILL” will be cataloged in this user catalog. We will normally add an ALIAS to the master catalog as a routine part of defining a new userid or groupid. You can verify that your user catalog and ALIAS entries are present in the master catalog by using the TSO command LISTC and looking for output lines such as:

```
ALIAS .....BILL
....
USERCATALOG.....CATALOG.SYSLAB
```

Several user catalogs are distributed as part of the AD system. You could add more ALIAS statements (for your userids) and point to any of these existing user catalogs. This is not a good idea because these distributed user catalogs are on AD system volumes. When you move to a new release of the AD system these volumes will be replaced.

For completeness, a job to delete an alias might be:

```
//CATALIAS JOB 1,OGDEN,MSGCLASS=X
// EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DELETE BILL ALIAS
/*
```

The IDCAMS commands shown here, **DELETE BILL ALIAS**, for example, can be entered as TSO line commands. There is no requirement to submit a batch job to execute them. However, most experienced administrators use batch jobs. A job provides better documentation (JES2 spooled output) and you can refer back to previous jobs if something goes wrong during a series of IDCAMS commands. (After doing a series of commands and verifying that they worked correctly, you would probably purge the output from JES2.)

⁶ This assumes that no data sets with the HLQ BILL are already cataloged in the master catalog. If the HLQ is already in the master catalog, the DEFINE ALIAS function will fail.

4.2.1 Protecting the catalogs

After creating your user catalog(s), you need to think about catalog protection. The normal plan would be:

- ▶ For the master catalog, allow everyone READ access and allow your administrators UPDATE or ALTER access. Several userids already have data sets cataloged in the master catalog, so you need to provide UPDATE authority for these userids.
- ▶ Allow everyone UPDATE access to the user catalogs. This is not much protection, but it will prevent normal users from completely deleting it.

*Unfortunately, the AD system master catalog cannot be easily protected this way. The reason is a little complicated, but important to understand. OS/390 has many *started tasks* that are run as part of a normal IPL process. In the AD system, these started tasks appear to run under a userid that is the same name as the started task.⁷ Other functions (such as creation of dynamic dump data sets) may run at random times long after the system is IPLed. If any of these tasks create data sets, they will be cataloged in the master catalog because there are no aliases for these “artificial” userids. If the master catalog is protected to be read-only, then these tasks will fail because they cannot catalog their data sets.*

We realize that reading about user catalogs is heavy going when you have a new system and are anxious to start using it. *However, user catalog planning and creation is the single most important “local customization” step you should take when working with your system.*

4.3 Defining user groups

Large OS/390 installations often have very elaborate structures of groups. Small OS/390 installations often ignore group definitions completely and use whatever is supplied with their initial system. There are two reasons you should define groups:

- ▶ A little thought and work with groups will greatly simplify protection of data sets.
- ▶ A group name can be a project name, and project data sets can use the HLQ of the group name. For example, if you have a project named PAYROLL, you can then have data sets named PAYROLL.JAN.INPUT, PAYROLL.SOME.OTHER.NAME, and so forth.

A common (and recommended) plan for establishing groups is something like this:

- ▶ Every userid belongs to an initial group, say EMPLOYEE. This group has access to various data sets that all employees should be able to access. As a trivial example, every user should be able to read the company telephone directory data set.
- ▶ Groups are defined for major projects, departments, and functions. Each group is permitted appropriate access to data sets relevant to that group. For example, a group named SECRETARY might have update access to the telephone directory data set.
- ▶ Userids are *connected* to various groups as their duties require. For example, all secretaries would be members of the EMPLOYEE and SECRETARY groups. Individual secretaries would probably also be members of groups related to their particular departments.
- ▶ Data sets are protected and appropriate groups are permitted access to various data sets. Users can access protected data sets by virtue of being a member of a group that has access to the data set. Permitting individual userid access to various data sets should be rare; permissions should be at the group level instead of the user level.

⁷ In a system with a more complete security environment, all the started tasks we mention would use the RACF STARTED class to assign one userid (or a few userids) to the started tasks. There would be a PERMIT profile allowing this userid to have UPDATE access to the master catalog.

This scheme, if used with well-designed generic data set profile names, can result in a very simplified, easy-to-manage RACF structure for data set protection. It is important, however, not to get carried away with a group structure that is too elaborate.⁸ For a small AD system, anything more than, say, 20 or 30 groups should be viewed with suspicion. (Recommendations like this do not count artificial groups, such as group CATALOG, that are not associated with real userids.)

For the purposes of this documentation, we will create a group named WORKSHOP. In addition we will use group SYS1 as the “all users” group. Almost every OS/390 system has a group named SYS1,⁹ and this is often the default group if a group name is not supplied as part of a new user definition. The commands are:

```
AG workshop OMVS(GID(500))
AD 'workshop.*' UACC(NONE)
SETROPTS GENERIC(DATASET) REFRESH
PERMIT 'workshop.*' ID(workshop) ACCESS(ALTER)
```

What we did here was:

- ▶ Create a group named *workshop*. We assume that members of this group might use USS functions; therefore the group should have a UNIX GID. We arbitrarily selected GID 500. Basic OS/390 does not have a mechanism to automatically assign GID numbers; you must keep track of the numbers you have used.
- ▶ The (optional) **AD** command creates a generic data set protection profile. The default access to any data set with the HLQ workshop is NONE.
- ▶ The *workshop.** profile is a generic profile. These may be cached in memory. The **SETROPTS** command refreshes the in-memory cache, if it exists.
- ▶ The **PERMIT** command gives any member of group workshop complete access to all data sets with the HLQ workshop.

The profile formats used here (with a single asterisk) assume that EGN is off, as it is in the distributed AD system. There are other ways to accomplish the same group authority, using CREATE authority, for example, when connecting a userid to a group. We elected to use the method shown here.

Assuming we want to use our user catalog for all data sets owned by the workshop group, we would run this job:

```
//CATALIAS JOB 1,OGDEN.MSGCLASS=X
// EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DEFINE ALIAS (NAME(WORKSHOP) RELATE(CATALOG.SYSLAB))
/*
```

All that remains is to connect appropriate userids to group workshop. This is covered in the next section.

We suggest you devise and implement a group structure, even if it is as simple as two or three groups, before you start any significant use of your AD system.

⁸ Group structures that are out of control can become a nightmare for a security manager. The author once participated in an audit of an MVS installation with over 50,000 group definitions for approximately 20,000 TSO users. It was totally out of control and extremely difficult to even begin to fix the problems without impacting production jobs.

⁹ The normal RACF installation process creates a SYS1 group.

4.4 Adding new userids

The next few sections provide examples of typical processes used when adding, deleting, or administering users. It assumes the use of RACF (which is a component of the OS/390 Security Server), since this is a standard part of the AD CD-ROM system. As with many OS/390 administrative tasks, there are a number of ways to do things. For newer owners, we recommend the following:

- ▶ Use a batch job to define new users.
- ▶ Use TSO RACF commands to alter the RACF profiles of existing users.
- ▶ Use RACF ISPF panels to work with RACF general resource profiles and other more advanced RACF functions.
- ▶ Use TSO RACF commands to manage the more straightforward RACF data set protection profiles that you may define.

These recommendations may be contrary to your experience. In particular, you may prefer to use the RACF ISPF panels for all RACF functions. This is fine if you have the experience to do it; however, many new users find the panels more difficult than the TSO commands. Also, as with many panel and menu interfaces, documentation and examples are more difficult than with command-line interfaces. We think examples of the RACF command-line interfaces are easier to read than examples of RACF panel usage.

Adding a new user usually involves several operations, not all of which involve RACF. The RACF portion can be done with the RACF panels, with TSO commands, or with a batch job that executes TSO commands. We recommend the batch job approach, and that is what is documented here. In particular, *we recommend that you set up a batch job that you use every time you add a new user.* This provides a simple, repeatable method that can be used by anyone (anyone with sufficient RACF authority, that is), with a low error potential.

The specific examples here match the AD system. With minor adjustments they should be usable for other systems. Read and understand this entire section before attempting to use this job.

The following is a complete listing of a batch job to add a new user. In this example, *ogden* is the new userid, and *ogden* will have access to TSO and OMVS.

```
//USERADD JOB 1,BILL,MSGCLASS=X,NOTIFY=P390A
//TSO EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUADS DD DSN=SYS1.UADS,DISP=SHR
//SYSLBC DD DSN=SYS1.BROADCAST,DISP=SHR
//SYSTSIN DD *
AU ogden NAME('BILL OGDEN') PASSWORD(ogden) OWNER(SYS1) +
  DFLTGRP(sys1) +
  TSO(ACCTNUM(1) PROC(BILLPROC) JOBCLASS(A) MSGCLASS(X) +
  HOLDCLASS(X) SYSOUTCLASS(X) SIZE(40000) MAXSIZE(0)) +
  OMVS(HOME(/u/ogden) PROGRAM(/bin/sh) UID(525))
CONNECT ogden GROUP(WORKSHOP) AUTHORITY(CREATE)
PERMIT JCL CLASS(TSOAUTH) ID(ogden) ACCESS(READ)
PERMIT OPER CLASS(TSOAUTH) ID(ogden) ACCESS(NONE)
PERMIT ACCT CLASS(TSOAUTH) ID(ogden) ACCESS(NONE)
PERMIT 'WORKSHOP.*' ID(ogden) ACCESS(ALTER)
AD 'ogden.*' OWNER(ogden) UACC(NONE) GENERIC
SETROPTS RACLIST(TSOAUTH) REFRESH
BPXBATCH SH mkdir /u/ogden
BPXBATCH SH chown ogden /u/ogden
```

```

LU usery TSO OMVS
LD DA('ogden.*') ALL
/*
//ALIAS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//*                               VERIFY THAT ALIAS NAME IS UPPER CASE
//SYSIN DD *
  DEFINE ALIAS (NAME(OGDEN)    RELATE(WORKSHOP.CATALOG))
/*
//ISPF EXEC PGM=IEFBR14
//D1 DD DSN=OGDEN.ISPF.ISPPROF,DISP=(,CATLG),VOL=SER=WORK01,
//  UNIT=3390,SPACE=(TRK,(4,2,10)),DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
//*                               VERIFY THAT ALL JCL IS UPPER CASE

```

To run this job (or to use the equivalent RACF commands from TSO), you must have SPECIAL authority in RACF. The P390 userid supplied with the AD system has SPECIAL authority, and we suggest you log onto TSO with this userid when adding or changing user definitions. The IBMUSER userid also has SPECIAL authority and can also be used. You can define new users with SPECIAL authority, of course, and use these new userids for further administration.

We need to explain some of the details in this job:

- ▶ There are three job steps. The first executes the TSO command processor, and supplies several RACF and other commands as input. The second executes IDCAMS and adds an ALIAS to your master catalog. The third simply allocates a new data set that will be the ISPF profile data set for this user. The ALIAS step is needed only if you have defined a user catalog (and want this new userid to be in it). The AD system, as distributed, does not have a user catalog for TSO users and you do not need this step. (TSO user data sets are cataloged in the master catalog.) The AD system will automatically create an ISPF profile data set for a new user; this is done by the CLIST that is run at logon time. However, you can create the ISPF profile data set as shown here and control where it is placed.
- ▶ Several RACF commands are executed: **AU** (ADDUSER), **CONNECT**, several **PERMITs**, **AD** (ADDSD or add data set profile), **SETROPTS**, **LD** (LISTDSD or list data set profile), and **LU** (LISTUSER). The **LD** and **LU** are optional, of course; their output should be used to verify that the **AU** and **AD** commands produced the expected results.
- ▶ The **AU** command includes segments for TSO and OMVS, and could be extended to include CICS and OPER segments if appropriate.
- ▶ The key parameters that *must* be changed for each new user are:
 - The userid (which is the operand just after the AU command)
 - The NAME field (which can be listed in RACF reports, but which has no direct processing function, should reflect the user's name)
 - The UID value in the OMVS segment
 - The HOME directory in the OMVS segment (usually lowercase)
 - The userid in the **CONNECT** command
 - The userid in the **PERMIT** commands
 - The userid in the **AD** command
 - The directory names in the **BPXBATCH** commands (and these are usually lowercase)
 - The userids in the **LD** and **LU** commands
 - The NAME field in the **DEFINE ALIAS** command (uppercase)
 - The HLQ in the DSN parameter in the ISPF job step (which should be set to the new user's userid)
This must be uppercase.

- ▶ Note that many of these can be changed by a single global **change** command using the ISPF editor.

Assuming the group names (workshop and sys1) do not change, the new userid in this example appears in several places (including directory names and MVS data HLQs). If you use a global change with a file like this, be certain that all your final JCL is upper case, the ALIAS name is upper case, and UNIX directory names are lower case. Unfortunately, there is no single **change** command that will make all the upper/lower case entries correct.

Other considerations for this sample job are:

- ▶ Many of the basic parameters shown will probably be the same for most of your users. These include the OWNER, DFLTGRP, ACCTNUM, PROC, JOBCLASS, MSGCLASS, HOLDCLASS, SYSOUTCLASS, SIZE, MAXSIZE, TIMEOUT, XRFSSOFF, and PROGRAM fields. The values shown in the example are reasonable starting points for the AD CD-ROM systems.
- ▶ The PASSWORD field provides an expired password. The new user will be forced to select a new password the first time he logs onto TSO. You probably should select a unique PASSWORD for each new user (even though he will be forced to change it immediately), but most administrators use a constant initial password. The security exposure is real, but slight.
- ▶ The value in the ACCTNUM field must be known to RACF. If you are not planning to use the account information (and few AD owners are interested in this), we suggest using the value "ACCT#" which is already defined to RACF in the AD CD-ROM system. This account number applies only to the TSO logon account number; it is not related to the account number field in a JOB statement.
- ▶ The UACC value of NONE (in the AD statement) will prevent any other user or job from reading this user's sets. You could set the UACC value to READ, depending on the needs of your installation. READ is a typical value for a development shop, while NONE would be more common in a commercial production operation.
- ▶ The **PERMIT ACCT** and **PERMIT OPER** commands are used primarily to control SDSF security, and this is addressed in "Understanding basic SDSF security" on page 181. These commands have side effects and you may want to delete them from this job.
- ▶ After a new user is added by using this job, you can use direct RACF commands to fine tune any unique requirements for that user. In other words, we suggest using a fairly generic set of parameters (as shown in the example) to add all new users. You would then use the RACF **ALTUSER** and **PERMIT** commands to make minor changes or additions.
- ▶ If you are not using OMVS, you would delete the associated line from the job. You might add other segments, such as OPERPARM, CICS, DCE, DFP, NETVIEW, or WORKATTR if some of these will apply to all your normal users.
- ▶ The system does not automatically assign a UID. You must manually manage UID assignment. Any number up to 2**31 can be used, but most installations prefer an orderly assignment of UIDs.
- ▶ An OMVS user (any user who has an OMVS segment in RACF) must belong to a group that also has an OMVS segment. Group SYS1, which is assigned as the default group in the job above, has an OVMS segment.
- ▶ Note that the **AU** command is a single command. that flows over several lines. Remember to maintain the continuation marks (which are the + symbols in the example).
- ▶ The BILLPROC in the PROC parameter is the name of the small TSO logon procedure we described in "ISPFLITE procedure" on page 45 and "ISPFLITE procedure" on page 75. You could use the full-function procedure, ISPFPROC, that is supplied with the AD system.

- ▶ The example here places the user in the default group SYS1, and connects him to group WORKSHOP. If you have defined a group structure, you would probably have more **CONNECT** commands and you would probably need to tailor these for each new user. If you are using your own group structures, you want to have List of Groups enabled. See “Unexpected RACF parameters” on page 42 and “Unexpected RACF parameters” on page 72.
- ▶ If you have enabled Enhanced Generic Naming (EGN) in your RACF system, the parameter in the AD command (in the sample job above) would change to OGDEN.** (with two asterisks). By default, EGN is not enabled in the AD CD-ROM systems.

For further information refer to *Security Server (RACF) Security Administrator's Guide* SC28-1915, or *Security Server (RACF) Command Language Reference*, SC28-1919.

4.4.1 Adding TSO account numbers and procedures

A TSO user must be authorized to use whatever logon account numbers and TSO logon procedures he needs to use. In the example above, account number 1 and logon procedure BILLPROC are used.¹⁰ Neither of these are present in the distributed AD system. The system administrator can make these available with the following commands:

```
RDEFINE TSOPROC billproc UACC(READ)
RDEFINE ACCTNUM 1 UACC(READ)
```

The distributed AD system already has these defined:

```
RDEFINE ACCTNUM ACCT# UACC(READ)
RDEFINE TSOPROC ISPFPROC UACC(READ)
RDEFINE TSOPROC ISPFLLITE UACC(READ)
RDEFINE TSOPROC DBSPROC UACC(NONE)
```

The significance of UACC(READ) is that anyone can use these resources for their intended purpose (TSO account number or logon procedure). If the UACC is NONE, then an explicit PERMIT command is needed to authorize someone to use the resource. For example, the TSO logon procedure DBSPROC is distributed with the AD system; it provides additional DB2 options on the extended ISPF primary menu. The TSOPROC profile for DBSPROC has UACC(NONE), and explicit PERMIT commands were issued for a few users. If user *ogden* (in the example above) wants to use this logon procedure, then the system administrator must issue **PERMIT DBSPROC CLASS(TSOPROC) ID(OGDEN) ACCESS(READ)** to allow this.

If your installation is working with DB2, you may want to allow all your TSO users to use the DBSPROC logon procedure.¹¹ You can do this by issuing PERMIT commands for all of them, or by changing the UACC for this profile to READ. The command to do this is:

```
RALTER TSOPROC DBSPROC UACC(READ)
```

If you examine the RACF data base for the distributed AD system, you will see that profiles with UACC(READ), such as the ISPFPROC profile, also have a long list of permitted users. This is redundant, since all users have default READ access. The list of permitted users is left over from an earlier AD release in which the UACC was NONE.

¹⁰ Very few AD installations use the TSO account number for anything. However, an account number must be specified in certain fields. We arbitrarily use an account number of 1. Many batch job examples also use 1 as the account number. No special RACF permission is needed to use a batch job account number.

¹¹ A TSO user can specify the logon procedure he wants to use by changing a field in the logon password panel. If he is authorized to use the new logon procedure (and the procedure exists and works correctly), the logon process will use his specified procedure.

4.5 How to add an HFS file system

The recent AD systems are provided with a very small /u file system. The intention is that you use your own file system (not on a system volume) for /u. The following steps can be used to do this:

- ▶ Allocate a new HFS data set on one of your volumes. You need to give some thought to the size; this is so variable that we cannot make a specific suggestion. For our example, we will use 500 cylinders of 3390 space.

Go to ISPF option 3.2, allocate (option A) a data set named 'HFS.BILL.U', and place it on your target volume (we used volume WORK03). On the second panel of the 3.2 function, use:

```
Space units:          CYLS
Primary quantity:    500
Secondary quantity:  50
Directory Blocks:    0
Record Format:        U
Block Size:          0
Data Set name type:  HFS
```

- ▶ See if it works. You can do this (assuming no one is using the old /u file system) with the following steps:

```
In ISPF option 6, enter the command ish
On the top line of the ish shell, select File-Systems
Select option 1 (Mount Table)
Enter U next to HFS.USERS to unmount this file system (the original /u)
PF3
Select File-Systems again
Select option 3 (Mount)
Mount point: /u
File System Name: 'HFS.BILL.U'      (the single quotes are needed)
File System Type: HFS
Enter
```

- ▶ You should be able to access /u and the **df** command should indicate a much larger size.
- ▶ To automatically mount the new /u, edit the BPXPRMxx member in PARM lib that you will IPL with the next time, and change it like this:

```
...original content....          ....new content.....
MOUNT FILESYSTEM('HFS.USERS')    MOUNT FILESYSTEM('HFS.BILL.U')
TYPE(HFS)                        TYPE(HFS)
MODE(RDWR)                       MODE(RDWR)
MOUNTPOINT('/u')                 MOUNTPOINT('/u')
```

- ▶ Shut down your system and reIPL. You should now have your new /u file system. If you are happy with it, update all the BPXPRMxx members you are likely to use.

The name HFS.BILL.U is arbitrary, of course, but we suggest you use a HLQ of HFS. This will avoid making changes to RACF. (See "OMVS not trusted" on page 73 for details.) When you install the next release of z/OS, you can simply mount this HFS instead of the distributed /u and you have effectively migrated your /u filesystem to the new release.

4.6 Data set naming conventions and RACF

Very briefly, the rules for OS/390 data set names are these:

- ▶ A name is up to 44 characters (upper case, counting periods), separated into qualifiers of no more than 8 characters separated by periods. For example, XXX.YYY.AAA.BBB is a

valid name, with four qualifiers. The high level qualifier (HLQ) is XXX. The names XXX.YYY.abc.CCC and XXX.ABCDEFGHIJ.ZZZ are invalid; one contains lower-case characters and the other contains a qualifier that is longer than eight characters.

- ▶ Single-qualifier names are strongly discouraged.
- ▶ There is no inherent meaning to a name. In the PC world, for example, game.exe indicates an executable file; there is no equivalent naming requirement for OS/390. A data set named XXX.YYY might contain executables, or C source, or REXX commands, it might be a library, or it might be a simple sequential file, and so forth. Names do not mean anything.¹²
- ▶ There are naming *conventions*, not automatically enforced by the system. The conventions are a little fuzzy, but can usually be recognized. Common conventions include:
 - An HLQ should be a userid or group name.
 - The letters LIB somewhere in a lower-level qualifier usually indicates a library (PDS or PDSE).
 - Qualifiers of CNTL, JOB, or JCL usually indicate JCL and utility programs.
 - Qualifiers of COB, COBOL, C, ASM, FORT, and so forth, usually indicate source code (possibly with JCL mixed in).
 - Qualifiers of OBJ or OBJECT usually indicate compiler output (before linking).
 - Qualifiers of LINKLIB, LOAD, LOADLIB indicate a library of executables.
 - A qualifier of CLIST usually indicates a TSO or REXX command script.

There are only two significant goals for the names. They should indicate the nature of the data set, within reason, and they should meet your security naming requirements.

By default, OS/390 does not protect data sets. By default, any user can create a data set with any name he chooses, and place it on any disk volume that has space. This degree of freedom may be acceptable (but not necessarily desirable) for very small installations. In practice, we need some amount of data set protection—if only to prevent accidental erasure of key system data sets.

The release 10 AD system contains a number of RACF profiles to protect data sets, including:

- ▶ A generic profile for SYS1.* with UACC(READ). This protects key system data sets such as SYS1.PROCLIB, and permits all users to read them. User P390 and any member of group SYS1 are permitted ALTER access. (The inclusion of SYS1 is possibly too broad, since this is the default group for any new user.)
- ▶ Many profiles for product data sets, with UACC(READ). These have no permitted users with higher levels of access. Of course, a user with RACF OPERATIONS privileges can alter any of them. Userids P390 and IBMUSER have this privilege and would normally be used to make changes to various products included with the system.
- ▶ A generic profile for USERCAT.* with UACC(READ), which protects the user catalogs distributed with the system.
- ▶ A number of data set profiles that appear to serve no useful purpose and were probably left over from system testing.
- ▶ No data set profiles for any of the common, supplied userids, such as P390, P390A, P390B, and so forth. There is a profile for IBMUSER.* with UACC(READ).

In summary, there is reasonable protection for the system itself (except for the master catalog, as discussed in “Protecting the catalogs” on page 86), but not for anything else. Therefore, you need to plan your own data set protection and you can assume you are starting from a relatively clean RACF position.

¹² As always, there are exceptions. Low-level qualifiers of DATA and INDEX have default meanings for VSAM, for example.

An ideal situation would be something like this:

- ▶ You have a number of users: *joe*, *patti*, *chris*, and *bill*. When you create each userid, you create a generic data set profile for that userid:

```
AD 'joe.*' OWNER(p390) UACC(read) GENERIC
AD 'patti.*' OWNER(p390) UACC(none) GENERIC
AD 'chris.*' OWNER(p390) UACC(read) GENERIC
and so forth
```

- ▶ You create groups with the names of your major projects: *payroll*, *secret1*, *projectx*, *bigbucks*, and so forth. When you create each group, you create a generic data set profile for that group name and permit members of the group appropriate access:

```
AD 'payroll.*' OWNER(p390) UACC(none) GENERIC
PERMIT 'payroll.*' ID(payroll) UACC(ALTER)
AD 'secret1.*' OWNER(p390) UACC(none) GENERIC
PERMIT 'secret1.*' ID(secret1) UACC(ALTER)
AD 'projectx.*' OWNER(p390) UACC(read) GENERIC
PERMIT 'projectx.*' ID(projectx) UACC(ALTER)
and so forth
```

- ▶ In every case you select a UACC of *alter*, *read* or *none*, as appropriate. The OWNER can be the target userid, but it is typically the system administrator in simple cases.
- ▶ You connect users to groups, as required:

```
CONNECT joe GROUP(payroll) AUTHORITY(CREATE)
CONNECT joe GROUP(projectx) AUTHORITY(CREATE)
CONNECT patti GROUP(secret1) AUTHORITY(CREATE)
and so forth
```

- ▶ You create a user catalog, and create an ALIAS in the master catalog (pointing to the user catalog) for every userid and groupid that you create.
- ▶ You protect the master catalog with UACC(READ). (This is awkward to do for the AD system, for reasons pointed out in “Protecting the catalogs” on page 86.

With this arrangement, all the data sets on your volumes should have an HLQ that matches a userid or groupid.¹³ All these data sets will have basic protection provided by the generic profiles for the HLQ. If a user needs access to another project, this is done with a single **CONNECT** to make the user a member of the group that deals with the appropriate project.

Situations are seldom as clean as this, so you may need a few individual **PERMIT** commands, such as:

```
PERMIT 'projectx.*' ID(bill) ACCESS(read)
```

Ideally, your goals might be as follows:

- ▶ Have as few RACF data set profiles as possible, consistent with protecting your data. Fewer is better because it is much easier and cleaner to administer over time.
- ▶ Discourage users from entering their own RACF data set protection commands. Any user can enter **AD** and **PERMIT** commands to protect data sets whose HLQ matches his userid. Over time, this can create large numbers of profiles that are difficult to clean up and control. It might be best if normal users are unaware that they can issue RACF commands.¹⁴

¹³ If the master catalog can be protected to be read-only, the inability of normal users to catalog data sets in the master catalog would discourage them from creating data sets with unplanned HLQs.

¹⁴ There are many ways a system owner can control the availability and effect of RACF commands. However, the purpose of this document is to discuss using the AD system in a fairly simple environment. Administrators with the skills to highly customize the RACF command environment will probably not be reading this document.

- ▶ Attempt to have **PERMIT** commands reference group names instead of individual users. However, do not create unreasonable numbers of groups solely for this purpose. You will always have some **PERMIT** commands for individual users.

4.7 Authority for administrators

Systems administrators need more authority than normal users. On smaller systems, such as a typical AD installation, there is only one level of administrator, with all possible authority. As shipped, userids P390 and IBMUSER have the most authority.

In general, RACF SPECIAL is the starting point for authority. With SPECIAL, a user can give himself (or others) whatever other authority they need. For initial system administration, and for most common tasks, the administrator needs these authorities:

- ▶ RACF SPECIAL (to execute any RACF command)
- ▶ RACF OPERATIONS (allow access to any data set)¹⁵
- ▶ UID=0 (for UNIX System Services)
- ▶ At least READ access to OPER in the TSOAUTH class
- ▶ At least READ access to ACCT in the TSOAUTH class
- ▶ At least READ access to JCL in the TSOAUTH class

As a quick review, if you have SPECIAL authority with RACF, you could issue the following commands (from a TSO command line) to give *joe* all these authorities:

```
ALU joe SPECIAL OPERATIONS
ALU joe OMVS(UID(0))
PERMIT OPER CLASS(TSOAUTH) ID(joe)
PERMIT ACCT CLASS(TSOAUTH) ID(joe)
PERMIT JCL CLASS(TSOAUTH) ID(joe)
SETROPTS RACLIST(TSOAUTH) REFRESH
```

You can remove the authorities with these commands:

```
ALU joe NOSPECIAL NOOPERATIONS
ALU joe OMVS(UID(nnn)) (UID other than 0)
PERMIT OPER CLASS(TSOAUTH) ID(joe) DELETE
PERMIT ACCT CLASS(TSOAUTH) ID(joe) DELETE
PERMIT JCL CLASS(TSOAUTH) ID(joe) DELETE
SETROPTS RACLIST(TSOAUTH) REFRESH
```

General UNIX advice applies about UID=0. It is better to have a userid with a non-zero UID, and then use the **su** command to switch to UID zero. If your OS/390 system has profile BPX.SUPERUSER defined in the RACF FACILITY class,¹⁶ then a user (who is not currently UID=0) must be permitted to this profile in order to **SU** to UID=0. Instead of defining userid *joe* with UID=0, you could use the following command to permit *joe* to **SU** to UID=0:

```
PERMIT BPX.SUPERUSER CLASS(FACILITY) ID(joe) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

4.7.1 SDSF authority

The OPER and ACCT authorities permit a user to access the output of any job in SDSF. OPER permits a user to enter operator commands from TSO and SDSF. JCL permits a user to submit jobs from TSO. (All users normally have JCL authority.)

¹⁵ There are minor exceptions. If the user with OPERATIONS has an explicit NONE access to a data set (or is a member of a group that has NONE access to a data set) then this NONE overrides the power of OPERATIONS authority.

¹⁶ The 10 AD system has this profile defined. IBMUSER is the only user in the distributed access list for the profile. (This does not make complete sense, because IBMUSER is defined with UID=0.)

It is unfortunate that the default SDSF characteristics use the OPER and ACCT privileges as authority indicators. With OPER authority, you can see all jobs with SDSF (but not the data in other users' SYSOUT). (Without OPER, you can see only your own jobs.¹⁷) With OPER and ACCT, you can see all jobs with SDSF and inspect all the SYSOUT data sets. However, OPER also gives the authority to read the system log *and issue operator commands* through the log screen. In most cases, you do not want normal TSO users issuing system operator commands. This means you do not give them OPER authority. In turn, this means that the normal TSO users can see only their jobs with SDSF.

See "Understanding basic SDSF security" on page 181 for more discussion about SDSF authorities.

4.8 Initial TCP/IP customization

You probably want to connect and bring up your OS/390 TCP/IP connection as soon as possible after installing your AD system. If we assume the following simple case, we can do this easily:

- ▶ You have a single Ethernet connection for OS/390, through an emulated I/O connection.
- ▶ You have MPTS set up correctly and you want to use LAN adapter number one. (This is an OS/2 function, used by emulated I/O.)
- ▶ Your DEVMAP has two addresses (with even/odd numbers) defined for the LCS3172 device manager. For a P/390-based system, the addresses are E20 and E21. For an MP3000, the DEVMAP has intermediate Unit Addresses and the IOCDs uses these Unit Address for device numbers E20 and E21.
- ▶ You know the fixed IP address you want to use.

You will need to edit file TCPIP.PROFILE.TCPIP. However, you cannot edit this file while OS/390 TCP/IP is active (because the TCPIP started task allocates the file as SHR). The AD system automatically starts TCPIP during the IPL process. You need to issue these operator commands:

```
P TCPIP
P NFSS          (release 10 only)
C NFSS          (release 10 only)
C NFSC          (release 10 only)
P NFSC          (z/OS R1.1 only)
```

The reasons for the NFSS commands in R10 are discussed in "NFS server operation" on page 44. The NFSC task also needs to be stopped in order to edit the TCPIP DATA data set. After these are stopped, use ISPF to edit TCP.PROFILE.TCPIP:

- ▶ Find the section that has DEVICE and LINK statements. You want to comment out (or delete) all the existing statements and insert these two lines:

```
DEVICE LCS1 LCS E20
LINK ETH1 ETHERNET 1 LCS1
```

- ▶ Find the HOME statement and comment out (or remove) the existing IP address. Insert statements something like this:

```
HOME
150.160.170.123          (using your fixed IP address, of course)
```

¹⁷ In general, a job is your job if the jobname begins with your userid.

- ▶ Find the GATEWAY statement and comment out (or remove) the existing lines, including the GATEWAY statement itself and any *defaultroute* statement. Insert lines something like this:

```
BEGINRoutes
;   Destination      Mask           First-Hop  Link  MTU
ROUTE 150.160.170.0  255.255.255.0   =          ETH1  mtu 1492
ENDRoutes
```

- ▶ Find the START statements (last lines in the file) and comment out (or remove) the existing statements. Insert this line:

```
START LCS1
```

The goal is to define one IP interface to your immediate network. The “1” in the LINK statement is the MPTS adapter number; you might have a different number for this. The ROUTE statement defines a static route from the LAN adapter (represented by the = symbol) to your local subnet. Use the 1492 size (or smaller) for Ethernet connections through emulated I/O.

Edit TCPIP.TCPIP.DATA. Change the HOSTNAME and DOMAINORIGIN lines as appropriate. Comment out the NSINTERADDR line, or point it to your name server if you know the IP address.

Start OMVS and edit /etc/resolve.conf. Change the DomainOrigin, Domain, and Hostname lines to match those in TCPIP.TCPIP.DATA. Either remove the NSINTERADDR line or point it to your name server.

If you are not using a name server, create /etc/hosts with a single line:

```
150.160.170.123 my.host.name           (use appropriate values, of course)
```

If you start the default DB2 environment distributed with the AD system, it will start a remote component that performs a GETHOSTBYADDR function. With no name server or /etc/hosts data,¹⁸ this will fail and issue an operator warning message every few minutes. If you do not start DB2, you probably do not need the /etc/hosts entry (unless you want to expand it and use it to access other systems by name).

Start TCPIP again, with the operator command **S TCPIP**. It will restart FTP automatically. You should be able to **PING** from a TSO command line (such as the ISPF option 6 line) using numeric IP addresses. If this works, you should be able to connect a client TN3270 session to your AD system, using the IP address you specified in the HOME statement and port number 23. You should be able to telnet to OpenEdition by using port number 1023.

There is much more to TCP/IP setup and operation. A good reference is *OS/390 IBM Communications Server: IP Configuration Reference*, SC31-8726.

4.9 IPL customization

You can customize your IPL setup in many ways. By customize, we mean:

- ▶ Select which IODF is used.
- ▶ Specify whether to perform a CLPA function, which reloads the pagable line pack area.
- ▶ Cold start JES2.
- ▶ Select various PARMLIB members to cause different subsystems, link-list libraries, authorized libraries, and so forth to be selected.

¹⁸ This could probably be supplied through the SITEINFO function also, but we did not try it.

- ▶ Specify what subsystems are started automatically, such as VTAM, TSO, CICS, DB2, and so forth.

We cannot look into all the different ways you might customize your system. We will mention some very minor changes that the author uses for his system. We normally use a basic system, with SYS1.IODF01, and VTAM, TSO, and TCP/IP running. This is produced by IPLing (once) with a CS parameter and IPLing thereafter with a 00 parameter. (See “Detailed IPL parameter controls” on page 29 and “Detailed IPL parameter controls” on page 57 for more about these parameters.)

However, we sometimes use CICS and DB2. With the distributed system, this would mean re-IPLing with the DC parameter, and thereafter (as long as we wanted to use DB2 and CICS) with the DB parameter. With these parameters, DB2 and CICS are started automatically (by the VTAMDB command list for VTAMAPPL). We removed the following statements from the VTAMDB member in PARMLIB:

```
-START DB2
PAUSE 60
S CICSA
PAUSE 60
```

and then used the DC and DB IPL parameters. This causes the PLPA and other subsystem functions to be initialized for DB2 and CICS, but then does not automatically start DB2 or CICS. If we want to use them, the operator commands **-START DB2** and/or **S CICSA** will start them. The extra overhead from loading DB2 and CICS elements in PLPA and subsystem tables is minor, even on a P/390-based machine. Once started, the subsystems can be stopped with **-STOP DB2** and **C CICS** commands.

4.10 Migration

If you install a new release of the AD CD-ROMs, you potentially lose all the customization you have done with your old release. This includes:

- ▶ Changes made to RACF. The RACF database is on the OS39M1 volume. If you have defined new users or groups, changed or added general resource profiles, protected data sets, and so forth, you are unlikely to want to repeat all these steps again for a new release.
- ▶ Catalog entries in the master catalog. As distributed, any data sets that you catalog will be cataloged in the OS/390 master catalog. If your data sets have HLQs that match those used for catalog aliases defined in the AD system, such as DLIB, then these would be cataloged in one of the predefined user catalogs. Of course, these AD user catalogs are on volumes that are replaced by a new release, so they are also lost.
- ▶ Spool contents. This is usually not considered a serious problem.
- ▶ Normal configuration data sets such as SYS1.PARMLIB, SYS1.VTAMLST, TCP profiles, procedure libraries, and so forth.
- ▶ Modules you may have added (or altered) in various system load libraries.

Some (but not all) of these areas are handled by the migration jobs included with OS/390 AD CD-ROM releases. However, the migration jobs are designed only for moving from release n to release $n+1$; that is, they assume you install every new release. Because of this, and because the migration jobs do not automate all the release-to-release transition, you may want to manage your own release transitions and not use the migration jobs.

4.10.1 Managing your own migration

Moving to a new OS/390 or z/OS release can be a major undertaking in a large, complex installation, and there are many ways to go about it. Even in a small P/390-size installation there can be many ways to go about it. One way is embodied in the *migration jobs* that are supplied with the AD CD-ROM systems. If you plan to move to *every* new release of the AD CD-ROMs, you should consider using the migration jobs—this should require less effort than managing your own migration to new releases.

Managing your own migration to new releases may not be difficult *if you establish some early discipline in managing your OS/390 system*. To use the process described here, you must:

- ▶ Keep track of all changes you make to SYS1.PARMLIB, SYS1.VTAMLST, system procedure libraries, TCP/IP profiles, and so forth. For example, every time you change one of these members, you could copy the changed member to a TSO library on one of your local volumes.
- ▶ Isolate all catalog activities for your users into one (or more) user catalogs. In practice, this means adding an ALIAS to the master catalog for every TSO user (or major application suite) you define. It means ensuring that there are no entries in the master catalog for any locally-defined data sets. *Establishing a user catalog and ALIAS definitions should be done before defining any local TSO userids*; otherwise, these new userids will generate entries in the master catalog and you will not be able to define an ALIAS for them.
- ▶ Keep a list of the ALIAS names you have added to the master catalog.
- ▶ Keep all local data sets off the volumes provided by the CD-ROMs, so that they can be replaced without losing any of your files.
- ▶ Create your own /u file system on one of your local volumes; that is, your own HFS data set with the mount point /u.
- ▶ Establish and use your own RACF database. (You may be required to run one or two of the AD migration jobs, to update your old RACF data base with any new RACF *class descriptions* and new profiles. These job(s) are the most valuable migration jobs provided with the AD system.)

If you have established the disciplines just described, then moving to a new release is fairly simple:

- ▶ Delete the old release volumes, or keep them if you have sufficient disk space.
- ▶ Install your new release from the CD-ROMs and IPL it.
- ▶ Log on as userid P390 and take the following steps:
 - IMPORT your user catalog. (See “How to migrate your user catalog” on page 134.)
 - Add your list of ALIASs to the new master catalog.
 - Reapply your changes to SYS1.PARMLIB, SYS1.VTAMLST, and so forth. This will take some time, since you may need to verify that some of the changes are appropriate for the new release. If you maintained sufficient discipline to record your changes, as outlined previously, this step should go fairly smoothly. When finished, re-IPL to ensure there were no drastic errors. At this point, you can recover by simply unzipping the SCPMV5 volume (which contains parameter data sets you are updating) from CD-ROM again.
 - Uncatalog the distributed RACF data bases (main and backup), and catalog your old RACF data bases. (This assumes they have the same names.)
- ▶ IPL again.

This process will lose any new RACF profiles that are included with the new OS/390 release. This may or may not be a problem for you, and a resolution will depend on manual action. If your P/390 OS/390 system is used in a well-controlled environment (where no one is actively attempting to attack the system), you can probably ignore missing RACF profiles for new product data sets. If you know the HLQs for new product data sets, you can easily add a RACF generic profile to protect the product data sets.

4.11 OS/390 UNIX System Services considerations

You need some planning before using OMVS for productive purposes. These plans should include:

- ▶ A way to keep track of the next UID and GID available for assignment. You could use random numbers, or employee numbers, or something similar, but most UNIX people prefer simple sequential numbers. OS/390 does not automatically supply the next sequential UID number when you define a new user.
- ▶ Some of the UNIX system services publications recommend defining a separate HFS for each user, and using the *automount* function to mount the appropriate HFS when a user logs onto OMVS. For various reasons, this is seldom done by typical AD users.
- ▶ Whether or not you use separate HFSs for each user, you should define at least one local HFS (not on a system volume!) for /u. You can place your users' files in this, or create more file systems with mount points in /u or some other place.
- ▶ Decide how to define *root* users. You can simply define one or more users with UID=0 (remembering that P390 and OPEN1 - OPEN3 are already defined this way), or you can permit certain users to **su** to root. There are RACF controls that can limit the ability of users to **su** to root. (See "Authority for administrators" on page 95.)

It is important to plan for UNIX use early; once your users start working with it, it is difficult to change file systems and controls.



Routine administration and use

This chapter contains many small sections, mostly with “how to” information. The sections are in no particular order.

5.1 How to start the P/390 subsystem

With a P/390 or Integrated Server, you must start the P/390 support programs before you IPL OS/390. The normal way to do this is to click the IPL icon in the P/90 main menu. (You may need to *open* the single P/390 icon on the OS/2 desktop in order to obtain the P/390 icon window. The underlying program will start the various P/390 support programs, if they are not already started, before starting the S/390 IPL sequence.

There is an important point here: the IPL icon will not restart the P/390 support programs if they are already started. If you are restarting OS/390 because something failed—it crashed or something appeared to hang—then the P/390 support programs might not be in a proper status for reuse. We suggest you click the END P/390 icon before you re-IPL. This will cause all the P/390 support programs to end, and IPL P/390 icon will then start fresh copies.

You can start the P/390 subsystem with line commands in an OS/2 window instead of using the icons. This is useful if you want to use a different IPLPARM value than the one set in the DEVMAP. For example:

```
IPL A80
IPL A80 /PARM 0A82CS
IPL NOIPL                                (this starts the P/390 subsystem, but does not IPL
                                         an S/390 operating system)
AWSSEND                                  (stops the P/390 subsystem)
```

5.2 Understanding addresses and device numbers

In the good old S/360 days, a device address such as 583 meant channel 5, control unit 8, device 3 and this was the *address* of the device. No S/390 machines work this way today, but the old terminology is often used. This terminology is technically incorrect, but is so widely used that you must understand it. If someone tells you to “IPL from address 583” they really mean to IPL from *device number* 583.

For all S/390 machines (*excluding P/390-based system*) the hardware address of a device will be something like chpid 05, CU 3, UA 7. This is similar to the old S/390 addresses; a chpid is a channel, the CU number is a control unit address, and the UA is a unit address (that can normally range up to x'FF'). However, the software never sees this hardware address. The S/390 I/O subsystem, using the IOCDS data, converts the chpid/CU/UA “hardware address” into a *device number*. The operating system uses the device number to connect to I/O devices. The operating system and applications never see the actual hardware addresses (the chpid, CU, and UA numbers).¹

In general, the hardware address and the device number appear unrelated, although the low-order digit is often the same. For example, device number 583 (as seen by software) might have connections through chpids 01, 27, 3F, and 7B, to CU 3 and use UA 3.

The DEVMAP on a P/390-based device (including R/390s and S/390 Integrated Servers) specifies a *device number* (although the documentation usually refers to it as an *address*). There is no equivalent to chpid, CU, and UA hardware addresses on these machines.

The DEVMAP on a MP3000 specifies a Unit Address (UA), and any device that is specified in the DEVMAP is always on chpid FC. The active IOCDS converts this to a device number, and this is used by the software.

¹ The HCD utility is a special case that does work with chpid, CU, and UA addresses.

The *system definition* file of a FLEX-ES EFS system defines the addresses (device numbers) used. There is no equivalent to chpid, CU, and UA *hardware* addresses, although software equivalents are defined.

You must be aware that many S/390 people say *address* when they mean *device number*. These same people will usually say something like “*real hardware address*” when they are talking about chpids, CUs, and UAs. Throughout this document we sometimes use device address and device number interchangeably. If a discussion is about *real hardware addresses*, it will clearly indicate this.

5.3 Understanding IPL parameters

OS/390 provides great flexibility in determining the configuration its resources. The following discussion provides a brief summary of the components used by the AD system during IPL to manage this flexibility. There are other ways to arrange IPL controls and not all options are mentioned here.

The starting point is an IPL parameter, which can be provided in a number of places:

- ▶ The Update System Environment panel of a P/390 or R/390 configurator
- ▶ A command line in an OS/2 window of a P/390-based machine.
- ▶ The Load panel of the Support Element (or HMC) of an MP3000
- ▶ The activation profiles (automatic IPL) of an MP3000
- ▶ The IPL command line of an EFS system

The IPL parameter is eight characters long:

AAAAXPN

AAAA is the device number (address) of the device containing the IODF and (normally) SYS1.IPLPARM

XX is two characters that are used as a suffix for the constant ‘LOAD’ and this name (LOADxx) specifies a member name in SYS1.IPLPARM

P is a character than can produce additional prompts during IPL. This is sometimes referenced as the IMSI parameter

N is a character used to load an alternate nucleus

The specific LOADxx members provided with the AD system are described in “Detailed IPL parameter controls” on page 29 and “Detailed IPL parameter controls” on page 57 and will not be repeated here. For this more general discussion, we will use a CS parameter for examples. A typical IPL parameter might be:

0A82CS

This indicates the current IODEF data set is located on device number (address) 0A82 and to use member LOADCS in SYS1.IPLPARM. The blank following 0A82CS indicates the default is being used for the IMSI (Initialization Message Suppression Indicator) function. The default is to suppress some informational messages during IPL and not to prompt the operator for additional information.

The AD R10 system contains the following in member LOADCS of SYS1.IPLPARM:

```
IODF      01 SYS1
SYSCAT    0S39M1113CCATALOG.0S390.MASTER
SYSPARM   CS
IEASYM    00
PARMLIB   SYS1.ADCD10.PARMLIB                0S39RA
PARMLIB   SYS1.PARMLIB                       0S39M1
NUCLEUS   1
SYSPLEX   ADCDPL
```

```
123456789012345678901234567890123456789012345678901234567890
0           1           2           3           4           5           6
```

(The last two lines show the columns used and are not present in the SYS1.IPLPARM member.) The format is a bit odd, and certain columns must be used, as shown. Important data includes:

- ▶ The IODF statement, which names the current IODF data set as SYS1.IODF00. This is required to be on the disk at the address specified by the first four characters of the IPL parameter.
- ▶ The SYSCAT statement, which defines the master catalog as an ICF catalog named CATALOG.OS390.MASTER on volume OS39M1.
- ▶ The SYSPARM statement, which points to IEASYSCS in PARMLIB. This is a key link in the control chain that determines the overall IPL environment.
- ▶ The PARMLIB statement(s), which specify the data sets that make up the logical PARMLIB and the order in which the data sets are concatenated. The volsers are also specified, although the catalog will be searched, if necessary, to find them.

Continuing with this example, the IEASYSCS member in PARMLIB is the next link in the control chain. It contains various parameters, and points to many other PARMLIB members containing configuration information for components and subsystems of OS/390.

You can verify your currently active IODF name with the OS/390 command D IOS,CONFIG. This will produce (on the OS/390 console) a response like this:

```
ACTIVE IODF DATASET = SYS1.IODF01
CONFIGURATION ID = CBIPO   EDT = 00
```

Several characters can be used for the seventh IPL parameter character. The two most common are:

```
T      (Prompts the operator to Specify System Parameters)
A      (Prompts the operator for Specify System Parameters and for the Master Catalog
name)
```

For an example of how to use the T prompt, see 5.4, “How to provide a manual IPL” on page 105.

The eighth character, for an alternate nucleus, is not normally used with the AD systems.

The general flow of the IPL parameters is indicated in Figure 5-1 on page 105.

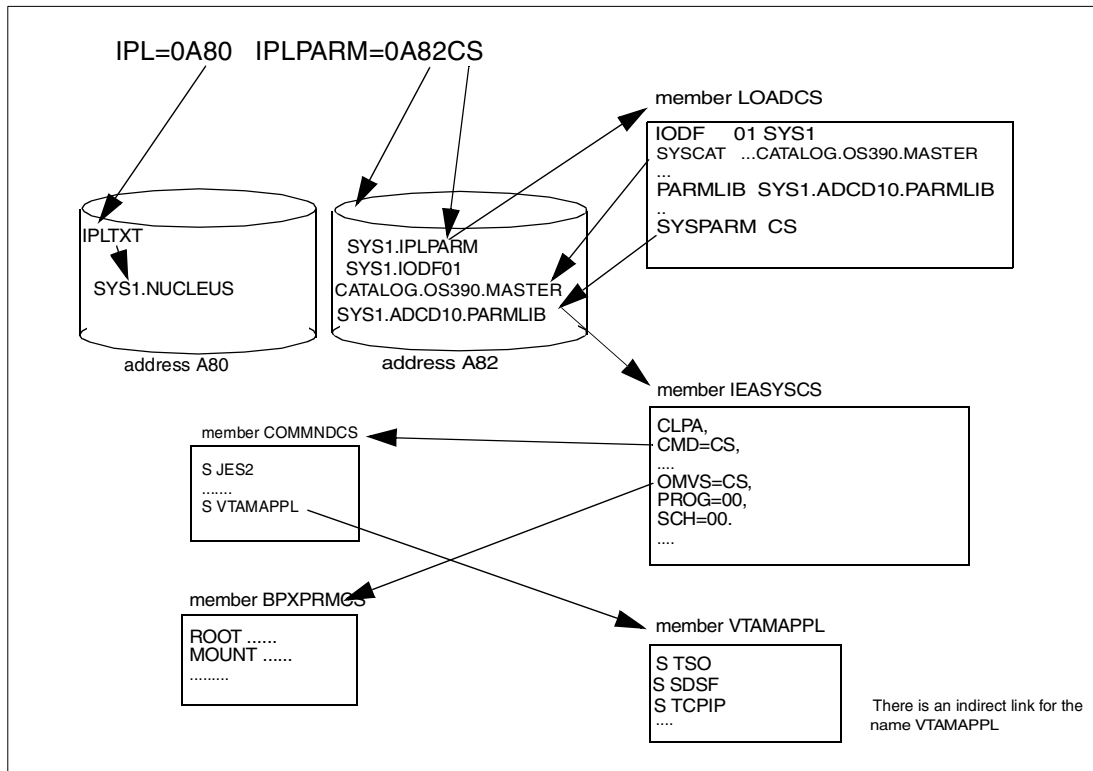


Figure 5-1 General IPL parameter flow

In this illustration, the IPL parameter CS points to member LOADCS in SYS1.IPLPARAM. The LOADCS member names the PARMLIB(s) and points to member IEASYSCS. Member IEASYSCS points to many other members of PARMLIB; only a few are shown here. In this example, IEASYSCS points to COMMNDCS. This member issues several START commands and starts a secondary command script program (VTAMAPPL) that (indirectly) points to member VTAMAPPL in PARMLIB. The VTAMAPPL member then issues more START commands.

It is common to use the same suffix (CS in this illustration) throughout the process; this helps to reduce the confusion level. However, this is not required. LOADCS could contain SYSPARM XY, and IEASYSCS might contain CMD=AA, and so forth. The general process is very flexible and can be extended to many more controls than shown here.

Much greater detail can be found in the following manuals:

OS390 Initialization and Tuning Guide

OS390 Initialization and Tuning Reference

OS390 System Commands

5.4 How to provide a manual IPL

Specific information for IPLing the AD system is presented in 2.4.1, “Detailed IPL parameter controls” on page 29 and 3.4.1, “Detailed IPL parameter controls” on page 57.

In general, smaller OS/390 systems (such as might be used for AD systems) are set up for either an *automatic IPL* or a *manual IPL*. The intention of an automatic IPL is that no console interaction is required until batch, TSO, TCP/IP, and so forth, are ready for use. That is, you will not need to respond to normal OS/390 startup messages. The various IPL parameters provided with the AD system all trigger automatic IPL scripts.

This may not always work; unexpected messages during IPL may occur and their context may not be obvious if you are not providing the normal startup responses. A manual IPL will require responses to several messages during the IPL sequence.² There are different degrees of automatic startup—for example, your IPL process might start TSO automatically, but require you to start CICS manually.

An automatic IPL is more convenient, but can present a problem if you want to intervene during IPL to bypass a problem. The automatic IPL may not let you intervene at the right moment. A manual IPL is less convenient, but offers more opportunity to bypass errors or take alternate actions. The setup for automatic IPL is done by a path created in the LOADxx member (SYS1.IPLPARM), the COMMNDxx member, and VTAMAPPL-like members of the PARMLIB. You can make an IPL sequence more manual simply by removing commands from COMMNDxx or the appropriate VTAMAPPL-type member. (See 5.5, “What is VTAMAPPL” on page 108 for an introduction to this program.)

The key link for the automatic commands is the COMMNDxx member that is chosen by whatever IEASYSxx member is specified in LOADxx. To produce a manual IPL, you might create member COMMNDZZ³ in PARMLIB like this:

```
COM='S JES2'  
COM='S VLF,SUB=MSTR'  
COM='S DLF,SUB=MSTR'  
COM='S LLA,SUB=MSTR'  
COM='S EZAZSSI,P=NODENAME'  
COM='S IRRDPTAB'
```

These commands start JES2 (without specifying the required start parameters), start some of the look-aside functions (you normally want these, and there are seldom startup problems with them), starts EZAZSSI (a small preparation job for later use of TCP/IP), and IRRDPTAB (which enables some RACF functions). You could omit some of these start commands and perform them manually, but this list is a reasonable starting point. Note that there is no command to start VTAMAPPL; this means that none of the VTAMAPPL-type scripts will be run automatically.

To start a manual IPL, you could then add the letter ‘T’ as the seventh character of the IPL parameter. For example, if you would normally use IPL parameter 0A82DB, you would now use 0A82DBT. This will force the following message as the first (or almost first) message from OS/390:

```
IEA101A SPECIFY SYSTEM PARAMETERS
```

The following are examples of appropriate replies:

```
REPLY 00,'CMD=ZZ' (example A)  
REPLY 00,'SYSP=CS,CMD=ZZ' (example B)  
REPLY 00,'SYSP=00,CLPA,CMD=ZZ' (example C)
```

² The terms *automatic IPL* and *manual IPL* are not standard terms.

³ There is nothing special about the ZZ suffix; it's simply a suffix not used for existing COMMNDxx members in the AD system.

These specify overrides for the selected IEASYSxx member:

- ▶ Example A says to use the IEASYSxx member specified in the LOADxx member. (Remember, the LOADxx suffix is specified in your IPL parameter.) However, do not use the COMMNDxx member specified in IEASYSxx; instead use the COMMNDZZ member.
- ▶ Example B says to ignore whatever IEASYSxx suffix was supplied by the LOADxx member you specified; instead use IEASYSCS. Furthermore, ignore the COMMNDxx suffix specified in IEASYSCS and use suffix ZZ (which specifies COMMNDZZ) instead.
- ▶ Example C says to use IEASYS00 (ignoring what was specified in your LOADxx parameters), use COMMNDZZ (ignoring what was specified in IEASYS00), and to perform a CLPA function. CLPA causes the Pageable Link Pack Area paging data set to be reloaded.

Example A is the most typical reply. It will cause all the normal parameters associated with your LOADxx selection to be used, except it will use PARMLIB member COMMNDZZ instead of whatever COMMNDxx member it would have normally used. The COMMNDZZ member shown above starts JES2 (omitting the required startup parameter) and starts a few other minor functions. It does not start major subsystems such as VTAM, TSO, and so forth. You will need to issue OS/390 operator commands to start these.

When JES2 starts, you will see the message:

```
01 $HASP426 SPECIFY OPTIONS - JES2 OS ---
```

The normal responses are one of the following:

```
1,NOREQ                (Example A)
1,COLD                  (Example B)
```

The NOREQ reply indicates a normal *warm start* for JES2, and normal JES2 processing should begin. This is the most common reply to this message. The COLD reply indicates that JES2 should reformat its spool data set(s); this will delete all the jobs and output currently held by JES2. A JES2 cold start is rarely done. The most common reason for a cold start is that the spool area is completely full and you do not know how to selectively remove enough jobs to free a reasonable amount of space. After a COLD start, you must enter the command **\$S** (two characters, no leading number) to tell JES2 to start its normal processing. (In the reply examples, the leading "1" indicates reply number 1; you should use whatever reply number your system provides, of course.)

Once JES2 is started, you need to issue operator commands to start whatever started tasks you need. The names (such as VTAM, TSO, and so forth) are the names of members in SYS1.PROCLIB. The release 10 AD system uses the names shown here; your system might use different names and you will need to know the proper names. In a reasonably simple situation, you might issue the following operator commands:

```
S VTAM                (wait for it to start before issuing the next command)
S TSO
S SDSF                (recent AD systems use this form of SDSF control)
S TCPIP
```

If you start TSO before VTAM is ready, you will receive the message:

```
02 IKT003D TCAS UNABLE TO ACCEPT LOGONS
REASON CODE=0092 REPLY 'RETRY' or 'TERM'
```

You must wait until you see the message:

```
IST020I VTAM INITIALIZATION COMPLETE FOR VxRy
```

and then reply:

2,RETRY (you can use lower case)

If you reply RETRY too soon, OS/390 will simply issue the IKT003D message again. Your message number may not be 2, of course; use whatever number is assigned for the reply.

During the IPL you may encounter several red error messages from various system components. These will often disappear when other components complete their initialization. Do not worry about the messages unless they are not resolved by the time all the IPL-related startup is complete.

You can still use one of the SHUTDOWN scripts in PARMLIB, even if you controlled your own system startup. Alternately, you could issue your own commands to stop system components. For example;

```
P TSO (P is the OS/390 STOP command)
P SDSF
C NFSC (NFSC does not monitor STOP commands)
P TCPIP
#STOP (for a RACF function)
F DLF,MODE=DRAIN
P LLA
P VLF
P DLF
Z NET,QUICK (stops VTAM)
```

5.5 What is VTAMAPPL

In current AD releases, VTAMAPPL is the name of three things:

- ▶ There is a program (load module) named VTAMAPPL, in one of the system libraries.
- ▶ There is a member (procedure) in SYS1.PROCLIB named VTAMAPPL.
- ▶ There is a member in PARMLIB named VTAMAPPL.

The VTAMAPPL program reads an input file containing a *script* of system commands (such as S TCPIP) and issues these commands, one at a time, through an internal interface. It has two special characteristics:

- ▶ It waits until VTAM is active before it starts submitting commands. (The program was originally created to solve a timing problem in starting TSO; if TSO was started before VTAM completed initialization, the operator needed to reply to an additional message.)
- ▶ It accepts **pause** commands in the input script, allowing the script writer to have some control over the timing of the commands.

In earlier releases, the VTAMAPPL procedure executed the VTAMAPPL program and pointed to (with a DD statement) the VTAMAPPL member in PARMLIB. A statement in the COMMNDxx member of PARMLIB during IPL started the VTAMAPPL procedure.

Recent releases have expanded this. There is, for example, a VTAMDB procedure (in SYS1.PROCLIB) that executes the VTAMAPPL program and points to a VTAMDB member in PARMLIB for its command script. (There is no requirement for the PARMLIB member to have the same name as the PROCLIB member, but doing so saves some confusion.) The SHUTDOWN procedure is another example of using the VTAMAPPL program; it issues a series of STOP, MODIFY and CANCEL commands to bring down the system.

You can use the VTAMAPPL program to build your own scripts of system commands.

5.6 Basic introduction to IODFs

The OS/390 Hardware Configuration Definition (HCD) function is an ISPF application that allows you to define hardware and esoteric definitions for OS/390 systems. It replaces the MVS Configuration Program (MVSCP) and SYSGEN/IOGENs from earlier MVS systems.

HCD keeps its definitions in data sets known as I/O Definition Files (IODFs). An IODF is normally read only during IPL. With the proper setup, a new IODF can be enabled by the Dynamic Reconfiguration function -- permitting hardware address changes (and esoteric device name changes) without an IPL. This is done, in part, by dynamically replacing the IOCDS.⁴ A P/390-based system or Netfinity/EFS system does not use IOCDS functions; consequently the Dynamic Reconfiguration function provided through IODFs is not available for these systems. You must re-IPL one of these to activate a new IODF.

A brief overview of IODF and HCD usage may be helpful.

Older releases of MVS changed their I/O configurations through SYSGENs, IOGENs, MVSCPs, and similar processes. With current versions, configurations are changed with the HCD program, which produces IODF files. An IODF file contains, among other things, the information needed to build MVS UCBs.⁵ HCD can also produce IOCDS files, but these are not used with P/390s, R/390s, or Netfinity/EFS systems. They are used with MP3000s and larger machines.

A full description of HCD and IODFs is beyond this document. The following is a very brief review of pertinent points:

- ▶ There are two types of IODF data sets: *production* IODFs and *work* IODFs. You can have many production and/or work IODF data sets; one of the production IODFs will be the *active* IODF that was used for your current IPL.
- ▶ Only a production IODF can be used by OS/390 at IPL time.
- ▶ An IODF can contain multiple system configurations. This is useful in a large sysplex mainframe environment, but not useful in a typical AD environment. The IODF supplied with the AD CD-ROM contains one configuration named CBIPO.
- ▶ You can display a production IODF (using option 1 on the main HCD menu), but you cannot alter it. You can only alter work IODF data sets.
- ▶ If you have a production IODF and you need to modify it, you must do the following:
 - Create a work IODF from the production IODF.
 - Alter it as needed, using HCD functions.
 - Create a new production IODF from the work IODF.
 - IPL using the new production IODF.⁶
- ▶ After a few cycles of IODF updates, you will have a number of old production IODF data sets. You can delete these, using another HCD option. The key point is that you cannot change a *production* IODF and you cannot test a *work* IODF; you must create a new production IODF for every change.
- ▶ You can write a new production IODF over an existing production IODF, even the active production IODF. Overwriting the only usable production IODF may not be a good idea. If your new IODF is unusable for some reason, you may not be able to IPL again.

⁴ The IOCDS is a file (stored in the Service Element, in OS/2) used by the S/390 hardware to define and manage channel, control unit, and device paths and mapping.

⁵ A Unit Control Block (UCB) is the operating system's most basic control block structure for dealing with I/O devices.

⁶ There is an option to dynamically switch to the new IODF without an IPL. However, the IOCDS and IODF must exactly match (and have a matching token) to do this. Systems that do not use an IOCDS cannot dynamically activate a new IODF. In general, AD systems will not use this function.

The normal way to create a work IODF from a production IODF is to use option 1 of the main HCD menu to view the production IODF. If you then try to alter the production IODF, HCD will offer to create a work IODF for you and ask for a name for the work copy. You should devise a naming convention: if your production IODF (that you are viewing with HCD option 1) is SYS1.IODFnn, you might name the work IODF created from it SYS1.IODFnn.WORK.

When you create a new production IODF from SYS1.IODFnn.WORK, you might name it nn+1. You can alter a work IODF any number of times and create any number of production IODFs from it. In this case you need to devise your own numbering scheme. Keep the established pattern of names (SYS1.IODFnn and SYS1.IODFnn.WORK) and remember to place all IODF data sets on the same volume.⁷ Remember to delete old IODFs when they are not needed; after a while, too many IODF data sets can become confusing. The HCD utility has a function to delete IODFs.

HCD is not an easy program to use without training or some help. It is definitely not intuitive. Many AD system owners avoid changing the IODFs that are provided with the AD system because they cannot take the time to learn HCD. An alternative is to change the hardware definitions (IOCDs, DEVMAPs, system definitions for Netfinity/EFS) to match the existing IODF(s). This is briefly discussed in 1.6, “I/O configuration for OS/390” on page 15.

5.7 How to work with PARMLIB

The partitioned data set SYS1.PARMLIB (PARaMeter LIBrary) was the primary source of information used when configuring and customizing earlier OS/390 systems. You can now have *multiple* PARMLIBs, by naming them in the LOADxx member of SYS1.IPLPARM. The AD systems normally have two PARMLIBs. (See “IPL parameters” on page 28 and “IPL parameters” on page 56 for more about the current AD systems.) We now tend to use the term *PARMLIB* to refer to SYS1.PARMLIB and whatever other libraries, if any, that have been concatenated with it.

The order of concatenation is extremely important, because the libraries are searched in order and the search stops whenever the desired member name is found. If member X exists in several of the concatenated libraries, only the first X (in the first library in the search order that contains an X member) is ever used. See “How to use ISRDDN” on page 226 for one technique for working with concatenated PARMLIBs.

PARMLIB contains many members and each member (or one of a set of similarly named members) is responsible for the configuration of a feature or component, subsystem or program product. This configuration process takes place at IPL time, or when some components or programs are started, or, in some cases, may be driven by an operator command from an OS/390 console.

PARMLIB is used heavily during IPL, starting very early in the IPL sequence and continuing through most parts of the IPL. Member IEASYSxx is the starting point for configuration. By default, IEASYS00 is used. However, you will almost always specify a particular IEASYSxx member via your IPL parameter and a LOADxx member in SYS1.IPLPARM. In our examples here, we assume you use IEASYS00. IEASYSxx contains two types of parameters:

- ▶ Controls for some part of the OS/390 configuration
- ▶ References to other members of SYS1.PARMLIB

The IEASYS00 member supplied with the release 10 AD system is as follows:

```
CLOCK=00,                SELECT CLOCK00
```

⁷ There is no absolute requirement that they all be on the same volume, but you will need to move SYS1.IPLPARM, the master catalog, and possibly some PARMLIBs if you change volumes.

CLPA,	
CMB=(UNITR,COMM,GRAPH,CHRDR),	ADDITIONAL CMB ENTRIES
CMD=CS,	
CON=(00,NOJES3),	SELECT CONSOL00
COUPLE=00,	ALL LOCAL
CSA=(3000,20000),	CSA RANGE
DIAG=00,	SELECT DIAG00, DIAGNOSTIC COMMANDS
DUMP=DASD,	PLACE SVC DUMPS ON DASD DEVICES
FIX=00,	SELECT IEAFIX00, FIX MODULES SPECIFIED
GRS=NONE,	NO COORDINATION OF GRS REQUESTS
ICS=00,	SELECT IEAICS00, CNTL SPECS FOR SRM
IPS=00,	SELECT IEAIPS00, PERF SPECS FOR SRM
LNKAUTH=LNKLST,	AUTHORIZE LNKLST00, APFTAB IS ALTERNATE
LOGCLS=L,	WILL NOT BE PRINTED BY DEFAULT
LOGLMT=999999,	MAX WTL MESSAGES QUEUED, MUST BE 6 DIGITS
LOGREC=SYS1.LOGREC,	ERROR RECORDING
LPA=00,	SELECT LPALST00
MAXUSER=250,	SYS TASKS PLUS INITS PLUS TSOUSERS
MLPA=00,	SELECT IEALPA00, MLPA PARAMETERS
MSTRJCL=00,	SELECT MSTJCLEX, MASTER JCL
OMVS=CS,	SELECT BPXPRMCS
OPI=YES,	ALLOW OPERATOR OVERRIDE TO IEASYS00
OPT=00,	SELECT IEAOPT00, SRM TUNING PARAMETERS
PAGE=(SYS1.PLPA.PAGE,	
SYS1.COMMON.PAGE,	
SYS1.LOCAL.PAGE,L),	
PAK=00,	SELECT IEAPAK00
PLEXCFG=ANY,	
PROG=00,	SELECT PROG00, DYNAMIC APF LIST
REAL=128,	ALLOWS 2 64K OR 1 128K JOB TO RUN V=R
RSU=0,	NO RECONFIG STORAGE UNITS
RSVNONR=5,	RESERVED ASVT ENTRIES
RSVSTRT=5,	RESERVED ASVT ENTRIES
SCH=00,	SELECT SCHED00
SMF=00,	SELECT SMFPRM00, SMF PARAMETERS
SQA=(15,64),	SQA SIZE APPROX 640K
SSN=(02,00),	SELECT IEFSSN00, SUBSYSTEM NAMES
SYSNAME=P390,	SYSTEM NAME
PROD=00,	SELECT IFAPRD01
SVC=00,	SELECT IEASVC00, USER SVCS
VAL=00,	SELECT VATLST00
VIODSN=SYS1.STGINDEX,	VIO DS
VRREGN=64	DEFAULT REAL-STORAGE REGION SIZE

Members in SYS1.PARMLIB have predefined names (up to six characters) followed by an installation-defined suffix of 2 characters. The base portion of the name sometimes indicates the function of the member, for example:

IEASYS	System Control member
IEAFIX	Lists of modules fixed in storage
CONSOL	Definitions for the OS/390 Consoles
LNKLST	Lists of libraries used for OS/390 standard program search

The installation-defined suffix can be any two alphanumeric characters; for example, specific instances of the members just listed might be named:

IEASYSCS	The control member
IEAFIX00	List of modules fixed in storage
CONSOLZZ	Definition for the OS/390 Consoles
LNKLST01	List of libraries used for OS/390 standard program fetch

The supplied PARMLIB also contains many versions of the various members (that is, they have suffixes such as 01, 99, MI, AB, ...). These are not used during normal operation but illustrate different configurations that could be specified. Which version of a PARMLIB member is used during IPL or component startup can be determined several ways:

- ▶ A line may exist in the IEASYSxx used for this IPL. For example the line:

```
LNK=01,
```

will cause member LNKLIST01 to be used to determine the system link list contents. (Note the comma. All the lines in IEASYSxx, except the last data line, should have commas.)

- ▶ The system operator can reply at the master console to message IEA101I SPECIFY SYSTEM PARAMETERS. For example,

```
R 00, LNK=A2
```

to message IEA101I will cause LNKLISTA2 to be used. (See “How to provide a manual IPL” on page 105 for a specific example.)

- ▶ For some functions, OS/390 will use the 00 version of a member if no specification is provided in IEASYSxx or by the operator. However, for other functions (such as IECIOSxx for MIH), the member suffix must be listed in IEASYSxx or the function is not used.
- ▶ Note that an operator reply to SPECIFY SYSTEM PARAMETERS will override a specification in the controlling member IEASYSxx. This principle is used to provide fallback when making changes to PARMLIB.

Making changes to PARMLIB members has associated dangers. A mistake in a critical parameter can produce an OS/390 system that will not IPL, or will be unable to start a critical component. If the system will not start, you are not able to use TSO to correct your mistake. Every MVS systems administrator has, at one time or another, made his system unworkable due to an error in PARMLIB. You can minimize this danger by implementing changes in a controlled manner with a guaranteed back out (or fallback). There are several ways to do this:

- ▶ You might have several OS/390 systems operational, with shared DASD, but without shared PARMLIBs. You can use another OS/390 (IPLing it, if necessary) to fix the broken OS/390. This is the most common solution if you have sufficient disk space. P/390-based users are unlikely to have this option, because they typically do not have sufficient disk space for multiple OS/390s.
- ▶ You can back up the volume containing the PARMLIB and perform a stand-alone restore if your system breaks. You always have this option, but it is slow and might be awkward.
- ▶ You might have a stand-alone program (on the S/390, or on OS/2, where appropriate) that can directly edit PDS members. This can be very convenient, and there are vendor products that do this.
- ▶ You might change the PARMLIB member and dynamically activate the change. If the dynamic activation fails, you can correct the member and try again since your system is still running and TSO is still available. There are some risks with this method,⁸ and not all PARMLIB changes can be dynamically activated.
- ▶ You can ensure that there are backup members in PARMLIB that you can use if your altered member fails. This is the most common solution if you do not have a second OS/390 system available. This method is not difficult, but it takes a little discipline and thought. We describe this method here.

⁸ There are undoubtedly some PARMLIB combinations and errors that would render a system useless after dynamically activating the changes.

The general method is quite simple. You must ensure that a working version of a member is in PARMLIB *and you must remember the suffix you gave the working version*. This last requirement is the most difficult, and is made easier if you always use a consistent suffix for the backup member. We arbitrarily use suffix 99.

Suppose I need to change member SMFPRM00 (assuming this is the member that is pointed to by all my IEASYSxx members). I first copy SMFPRM00 to SMFPRM99. I now have a copy in SMFPRM99 that is known to work (assuming my current system is working, of course). You need not have the latest SMFPRMxx version in SMFPRM99, you just need one that is known to work.

If you later IPL and the IPL fails due to the changes you made in SMFPRM00, you can perform a manual IPL (see “How to provide a manual IPL” on page 105) and use `REPLY 00,'SMF=99'` when asked to `SPECIFY SYSTEM PARAMETERS`. You can then correct the problem in the normal operation member (SMFPRM00) and retry the test or completely remove the change by copying the backup version (SMFPRM99) over the normal operation member.

All members of PARMLIB except for IEASYS00 use the principle that the selected member provides all parameters for that function. However, IEASYSxx members use a merging technique. If an alternative IEASYSxx member is specified by operator override in reply to message IEA1011:

```
R 00,SYSP=xx
```

then the parameters in IEASYSxx are merged with the parameters specified in IEASYS00. The parameters in IEASYSxx override a similar parameter in IEASYS00. Therefore, the parameters are resolved in the following order:

- ▶ Operator override, in response to `SPECIFY SYSTEM PARAMETERS`.
- ▶ The IEASYSxx member specified by `REPLY 00,'SYSP=xx'` to the `SPECIFY SYSTEM PARAMETERS` message.
- ▶ IEASYS00.
- ▶ System defaults if no parameter is supplied.

5.8 Commonly modified PARMLIB members

The following are the PARMLIB members that you are likely to change and some of the reasons you might change them:

- ▶ **COMMNDxx** contains MVS operator commands executed at IPL time. You can add your own commands, for example, to automatically start your databases or submit a job to perform disk space cleanup. The availability of the VTAMAPPL program has greatly reduced the use of COMMNDxx members. The VTAMAPPL program is superior because it permits you to insert **pause** statements to control the timing of startup commands. COMMNDxx would normally contain a command to **S VTAMAPPL**.
- ▶ **CONSOLxx** contains the definitions for the OS/390 system consoles. You might change this to add additional operator consoles.
- ▶ **IEAICSxx** and **IEAIPSxx** contain internal tuning definitions. We suggest you do not change these unless you understand the parameters or are following specific instructions.
- ▶ **IEASVCxx** contains definitions for user SuperVisor Calls (SVCs). Various program products (from IBM or other vendors) may require user SVC definitions. The products will contain specific instructions for what is needed.

- ▶ **IEASYSxx** contains parameters and pointers to parameters for the startup environment for OS/390. This chapter and the Initialization and Tuning manuals provide information about this member and the reasons that may require you to change it. Never make a change without checking the Initialization and Tuning reference manual.
- ▶ **IEFSSNxx** contains definitions of sub-systems that are to be added to your OS/390 system. Some IBM and non-IBM products will require additions to this member. Many products now perform this function dynamically at startup and do not require additions to this member.
- ▶ **IGDDFPKG** contains control statements for program product usage. Change instructions are provided with the documentation for program products that use these controls. Note that this member does not use suffixing.
- ▶ **IKJTSoxx** contains a list of authorized commands and programs that can be executed under TSO. Do not change these unless you understand what you are doing or are following specific instructions. Many vendor and informal TSO commands require additions to this list. However, every addition is a potential security or integrity hole.
- ▶ **LNKLSTxx** is normally not used on current systems. It has been replaced by PROGxx. If you have instructions to add a library to your LNKLST, you should add it to the LNKLST section of PROGxx.
- ▶ **LPALSTxx** contains a list of libraries whose members are used when constructing the OS/390 pageable link pack area (PLPA). Some products will include optional steps for adding libraries to this member. Placing modules in PLPA is a performance option, and is rarely a required function (except for some system libraries). The AD system is provided with reasonable LPALST members.

You may receive advice about adding many libraries or modules to PLPA.⁹ You need to manage the size of PLPA, especially the “below the line” size. A discussion of this is beyond the scope of this book, but we suggest you consult with someone familiar with this area before making significant changes. A significant change might be one that adds 500 KB or more below the 16 MB line.

- ▶ **MPFLSTxx** contains a list of messages that are suppressed from display on the OS/390 consoles. The messages are still displayed on the system log (SYSLOG). You may wish to add or remove messages from this list. Many AD system users remove \$HASP395, for example, so they will see Job Ended messages for batch jobs.
- ▶ **MSTJCLxx** contains the JCL used by the OS/390 master address space, and is often known as the *Master JCL*. Generally you will not need to change this member. The name Master Address Space is slightly misleading. It mostly deals with internal functions related to starting OS/390
- ▶ **PROGxx** contains two lists of libraries: (1) a list of libraries that are authorized under OS/390 to use restricted facilities, and (2) the Link List (LNKLST) libraries. Many IBM and non-IBM products will require you to add library names to this member, in one or both sections. Auditors may question the need for you to have libraries in the list, especially to the authorized library list.¹⁰ The LNKLST provides a list of libraries used to construct the normal OS/390 program search function; that is, any program from a library in this list is available to any started task, batch job or TSO session without special processing such as JOB or STEP libraries. Adding to and deleting from this PARMLIB member is a common occurrence in most installations.
- ▶ **SCHEDxx** provides special features to programs defined in the member. A few products may require you to add entries for selected programs. Be careful because the special features also incur special dangers (such as bypassing data set integrity).

⁹ A module must have specific attributes before it can be added to PLPA. In particular, it must be reentrant.

¹⁰ The AD system (in the IEASYSxx members) uses the LNKAUTH=LNKLST option, which makes all the libraries in the LNKLST also authorized. Adding a library to either list in PROGxx will make it an authorized library.

- ▶ **SMFPRMxx** controls System Management Facilities (SMF) operation. You may wish to change this member to start SMF recording, to add data sets for recording, and to change the system wait time (to prevent TSO users from abending with S522 abends).
- ▶ **TSOKEYxx** contains various TSO controls, mostly related to 3270 terminal functions. It also specifies the maximum number of concurrent TSO sessions allowed. The current AD system takes all the defaults for these values, but a TSOKEYxx member (containing only comments) is still required. Your most likely change might be to increase the default (40) number of concurrent TSO sessions allowed.
- ▶ **VATLSTxx** controls the way OS/390 mounts disk volumes and the way allocation is performed to these volumes. If you add volumes to your OS/390 system you should add entries to VATLSTxx so they are managed correctly. The VATLST00 member with the AD system specifies that any volser not listed in VATLST00 will be mounted as PRIVATE. You will probably want to add several PUBLIC or STORAGE volumes to your system, and you will need to update the VATLST entry to indicate these attributes. (See “Adding new disk volumes” on page 82 for more about this.)

5.9 How to handle PLPA page data set full

During IPL, you may see a highlighted (in red) message:

```
ILR005E PLPA PAGE DATA SET FULL, OVERFLOWING TO COMMON
```

This can probably be ignored.¹¹ If you added modules to PLPA, you might want to increase the size of its paging data set. The easy way to do this is:

- ▶ Determine the current size.
- ▶ Define a new, larger paging data set with a different name.
- ▶ Alter the PLPA paging data set name in all the IEASYSxx members you use, to point to the new PLPA paging data set.
- ▶ Re-IPL to use the new PLPA paging data set and then delete the old one.

If you wait until this message works its way to the top (first line) of your console, you can use the command `K E,1` to delete it.

5.10 How to modify a RACF user profile

A common requirement is to add or modify a segment in a RACF user profile. For example, if userid *ogden* exists and we want to add OMVS authorization, we could use the RACF TSO command:

```
ALTUSER ogden OMVS(UID(105) PROGRAM(/bin/sh) HOME(/u/ogden))
```

The ALTUSER command can be run in batch, but this is usually not justified for the small, ad-hoc changes made with this command. The RACF ISPF panels provide alternate ways to make equivalent changes. Remember that you will need SPECIAL authority to use these commands or to submit a job containing the commands. In the AD CD-ROM system, userid P390 has the necessary authority.

Here are typical ALTUSER commands. These use *ogden* as the userid involved, and you would substitute the appropriate userid. ALU is a recognized abbreviation for ALTUSER.

```
ALU ogden N00MVS
ALU ogden TSO(ACCTNUM(ax) PROC(ispfproc) UNIT(SYSDA))
```

¹¹ MVS folklore asserts that paging efficiency may be slightly improved if PLPA overflows into the COMMON paging data set. This happens if the PLPA paging data set is too small for all the libraries or modules specified for PLPA.

```
ALU ogden TSO(MAXSIZE(64000))
ALU ogden REVOKE
ALU ogden RESUME
ALU ogden PASSWORD(tempword)
ALU ogden SPECIAL OPERATIONS
ALU ogden NOSPECIAL
```

The parameter NOOMVS will remove the user's OMVS segment. Otherwise, the first reference to a segment will create the segment. Likewise, NOTSO, NOOPERPARM, NOCICS, and so forth will delete the indicated segments. REVOKE and RESUME are ways to temporarily suspend a user without deleting his identity.

One other command may be needed for existing user profiles. If you define new group profiles, you need to associate appropriate userids with the groups. One way to do this is with the DFLTGRP (default group) operand of an ADDUSER or ALTUSER command, but there is only one default group per userid. A userid can be connected to many group definitions. The command for this is CONNECT; for example:

```
CONNECT ogden GROUP(prod)
```

Designing good group plans and definitions takes some time and skill. We suggest you do not rush into this area. Groups can provide vital administrative mechanisms for large systems, but may not be needed in a small AD environment.

The PASSWORD command contains an option that may be important for user administration:

```
PASSWORD USER(ogden) NOINTERVAL
PASSWORD USER(ogden) INTERVAL(15)
```

This determines how often RACF will force a user to change his password. Remember that only a SPECIAL user can issue these commands.

5.11 How to delete users

There is no standard method for deleting OS/390 users. This section describes a reasonable approach, but you should recognize that there are many ways to go about this. Deleting a user is not as easy as adding a user for these reasons:

- ▶ The user probably owns data sets. Will you simply delete these? Transfer ownership? Inspect each one, to determine if the contents should be saved?
- ▶ The user may be on RACF access lists for various data sets or functions. All these references should be deleted. There might be a production batch job (perhaps run only once each year) that depends on these access lists.
- ▶ The user may own RACF profiles. Should these RACF profiles be automatically deleted? What if they cover data sets not directly owned by the user? What if the profiles are group definitions or definitions for other users or general resource profiles?
- ▶ The userid may be coded in JOB statements, or data sets using the user's HLQ may be in production jobs or in cataloged procedures.
- ▶ The user may be connected to various RACF groups. All these *connects* should be deleted.
- ▶ What do you do about data sets that cannot be easily traced to the user, but that the user effectively owns? For example, the user may have members in procedure libraries, or output in HOLD classes in spool. RACF and HLQ names do not help with this type of data.

In practice, deleting users is often a messy, badly-done task. The basic reason is that you, the system owner/administrator, will not take the time to fully examine and dispose of all the data sets and other artifacts belonging to the userid being removed. If the user has large libraries of JCL or source code or data, it can be quite difficult to determine if any of these must be kept. A very common approach is to REVOKE the user, and then leave all his data sets (and RACF profiles) intact for an extended period. This extended period might be a year or more -- the theory is that, if no one needs the data sets for a long period, then they can be safely deleted. Again, in practice, the data sets may remain on your disks for years (unless they are large enough that the disk space consumed is significant) because no one will take the responsibility for deleting them.

The remainder of this section ignores these issues, and assumes you intend to delete all traces of the userid.

The first step in deleting a user is to issue the RACF TSO command:

```
ALU userid REVOKE
```

This will prevent the user from logging onto the system again. It will not cancel an existing session if he is already logged onto the system and it will not cancel any batch jobs that are already running, so you may need to handle these special cases manually (using OS/390 console CANCEL commands). You must have SPECIAL authority to issue an ALU command. The REVOKE function does not delete any RACF entries or user data sets; it merely prevents the user from using the system.

The second step is to identify the user's data sets; in the most normal situation, this means to identify all data sets with the HLQ equal to the userid being deleted. There are two cases here:

- ▶ Cataloged data sets

These can be found easily with the ISPF 3.4 function, by simply entering the userid in the *Dsname Level* field. You should determine a reasonable disposition for every data set listed. This might mean deleting it, renaming it to another HLQ, copying some or all of the contents to other files, and so forth. If you do not specify a volser operand for option 3.4, it will list all the cataloged data sets for the HLQ specified.

- ▶ Data sets that are not cataloged

There is no easy way to find these, but a normal users should not have any of these. One approach is to use the ISPF 3.4 function, entering the HLQ in the *Dsname Level* field and entering, one by one, the volser of disk volumes that might contain relevant data sets. If you specify a volser operand for option 3.4, it will use the volume VTOC (and not the catalog) to list data set names.

In all cases, you would delete (and uncatalog) or otherwise dispose of all data sets with the HLQ of the departing userid. Once you have deleted all the data sets with the HLQ of the target userid, you can delete the HLQ ALIAS from your master catalog (if you defined aliases for user catalogs, of course):

```
//JOBW JOB 1,P390,MSGCLASS=X
//ALIAS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE userid ALIAS
/*
```

If Open Edition was used, you should delete the appropriate home directory and all its files and subdirectories. You would then use the FIND command (a standard UNIX command) to locate all the files owned by the user and dispose of them.

```
find / -user userid -print (use this command under OMVS)
```

If you are using RACF remote sharing, use the RACLINK LIST command to see if the user has any user ID associations defined. If so, use the RACLINK UNDEFINE command to delete them. You can not delete a user ID that has any associations defined. (The AD CD-ROM system is not set up to use RACF remote sharing; you should be aware if it has been locally implemented.)

A RACF tool, IRRRID00, can be used to locate all uses of a given userid in RACF. (This tool can also be used for other clean-up purposes. See the *Security Server (RACF) Security Administrator's Guide* for more details.) The IRRRID00 program requires an *unloaded* RACF data base as input. The output from the program is a file containing RACF commands necessary to delete the specified userid. The intention is that you would then manually review these RACF commands before using them. The following three jobs could be used:

```
//P3901 JOB 1,P390,MSGCLASS=X
/* RACF DATABASE UNLOAD
//STEP EXEC PGM=IRRDBU00,PARM=NOLOCKINPUT
//INDD1 DD DSN=SYS1.RACF.BKUP,DISP=SHR
//OUTDD DD DSN=SYS1.RACF.FLATFILE,DISP=(NEW,CATLG,DELETE),
// UNIT=3390,SPACE=(CYL,(10)),DCB=(LRECL=4096,RECFM=VB),
// VOL=SER=WORK01
//SYSPRINT DD SYSOUT=A

//P3902 JOB 1,P390,MSGCLASS=X
//CLEANUP EXEC PGM=IRRRID00
//SYSPRINT DD SYSOUT=A
//SYSOUT DD SYSOUT=A
//SORTOUT DD UNIT=SYSALLDA,SPACE=(CYL,(5,5))
//SYSUT1 DD UNIT=SYSALLDA,SPACE=(CYL,(3,5))
//INDD DD DSN=SYS1.RACF.FLATFILE,DISP=OLD
//OUTDD DD DSN=P390.RACF.IRRRID00.CLIST,UNIT=3390,
// SPACE=(TRK,(2,2)),VOL=SER=WORK01
// DISP=(NEW,CATALOG,DELETE),
//SYSIN DD *
MARKW <=====example of a userid
/*

//P3903 JOB 1,P390,MSGCLASS=X
//TSOBAT01 EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUADS DD DSN=SYS1.UADS,DISP=SHR
//SYSLBC DD DSN=SYS1.BROADCAST,DISP=SHR
//SYSTSIN DD DSN=P390.RACF.IRRRID00.CLIST,DISP=OLD
```

The first job unloads the RACF data base into a flat file. You can reuse this file for other purposes, but you should remember that it will not reflect any RACF changes made after the unload job is run. The userid you use to submit this job must have READ access to the RACF data base.

The second job searches the unloaded RACF data base for all uses of userid=MARKW, and will write (in the data set indicated by the OUTDD statement) the necessary RACF commands to delete the userid. Note that these commands are not automatically executed. They are placed in the OUTDD file for your later use. There is an additional option; if your SYSIN statement contains two userids:

```
//SYSIN DD *
MARKW MARTIN
/*
```

the program will generate RACF commands to replace all uses of MARKW (the first userid) with MARTIN (the second userid). No special RACF authority is needed to run this job.

The RACF commands generated by IRRRID00 might look something like this (to delete userid MARKW):

```
PE 'SYS1.**' ID(MARKW) DEL GEN /* DELETING ACCESS TO SYS1.** */
RE MARKW GRO(VSAMDSSET) /* REMOVING FROM GROUP VSAMDSSET */
RE MARKW GRO(SYSCTLG) /* REMOVING FROM GROUP SYSCTLG */
DD 'MARKW.**' /* DELETING GENERIC <ARKW.** */
DU MARKW /* DELETING USER ID MARKW */
```

The third job will execute (in TSO batch mode) all the commands in the SYSTSIN input data set. *You should inspect all the generated commands carefully before executing them.* There is no requirement to use this third job. You could enter the commands, one by one, as TSO commands or use the RACF ISPF panels to execute similar functions. You will need RACF SPECIAL authority to run this job, or to enter the commands yourself.

Another way to find all the uses of a given userid in RACF is with the IRRUT100 program. This can use the active RACF database as input, and produces a simple listing as output. You could use this job to find where your target userid is used in RACF:

```
//P390R JOB 1,P390,MSGCLASS=X
//RACF EXEC PGM=IRRUT100,REGION=4M
//SYSUT1 DD DSN=P390.CROSS.REFER,DISP=(NEW,CATLG,DELETE),
// UNIT=SYSDA,SPACE=(TRK,(10,1))
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
MARKW <=== example of a userid
/END
```

You can use many functions of the RACF ISPF panels to delete various RACF profiles and access list entries. We will not list the panels here, since the paths are fairly obvious. Remember that the functions from the ISPF panels (as well as RACF TSO commands) perform simple functions. For example, DELETEUSER (whether a TSO command or ISPF panel function) will delete the user profile entry in RACF, but will do nothing about other uses of the userid in access lists and so forth. You normally need to execute a number of RACF commands (TSO or ISPF panel functions) to completely delete a user.

5.12 How to reset a password

A system administrator is often asked to reset a user's password. There are two common reasons for this:

- ▶ The user forgot his password (or made too many errors when attempting to change it).
- ▶ The userid has been REVOKED for some reason.

You can use the RACF ISPF panels to reset passwords, but it is easier to use direct RACF TSO commands. There are two relevant TSO commands:

▶ PASSWORD

When used to reset another user's password, the only option is to set the password equal to the user's default group name. The default group name is often SYS1; if the PASSWORD command is used to reset a user's password, the password will probably be SYS1. This has obvious security consequences.

► ALTUSER

You can select the password to assign when you use this command. Furthermore, you can determine whether the password you set will be *expired* or not.

In both cases, the password is automatically marked as *expired*, by default. This means that the user will be forced to select a new password the next time he logs onto the system. With the ALU command, you can also set an unexpired password -- one that the user can use until he changes it for some reason or it expires in due course.

Before resetting a password, we suggest you always use the LISTUSER command to verify that the user definition exists, and determine if the user is REVOKED. For example,

```
LU ogden
USER=OGDEN   NAME=Bill Ogden           OWNER=SYS1
DEFAULT-GROUP=SYS1   PASSDATE=01.187   PASS-INTERVAL= 90
ATTRIBUTES=REVOKED                                     <== note this line
REVOKE DATE=01.108 RESUME DATE=NONE
LAST-ACCESS=01.012/10:03:05
CLASS AUTHORIZATIONS=NONE
INSTALLATION-DATA
NO-MODEL-NAME
LOGON ALLOWED   (DAYS)           (TIME)
```

In this case we could use these commands:

```
ALU ogden RESUME PASS(newpwd)   <== if REVOKED
ALU ogden PASS(newpwd)         <== if not REVOKED
ALU ogden PASS(newpwd) NOEXPIRED <== if not REVOKED

PASSWORD NOINTERVAL USER(ogden) <== if you want this
```

You would need to tell the user the new password you assigned. He will need it to log on, but will be forced to change it immediately to a password of his own selection unless you used the NOEXPIRED option. The PASSWORD NOINTERVAL command will prevent this user's password from ever expiring. Using this is very poor security, but may be appropriate in smaller, closed systems. You need SPECIAL authority to issue these commands, of course.

The PASSWORD command is most useful when a user wants to change his own password. The syntax is odd:

```
PASSWORD PASSWORD(olddpw newpw)
```

The word PASSWORD must be entered twice, once as the command name, and once as a keyword parameter. This command changes the password to *newpw*, which is then a working password until changed sometime in the future. An administrator cannot use this form of the PASSWORD command to target another user; this form applies only to the TSO user issuing the command.

You can also use the RACF ISPF panels to change or reset passwords. The end result is the same as if you used direct commands as shown above. The path to the appropriate RACF ISPF panels is:

```
ISPF Primary Option Menu
  RACF      (select RACF from the primary ISPF menu)
    RACF - Services Option Menu
      User Profiles and Your Own Password
        RACF - User Profile Services
          CHANGE (and enter target userid in the USER field)
```

This should produce the panel shown below, and you would carry on from this point. Remember that whatever password you assign must be changed by the user when he logs onto the system the next time. This same panel, and follow-on panels shown after you press ENTER, can be used to change the same elements as the ALTUSER command.

```

                                RACF - CHANGE USER OGDEN
COMMAND ==>>>

ENTER THE DESIRED CHANGES:

OWNER                ==>>>          user ID or group name

USER NAME            ==>>>

DEFAULT GROUP        ==>>>          Group name

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

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

REVOKE               ==>>>          YES, mm/dd/yy (date) or b1

RESUME               ==>>>          YES, mm/dd/yy (date) or b1

```

Figure 5-2 RACF Change User menu

5.13 How to provide emergency userids

An emergency userid is one that will function when key portions of OS/390 are not working correctly. This is usually due to an incorrect change made to a key subsystem. In practice, the range of problems that can be corrected by using an emergency userid is quite narrow, and usually involves RACF or ISPF problems. Nevertheless, you should maintain at least one emergency userid. There is a certain tradition for using IBMUSER as an emergency userid.

An emergency userid has these characteristics:

- ▶ A complete entry for it exists in SYS1.UADS, including a usable password. (IBMUSER is defined this way in the AD system. The password is IBMUSER; this password (the UADS password) is valid only if RACF is not operational. You can change the password, of course, by using the TSO ACCOUNT command.)
- ▶ The userid is functional without RACF.
- ▶ It has a very minimal logon procedure associated with it.
- ▶ It does not automatically start ISPF, and does not allocate any ISPF data sets (including an ISPF profile data set).
- ▶ It should have RACF SPECIAL and OPERATIONS privileges, and TSO ACCOUNT, JCL, and OPER authorities.
- ▶ It is awkward to use, because it may be difficult to start ISPF from it (since necessary preallocations were not made). (When was the last time you used TSO EDIT to correct a problem?)

We assume your system normally has RACF active. In this case, it is RACF that checks passwords and RACF does not reference any passwords kept in SYS1.UADS. An emergency userid will have, in effect, two passwords: one in RACF and one in SYS1.UADS. Password changes, under normal circumstances, are made only to RACF. To change the SYS1.UADS password, you must use the ACCOUNT command. (See “How to work with SYS1.UADS” on page 122 for more information about ACCOUNT.) In practice, the SYS1.UADS password is probably never changed. To use the emergency userid, when RACF is not functional, you must remember the seldom-used password in SYS1.UADS for that userid.

The P/390 OS/390 CD-ROM systems have IBMUSER set up to function as an emergency userid. You should use the ACCOUNT command to alter the password in SYS1.UADS to one that (a) is a quality password, and (b) you can remember indefinitely. In both CD-ROMS, IBMUSER is set (by the parameters in SYS1.UADS) to use the IKJACCNT logon procedure, which is a minimal procedure that does not allocate ISPF data sets. You might need to specify procedure IKJACCNT in the logon password panel.

The current AD system contains a SYS1.UADS entry only for IBMUSER. Some earlier versions defined a number of other users in SYS1.UADS.

5.14 RACF execute vs. read permission

If a user has only RACF *execute* permission for a program, he can execute that program only in a *clean* environment. If the user has *read* permission for a program, he can execute that program in any environment.

In general, a TSO environment is not a *clean* environment. If a module executes in a clean environment, you cannot subsequently load an *unprotected* program in the same environment. Most TSO programs are from unprotected libraries. The intention is that these rules prevent you from hacking the protected program.

Setting up an environment that effectively uses execute permissions requires some planning and involves use of the PROGRAM class in RACF. You would start by reading the relevant sections of the RACF Security Administration Guide.

5.15 How to work with SYS1.UADS

SYS1.UADS is the original definition file for TSO users. It can still be used, but most installations transfer the TSO user definitions to RACF and administer them there. This has already been done for the AD CD-ROM system, and *you will never need to work with SYS1.UADS under normal circumstances*. This section is included for those who have special situations that require them to work with SYS1.UADS.

SYS1.UADS administration is done with the ACCOUNT command. This is a basic TSO line command, and is best not used under ISPF. Your userid must have the ACCT attribute in SYS1.UADS (or in RACF, if RACF is operational and you have a TSO segment in RACF) before you can use the ACCOUNT command. Once you execute it, you remain in the ACCOUNT program until you issue an END subcommand. While in the ACCOUNT program, you can issue ADD, CHANGE, LIST, and DELETE subcommands. ACCT authority for SYS1.UADS is similar to SPECIAL authority for RACF, but the two authority controls are completely separate—one does not require or imply the other.

If you issue the ACCOUNT command and receive an allocation error message when you attempt to issue a subcommand, issue an END subcommand, and then enter the TSO command:

```
ALLOC DA('SYS1.UADS') F(SYSUADS) SHR
```

Then try the ACCOUNT command again.

SYS1.UADS is a partitioned data set. Each defined TSO userid is represented by a member.¹² A member name is the same as the userid, with the digit "0" added to it. Each userid can be defined with an elaborate structure of multiple passwords, with each password having multiple account numbers, and each account number having multiple logon procedures. The following examples describe very simple userids: with one password, one account number, and one logon procedure. If you need something more complex, you should read the formal *TSO Administration* manual.

The following examples use account number 1, which is an arbitrary choice. The author's system does not collect (in SMF) accounting data. However, we often find it more convenient to specify a trivial account number than to indicate that no account information exists. We use 1 for the account number for locally-added users, all JOB statements, and so forth.

The following are a few examples of basic ACCOUNT subcommands that can serve as patterns for many simple administrative tasks:

LISTIDS	brief list of all userids
LIST (*)	list all entries in SYS1.UADS
LIST (KURT)	list a specific userid
ADD (OGDEN PATTI 1 ISPFPROC)	add user OGDEN, password = PATTI, account number = 1, logon procedure = ISPFPROC
ADD (OGDEN PATTI 1 ISPFPROC) JCL OPER	add user with attributes
CHANGE (OGDEN PATTI 1 ISPFPROC) DATA (MYPROC)	change logon procedure
CHANGE (OGDEN) JCL OPER ACCT	change attributes
CHANGE (OGDEN PATTI) DATA(FRANCAT)	change password to FRANCAT
DELETE (OGDEN)	delete a user

The syntax and logic of the ACCOUNT subcommands can be confusing. It is not intuitive, and you are unlikely to guess the proper subcommand syntax. The use of the DATA parameter is especially confusing, because its meaning depends on the number of operands in the first parameter set (the one containing the userid). If you need to do anything beyond the functions shown here, you should consult the full manuals first.

When you list SYS1.UADS entries, you will notice that passwords are displayed in clear text. For this reason, SYS1.UADS should not be readable (or writable) by normal TSO users or batch jobs. You should use RACF to protect it, with UACC(NONE).

A number of attributes can be associated with a userid, including MAXSIZE, SIZE, UNIT, ACCT, DEST, JCL, MOUNT, OPER, HOLD, JOBCLASS, MSGCLASS, SYSOUT, RECOVER, PERFORM, and USERDATA. You can read about these in the full documentation. In general, the ACCT attribute should be given only to the system administrator (and to emergency userids). ACCT permits the userid to use the ACCOUNT command. JCL, OPER, and HOLD should probably be given to all users in a AD development environment. The MOUNT attribute might be given, permitting the user to MOUNT a tape or disk volume from the TSO environment. The other attributes may not be needed in a small development environment.

¹² To be clearer, most userids are not defined in SYS1.UADS. Normally only a few emergency userids are defined there.

Unfortunately, default SDSF controls (as set up in the AD system) use ACCT or OPER authority to determine if a user can display output for other users' jobs. You may need to give ACCT authority to users who are not expected to use the ACCOUNT command. Also note that most of the authorities (ACCT, OPER, JCL, and so forth) can be granted through RACF, for RACF defined users.

If RACF is in use, the interaction between RACF and SYS1.UADS is not obvious. For TSO logon, it has these characteristics:

- ▶ If the logon userid exists in RACF, the RACF database is searched for a TSO segment.
- ▶ If a TSO segment does not exist, SYS1.UADS is searched for a matching userid, and its attributes are used.
- ▶ If the logon userid does not exist in RACF, SYS1.UADS is searched for the userid.
- ▶ If it exists in UADS (and not in RACF), the logon is permitted (using the password in UADS), but the user will have only RACF public (UACC) access to data sets.

5.16 How to RACF protect your data sets

Assume you have data sets PROJECT2.LIB.C, PROJECT2.LIB.LOAD, PROJECT2.TEST.DATA1. and so forth. You want to protect these data sets. How should you start? There are many ways to go about this. The following is a simplified cookbook approach, using RACF commands issued from a TSO command line:

- ▶ Determine if there are any discrete RACF profiles already protecting these data sets.

```
LISTDSD DATASET('PROJECT2.LIB.C')
LISTDSD DATASET('PROJECT2.LIB.LOAD')
LISTDSD DATASET('PROJECT2.TEST.DATA1')           and so forth to check all data sets
```

You will get the message NO RACF DESCRIPTION FOUND FOR PROJECT2.LIB.C, and the same message for the other data sets. If a profile exists you can use it (perhaps after alteration, as described below) or delete it. If your data sets all begin with the HLQ PROJECT2 (and if no other data sets begin with the same HLQ), you probably want to delete these discrete profiles and create a generic profile, as described below. A profile can be deleted with the RACF command DELDSD 'PROJECT2.LIB.C'. However, to issue this command you must own the profile, or have *alter* access to it, or have RACF SPECIAL authority.

- ▶ Determine if a generic profile exists:

```
LISTDSD DATASET('PROJECT2.LIB.C')
```

If no profile exists, you will receive the same NO RACF DESCRIPTION FOUND FOR ... message. You might find that a profile already exists and, in our example, the profile could be for PROJECT2.* or PROJECT2.LIB.* or PROJECT2.LIB.C. If it exists, it is probably for PROJECT2.* and you already have a profile protecting your data sets. You may want to alter it. (You may also find the profile name has two asterisks instead of one asterisk. This is related to the EGN function of RACF and is described elsewhere. We will assume that EGN is disabled and a single asterisk is used.) You can delete an existing profile with the DELDSD command, providing you have the necessary authority.

- ▶ Now create a generic profile to protect your data sets. If you control the HLQ PROJECT2, you can work with this name. If not, are you manager of the names PROJECT2.LIB... and PROJECT2.TEST...? We will assume this case:

```
ADDSD 'PROJECT2.LIB.*' GENERIC UACC(NONE) OWNER(mygroup)
ADDSD 'PROJECT2.TEST.*' GENERIC UACC(NONE) OWNER(mygroup)
PERMIT 'PROJECT2.LIB.*' ID(joe) ACCESS(alter)
PERMIT 'PROJECT2.TEST.*' ID(joe) ACCESS(read)
PERMIT 'PROJECT2.LIB.*' ID(mygroup) ACCESS(READ)
```

```
PERMIT 'PROJECT2.TEST.*' ID(mygroup) ACCESS(ALTER)
```

Here you create two profiles. (If you want to manage all names PROJECT2..., they you would need only one profile.) You decided to set the default access to NONE. You then enter PERMIT commands for the users or groups you will allow to access these data sets. You are the owner of the profile (because you created it), and you automatically have *alter* access to it.

A normal user cannot create protection profiles for any arbitrary name. You can create profiles whose HLQ matches your userid. In our example, we assume you are logged on as user PROJECT2. (An alternative is to be a member of group PROJECT2, with CREATE authority in the group. See “Who can issue RACF data set profile commands” on page 125 for a quick overview of who can issue selected RACF commands.)

- ▶ To alter an existing profile, use the ALTDS command with the existing profile name, and new values for whatever parameters you want to change. For example, if profile PROJECT2.* already exists with UACC(READ), you could change it with this command:

```
ALTDS 'PROJECT2.*' UACC(NONE)
```

5.17 Who can issue RACF data set profile commands

Whatever user created a RACF data set profile is the *owner* of that profile unless an OWNER parameter was used to assign ownership to another userid. The following is a quick overview of the RACF commands related to data set profiles:

- ▶ ADDSD (or AD) commands to create a data set protection profile:
 - Current userid has RACF SPECIAL authority, or
 - Current userid matches the HLQ of the profile being created.
- ▶ ADDSD (or AD) commands to create a data set protection profile for a group data set:
 - Current userid has RACF SPECIAL authority, or
 - Current userid has RACF OPERATIONS authority, or
 - Current userid has CREATE authority in group.

If the data set HLQ is a group name, it is considered a group data set. A userid can be given CREATE authority with the command:

```
CONNECT job GROUP(mygroup) AUTHORITY(CREATE)
```

- ▶ DELSD (or DD) commands to delete a data set profile:
 - Current userid has RACF SPECIAL authority, or
 - Current userid is owner of the profile, or
 - Current userid matches the HLQ of the profile.
- ▶ PERMIT (or PE) commands to grant access to a data set profile:
 - Current userid has RACF SPECIAL authority, or
 - Current userid is owner of profile, or
 - Current userid matches HLQ of profile, or
 - (For discrete profiles only) current userid has *alter* access to the profile.

ALTDS (or ALD) commands to alter a profile. These have the same rules as the ADDSD command.

5.18 S/390 disk types, models, capacities

Normal disks, on mainframes, have specific sizes. For example, a 3390-1 volume contains 1113 usable cylinders.¹³ The emulated I/O support programs (for P/390 and MP3000, but not for EFS systems) are unusual in that they can create (emulated) disks with any number of cylinders, up to the maximum size of the largest model of the (emulated) disk type. For example, an emulated 3390 volume can contain from 1 to 3339 cylinders; an emulated 3380 volume can contain 1 to 2655 cylinders; an emulated 9345 volume can contain from 1 to 2156 cylinders. OS/390 supports these odd disk sizes, if they are initialized properly.

The following table will be helpful in understanding disk volumes:

Disk Model	Usable Cylinders	Tracks/Cyl	Bytes/Track	Bytes/Cyl	Bytes/Vol
3380-AJ4	885	15	47476	712140	630 MB
3380-AE4	1770	15	47476	712140	1.26 GB
3380-AK4	2655	15	47476	712140	1.89 GB
3390-1	1113	15	56664	849960	946 MB
3390-2	2226	15	56664	849960	1.89 GB
3390-3	3339	15	56664	849960	2.83 GB
3390-9	10017	15	56664	849960	8.5 GB
9345-1	1440	15	46456	696840	1.0 GB
9345-2	2156	15	46456	696840	1.5 GB
emulated 3380	1 - 2655	15	47476	712140	max 1.89 GB
emulated 3390	1 - 3339	15	56664	849960	max 2.83 GB
emulated 9345	1 - 2156	15	46456	696840	max 1.5 GB

Note that emulated disks discussed here exist on P/390-based systems (including S/390 Integrated Servers), R/390s, and Multiprise 3000s. Netfinity/EFS systems also have emulated disks, although the emulation technology is quite different than the P/390-based machines. Also note that the *primary volumes* on an MP3000 are not emulated disks in the context used here.

The sizes listed can be misleading. A 3390 track, for example, is listed as 56664 bytes. This is true only if a single large record is written on the track. Multiple records require space (*gaps*) between the records. The gap size is complex for 3390s and is usually obtained from published tables. If 4096 byte records are written on a 3390, for example, a track will hold 12 of these. In this case, the effective track capacity is 12*4096 or 49152 bytes. The capacity of a 3390-3, filled with 4 KB records, would be 2.46 GB.¹⁴

¹³ There are actually several more cylinders. These include spare tracks and a cylinder for diagnostic use. P/390-based systems and MP3000 emulated disks do not emulate these additional cylinders. FLEX-ES EFS systems do emulate the additional cylinders. The numbers shown in the table in this section do not include these extra cylinders.

¹⁴ For disk sizes, 1K = 1000, 1 MB = 1,000,000, and so forth. Multiples of 1024 are used only for *memory* sizes. This convention is not universal, and there is often confusion about exactly what 1 KB, 1 MB, and 1 GB means.

If you have a *dump tape* from a mainframe disk, and you plan to *restore* the volume to an emulated disk, you should create an emulated volume that is the same disk type and size as the original mainframe drive. If you plan to restore individual data sets from a dump tape, you should, in general, create the same type of emulated disk (3380, 3390, 9345), but it need not have the same number of cylinders as the original mainframe drive. This is not always true, depending on the exact format of the original data sets. However, you can avoid problems by using the same device type as was used on the originating system.

We recommend that you do not create too many disk volumes. In particular, do not try to match the hundreds of minidisks that are found on VM systems. While OS/390 is capable of doing this, the system management aspects become unreasonable for a small system. A reasonable number of emulated disk volumes might be in the 10 to 20 range. The P/390-based machines and the MP3000 system have an absolute maximum of 255 emulated I/O devices, where each *device* is defined by a line in the P/390 or Emulated I/O configurator file.

5.19 How to add additional emulated disk volumes

The first step in adding additional (emulated) disks is to decide whether they should be 3380, 3390, or 9345 volumes, these being the only practical choices. You can install emulated 3375, 3350 and 3330 volumes, but OS/390 does not support these older devices. Some effort is needed to create additional emulated disk volumes, and we recommend that you plan ahead and create multiple volumes at the same time. In practice 9345 volumes are seldom used and 3380s are used only if backward compatibility is required for a particular purpose, so we will restrict our comments 3390 devices.

The following discussion excludes EFS emulated disks. They have a different set of rules.

- ▶ Decide how many disk volumes and which disk types you want, and how large each volume should be. If possible, select OS/390 addresses that are (1) already generated for the correct device type in your OS/390 system, and that (2) are not used by any of your existing volumes. The pregenerated addresses for the AD system are listed in “I/O configuration for OS/390” on page 15.
- ▶ Examine your OS/2 disks to determine if sufficient disk space is available. In OS/2 terms, a volume will be about 2% larger than the logical size required for a volume of the appropriate device type.
- ▶ Devise volume serial numbers (*volser*s) for the new volumes. These are six-character names that must be unique within your system. Typical *volser*s to be added to a AD OS/390 system might be TSOPAK, WORK01, SPOOL1, and so forth. You can assign any *volser* name you like, but we recommend using only letters and numbers. Upper-case letters are required.
- ▶ Decide on an OS/2 name for the OS/2 file used for each emulated volume. For example:

<i>volser</i>	OS/2 file name
TSOPAK	D:\MVS\TSOPAK.330
WORK01	D:\MVS\WORK01.A91
SPOOL1	E:\SPOOL1.A99

You can use any OS/2 names you like, but we strongly recommend a naming convention similar to that shown here. A name must be a full path name, starting with a drive letter followed by a colon followed by a back slash.

- ▶ If the drive will be larger than 2 GB (a 3390-3, for example), your OS/2 name must end with the two characters `_1` (that is, underscore one). A name might be `G:\BIGWRK_1.A92`. The disk emulated I/O creation function (accessed through the

configurator) will automatically make a second file with the last two characters as `_2` (underscore two).

- ▶ While OS/390 is running, use the ISPF editor to add your new volumes to PARMLIB(VATLST00). This can be done before the volumes exist. (Be certain to edit the right PARMLIB!) This will ensure that the new volumes are assigned the correct mount attributes during MVS IPL. The VATLST entries might be something like:

```

SPOOL1,1,2,3380 ,N
TSOPAK,1,2,3380 ,N
WORK01,1,0,3380 ,N

```

These entries cause SPOOL1 and TSOPAK to be mounted as PRIVATE, and WORK01 to be mounted as STORAGE. See “How to use OS/390 volume use attributes” on page 129 for more information about mount attributes.

- ▶ Using the P/390 configuration panel (or the MP3000 Emulated I/O Configuration panel), define new volumes. Enter the information for one volume in the top line of the configuration panel. When you press ENTER, the configuration function will check to determine if the OS/2 file name already exists. If it does not (and it should not, in our example), it will be allocated and formatted for use as a CKD volume. In this example, we are creating a new volume at address A99:

```

>A99 >3390 >SPOOL1 >440C >2 >E:\SPOOL1.A899
131 3390 WORK01 1113C 2 D:\MVS\WORK02.131
130 3390 TSOPAK 885C 2 D:\MVS\TSOPAK.130

```

These are large files and the formatting process can take many minutes. (For an MP3000, the first field in the configurator entry is a two-digit Unit Address.)

- ▶ You must enter the size of a new emulated disk, such as 440 cylinders in the example above. You can also enter the size as a number of megabytes. We suggest you specify a number of cylinders rather than a number of megabytes. A megabyte operand may result in a disk size that is slightly different than what you intended to produce. See the table in “S/390 disk types, models, capacities” on page 126 for standard disk sizes and number of cylinders. OS/390 is normally not sensitive to *standard* sizes, but other operating systems--such as VSE--can be sensitive to the exact number of cylinders needed to make a *standard* disk model. Do not exceed the number of cylinders listed in the second column of the table; note that 3390-9 emulated is not provided.
- ▶ An MP3000 must have the new drive(s) defined in its IOCDs.
- ▶ At this point, the P/390 subsystem must be restarted in order to include the new volumes. An MP3000 must have a Power On Reset (POR).
- ▶ After restarting the emulated I/O subsystem (restart P/390 or POR MP3000) and after OS/390 is IPLed, you should place the new volume offline, using an OS/390 command such as `V A99,OFFLINE`. The volume does not yet have a valid VTOC and will cause errors if OS/390 attempts to use it. Also, it must be offline in order for the following ICKDSF job to format it.
- ▶ Use TSO to create and submit a job something like this:

```

//OGDEN5 JOB 1,OGDEN,MSGCLASS=X
// EXEC PGM=ICKDSF
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
INIT UNITADDRESS(A99) NOVERIFY VOLID(SPOOL1) OWNERID(P390) -
VTOC(0,1,14)
/*

```

- ▶ This job will format the volume for OS/390, building a new VTOC (Volume Table of Contents). You can have multiple INIT statements in one job, if you need to initialize

multiple volumes. OS/390 will issue a console message asking permission to initialize each volume. You must reply 'U' to each message.

- ▶ After the job completes, you can vary the volume online and issue an appropriate MOUNT command if appropriate, or re-IPL MVS to cause the volumes to be mounted according to the VATLST entries.

The process of initializing a disk (with ICKDSF) is often called *DASDing* a volume, or *to DASD* a volume. These terms are not completely logical, but they are widely used.

5.20 How to use OS/390 volume use attributes

The following discussion does not apply to DFSMS-managed volumes. In principle, OS/390 *use attributes* apply to disk and tape volumes. In practice, the use of these attributes with tapes is not relevant, and we ignore it here. A disk volume is *always* mounted with one of the following *use attributes*:

- ▶ PRIVATE

New data sets will be created on this volume only if the user (via JCL or the TSO ALLOCATE command or an ISPF menu specification) specifies the volser of this disk volume.

- ▶ PUBLIC

OS/390 may place a temporary data set on this volume, if the user (via JCL or otherwise) did not specify a volser for the temporary data set. Data sets may also be placed on the volume by specifying the volser, as with PRIVATE volumes. A temporary data set is one with a DSNAME beginning with an ampersand and/or with a disposition equivalent to NEW,DELETE.

- ▶ STORAGE

OS/390 places permanent data sets on this volume if the user did not supply a volser for the data sets. In addition, temporary data sets, and data sets placed by volser name may also be placed on the volume.

The use attribute controls the usage of a volume for new data sets. It has no other function; it is not related to any security/RACF controls. PUBLIC and STORAGE volumes may be busy if many temporary data sets are in use. Compiler work files are examples of temporary data sets. PUBLIC and STORAGE volumes eventually accumulate temporary data sets that were not deleted when they should have been deleted. In addition, STORAGE volumes accumulate permanent data sets -- when a user forgets to specify a volser for a new data set, or perhaps when his normal volume is full.

Volume use attributes are set in two ways:

- ▶ Using the OS/390 MOUNT command
- ▶ Using the VATLST00 member in SYS1.PARMLIB.

The MOUNT command was heavily used when mainframe disks were removable (via *disk packs*). This is no longer the case, and the MOUNT command today is used only for special disk manipulation, usually by a systems programmer.

The VATLST function is used automatically, during IPL. It is a simple text file, in PARMLIB(VATLST00); you can edit it with the ISPF editor. The format is explained in detail in the *OS/390 Initialization and Tuning Reference* manual. Changes are not effective until the next IPL. Typical VATLST entries look like this:

```
VATDEF IPLUSE(PRIVATE)
```

```

TSOPAK,0,2,3380  ,N          (column sensitive)
WORK01,0,0,3380  ,N
PAY*  ,0,2,3390  ,N
1234567890123456789012    <-- indicates columns
      1          2

```

The VATDEF line specifies the default use attribute for disk volumes found during IPL; it is used if no other line in the VATLST applies to the volume. In this example, volumes are set to PRIVATE if there is no VATLST entry for them. The default IPLUSE value is PUBLIC. The remaining lines in VATLST are sensitive to column location. (The column numbers in the example above are not included in VATLST.) Briefly, the parameters are:

- ▶ Columns 1 - 6 (six characters) contain the volser. Wild cards may be used; an asterisk representing any number of characters and a percent sign representing one character.
- ▶ Column 8 (one character) is 0 for permanently resident volumes, and 1 for reserved volumes. These states are no longer very relevant, and you can use 0 in most cases.
- ▶ Column 10 (one character) is the use attribute. STORAGE is denoted by 0, PUBLIC by 1, and PRIVATE by 2.
- ▶ Columns 12 - 19 (eight characters, left justified) is the device type.
- ▶ Column 21 (one character) indicates whether the operator should be requested to mount the volume if it is not found during IPL. Always specify N (for NO) in this column.

5.21 How to inspect the master catalog

You can use a batch job such as the following to list everything in the master catalog.

```

//CATLIST JOB 1,OGDEN,MSGCLASS=X
// EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
LISTCAT ALL CAT(CATALOG.OS390.MASTER)
/*

```

Or use the TSO command LISTCAT ALL CAT('CATALOG.OS390.MASTER') to list everything in the master catalog. The name shown here, CATALOG.OS390.MASTER, is the name of the master catalog in the AD R10 system. There is nothing magic about this name, and other OS/390 systems may have a different name for the master catalog. Our small AD system produced over 5,000 lines of output for this command (which is another reason to use batch jobs instead of TSO commands for IDCAMS). Such a listing contains many details you probably do not want.

The TSO command LISTC will produce a smaller listing in a better summary format.

The catalog information is structured something like this:

```

NONVSAM ----- SYS1.SCLDBLD
HISTORY
  DATASET OWNER -- (NULL)    CREATION ---- 1997.030
VOLUMES
  VOLSER ----- OS39RB     DEVTYPE -----X'xxxxxxxxx'
ASSOCIATIONS
  ALIAS ----- CBC.SCLDBLD
.....
.....
.....
ALIAS ----- CBC.SCLDBLD
ASSOCIATION

```



```
NONVSAM ----- SYS1.SCLDBLD
```

We have extracts from two catalog entries here: a NONVSAM entry and an ALIAS entry. In this case, the ALIAS refers to the NONVSAM entry shown. In this example, CBC.SCLDBLD is another name for SYS1.SCLDBLD.¹⁵ Note that this is a different use of the term *ALIAS* than the ALIAS associated with user catalogs. Also note that the OWNER field shown is not related to the RACF concept of data set profile owners.

The DEVTYPE field is a hex constant that may be useful for certain assembly-level interfaces. The VOLSER field contains the most relevant information; asterisks in this field mean that the volser of the current IPL volume should be used.

A more useful way to inspect the master catalog uses this command:

```
LISTCAT UCAT ALL CAT(CATALOG.OS390.MASTER)      (use quotes in TSO)
```

This will list the user catalogs that are known to the master catalog and the ALIAS entries associated with these user catalogs. These are the ALIAS entries that we want to use to catalog your data sets (your TSO users and your major applications) in your user catalog(s). This listing is much smaller, and examples might look like this:

```
USERCATALOG --- MCAT.SYSA          (this entry not relevant)
VOLUMES
VOLSER ---- SCPPAK
ASSOCIATIONS (NULL)
USERCATALOG --- USERCAT.OS39D2     (catalog on DLIB volume)
VOLUMES
VOLSER ---- OS39D2
ASSOCIATIONS
ALIAS ---- OS39D2AD
ALIAS ---- DLIB
ALIAS ---- IPO1
USERCATALOG --- USERCAT.OS39H1     (catalog on previous HFS volume)
VOLUMES
VOLSER ---- OS39H1
ASSOCIATIONS (NULL)
USERCATALOG --- USERCAT.OS39H2     (catalog on current HFS volume)
VOLUMES
VOLSER ---- OS39H2
ASSOCIATIONS
ALIAS ---- HFS
USERCATALOG --- USERCAT.OS39R1     (catalog on previous IPL volume)
VOLUMES
VOLSER ---- OS39R1
ASSOCIATIONS
ALIAS ---- OS39R1AD
ALIAS ---- OS39R1
ALIAS ---- MVSTZN
USERCATALOG --- USERCAT.OS39R2     (catalog on current IPL volume)
VOLUMES
VOLSER ---- OS39R2
ASSOCIATIONS
ALIAS ---- OS39R2AD
ALIAS ---- SMPE
USERCATALOG --- USERCAT.P390DX     (catalog on CICS/DB2/IMS volume)
VOLUMES
```

¹⁵ In some ways, this use of a catalog alias for a data set name is similar to the UNIX symbolic link function. As long as the data set is accessed through the catalog, it has multiple names. This multiple naming is quite unusual and nontraditional in OS/390, and is a relatively recent feature of the catalogs. We strongly suggest you do not use it often--traditional MVS users find it disconcerting. We intentionally used it in this example to make this point.

```

VOLSER ---- P390DX
ASSOCIATIONS
ALIAS ---- CICSTZN
ALIAS ---- CICTZN
ALIAS ---- CICSZDN
ALIAS ---- DBSDZN
ALIAS ---- DBSTZN
ALIAS ---- CICS
ALIAS ---- DB2
ALIAS ---- DB2PM
ALIAS ---- DBS
ALIAS ---- IMS
ALIAS ---- QMF
etc

```

A few points about user catalogs should be emphasized:

- ▶ The location (volume) containing the user catalog is not related to the volume containing the data set that is cataloged. A user catalog on volume USER01 can catalog data sets that reside on WORK01, USER01, OS39M1, or almost anywhere else.
- ▶ The ALIAS for a user catalog typically equates to the HLQ of a data set name. In the above listing, OS39R1AD is an alias for user catalog USERCAT.OS39R1. If you create data set OS39R1AD.MY.LIB (and cause it to be cataloged), the system will automatically catalog it in USERCAT.OS39R1.
- ▶ If no ALIAS exists for an HLQ, a data set will be cataloged in the master catalog. In the above listings, there is no alias for JOE, so data set JOE.HIS.LIB will be cataloged in the master catalog.
- ▶ These cataloging locations (master catalog or user catalog), based on the HLQ, can be overridden by the use of JOBCAT or STEPCAT statements or direct use of IDCAMS commands. This is rarely done.
- ▶ Relevant volumes must be mounted. In the example just used, you would be unable to catalog OS39R1AD.MY.LIB. The ALIAS in the master catalog points to USERCAT.OS39R1 which is on volume OS39R1. You probably do not have this volume if you are using a later version of the AD system. The implication is that you should use release-independent user catalogs for your data sets.

5.22 How to define user catalogs

Use the JCL shown here to define a user catalog:

```

//P390A JOB 1,P390,MSGCLASS=X
//DEFINE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE USERCATALOG (           +
    NAME ( CATALOG.SYSLAB )    +
    ICFCATALOG                  +
    VOLUME ( LOCAL1 )          +
    CYLINDERS ( 15 5 ) )

```

This JCL defines a User Catalog named CATALOG.SYSLAB on volume LOCAL1 with 15 cylinders primary allocation and 5 cylinders secondary allocation. You can change these values, and anything else in this example, to fit your needs. This is a rather large catalog and should be ample for most AD system needs.

After defining a new User Catalog you should add an entry to the VLF control member to improve performance of the new catalog. Add the following line to member COFVLFxx (usually COFVLF00) in PARMLIB in the CLASS NAME(IGGCAS) section; that is, add after other EMAJ(xxxxx) statements but before the MAXVIRT(256) line

```
EMAJ(CATALOG.SYSLAB)
```

Use the following JCL to define ALIAS entries in your master catalog, pointing to your new user catalog:

```
//P390B JOB 1,P390,MSGCLASS=X
//ALIAS EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DEFINE ALIAS ( NAME(USERID1) RELATE(CATALOG.SYSLAB) )
  DEFINE ALIAS ( NAME(USERID2) RELATE(CATALOG.SYSLAB) )
  DEFINE ALIAS ( NAME(PAYROLL) RELATE(CATALOG.SYSLAB) )
  DEFINE ALIAS ( NAME(JOE) RELATE(CATALOG.SYSLAB) )
/*
```

You would need to ensure that no data sets are cataloged in your master catalog with the HLQs USERID1, USERID2, PAYROLL, or JOE before the above job is run. After it runs, any data set with a HLQ matching one of these names will be cataloged in CATALOG.SYSLAB and not in the master catalog. You would change CATALOG.SYSLAB to match the name of your new catalog, of course. Once the User Catalog has been defined you can add additional ALIAS entries at any time.

A job to define ALIAS entries, such as example above, is useful for two purposes:

- ▶ Defining initial ALIAS entries when a new user catalog is created,
- ▶ Re-defining all your ALIAS entries when a new master catalog is installed.

We suggest you maintain a job such as this, in your TSO library, with *all* the ALIASes you have added to your system. That is, every time you add a new TSO user, you should add the corresponding ALIAS line to this job. In this way, this job is always ready to update a new master catalog. We suggest you *do not* actually use a separate job, such as this example, to routinely add an ALIAS every time you define a new TSO user. (You would *update* this ALIAS job, but not run it.) Adding an ALIAS for a new user should be an integrated part of the job you use to define new users. This is discussed in “Adding new userids” on page 88.

You can add an ALIAS using TSO commands, although we do not recommend this because the potential for introducing errors is greater and you have no organized record of what you did. From a TSO READY prompt, or from ISPF option 6, you can:

```
Enter command: tso def alias
TSO responds: IDC3247I ENTER SUBFIELD OF KEYWORD 'NAME' -
You enter: 'userid1'
TSO responds: IDC3247I ENTER SUBFIELD OF KEYWORD 'RELATE' -
You enter: 'catalog.syslab'
```

and the new alias is defined and ready for use.

The migration jobs included with the AD CD-ROM assume that all your data sets are cataloged in the master catalog. The migration jobs preserve your existing master catalog entries when moving to a new OS/390 release. The migration jobs are unaware of any user catalogs you might define. See “Adding a user catalog” on page 84 for more discussion about the need for user catalogs.

5.23 How to migrate your user catalog

If you move to a new release of OS/390 by completely loading the new system, you will receive a new master catalog. If you had placed catalog entries in the old master catalog they will now be lost. If you put all of your catalog entries in one or more user catalogs, they are still available. You should maintain a job with entries for all ALIASes used in your system. You can then use this job during a move to a new release of the AD system to recreate your catalog environment.

If you are not certain about all your ALIASes for your user catalog, you can run this job while the old master catalog is still in use (that is, before you install your new OS/390 system).

```
//P390D JOB 1,P390,MSGCLASS=X
//LIST EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
LISTCAT ENT(catalog.syslab) ALL CAT(catalog.os390.master)
/*
```

You can use the output from this job to ensure that your ALIAS job is complete. CATALOG.OS390.MASTER is the name of the master catalog in the release 10 AD system; your system might have a different name, of course. In this example, catalog.syslab is the name of a locally-added user catalog.

After you have installed your new OS/390, with its new master catalog, you must re-establish the connection to your user catalog(s). This is a two step process. The first step is to reconnect your user catalog(s) to the new master catalog. You can use this job:

```
//P390E JOB 1,P390,MSGCLASS=X
//CONNECT EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
IMPORT CONNECT OBJECTS (( catalog.syslab +
DEVICETYPE(3380) VOLUMES(local1) ) )
/*
```

The second step is to re-establish the ALIAS pointers to your reconnected user catalogs. The job shown in “How to define user catalogs” on page 132, using the DEFINE ALIAS statements you have maintained as you added aliases to the system, will complete the catalog migration.

5.24 How to erase scratch disks

A *scratch disk* is one with the *use attribute* PUBLIC or STORAGE. If a job (batch or TSO) requests temporary disk space, the space will be allocated on a scratch volume. Often, a PUBLIC volume will have only scratch data sets (unless someone specifically places a permanent data set on the volume), while a STORAGE volume may have a mixture of temporary and permanent data sets. The OS/390 command D U,DASD,ONLINE can be used to determine the use attribute of online disk volumes.

In principle, OS/390 will delete temporary data sets when the job that created them ends. In practice, temporary data sets are not always deleted and tend to accumulate on PUBLIC and STORAGE volumes. Eventually they will consume enough space so that new jobs cannot obtain sufficient scratch space.

Approaches to clearing scratch disks are:

- ▶ Initialize the whole volume, using an ICKDSF job such as the one shown in “How to add additional emulated disk volumes” on page 127. This is sometimes done on very large systems, where scratch disks are only for temporary data sets, and where it is unreasonable to clean the disks by hand.
- ▶ Inspect the scratch volume(s) (using ISPF option 3.4) and delete unwanted temporary data sets one-by-one. This has the advantage that valid data sets, which happened to be placed on a scratch volume, can be recognized and preserved. We recommend this method for small OS/390 users.
- ▶ Use IEHPROGM. (See the *Utilities* manual.)

Use ISPF option 3.4, and enter the volser of the volume to be inspected. This will list all the data sets on the volume. Temporary data sets usually have a name such as SYS01180.T120135.RA000.OGDEN.R0100029. The first qualifier contains the date. If the date is older than the current date, the data set is almost certainly unwanted. The OS/390 command D T can be used to display the current date in YYDDD format. To delete a data set, enter D at the beginning of the line (in the ISPF 3.4 display) and press Enter. The data set will be deleted. You can quickly delete a few dozen data sets this way. This process will also uncatalog the data set, if it happens to be cataloged.

A few special cases may occur. The data set might be RACF protected. This is unusual for a temporary data set, but it can happen. You will need RACF OPERATIONS authority to delete it.

Unfortunately, VSAM data sets cannot be deleted as easily as other data sets. VSAM data sets are never *temporary* in the sense discussed here, but unwanted VSAM data sets (perhaps left when a test job failed) can occur. You must use IDCAMS to remove these.

5.25 How to change a volume serial number

You may need to change the volser of a disk volume without otherwise disturbing the contents of the volume. This is sometimes referred to as *clipping* the volume. There are several ways to do this but the easiest is to take the volume offline (V 135,OFFLINE) and use this job:

```
//P390Q JOB 1,P390,MSGCLASS=X
// EXEC PGM=ICKDSF
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
   REFORMAT UNITADDRESS(135) VERIFY(oldsr) VOLID(newsr)
/*
/* BE CERTAIN TO USE THE CORRECT UNITADDRESS
```

The operator will be asked to reply 'U' during the job.

Please note that there are some considerations about changing a volser that may not be obvious:

- ▶ If a VTOC index exists, it is not renamed. The name of a VTOC index usually contains the volser as the third qualifier. The index will continue to work, but the possibility of duplicate data set names (for the index) exists if another indexed volume is created with the old volser.
- ▶ Catalog pointers to data sets on the volume may become invalid.
- ▶ If a user catalog exists on the volume, you may need to EXPORT and then IMPORT it (and redefine all the ALIAS entries) to have correct volser pointers in the master catalog.

- ▶ VSAM may be confused if the volume contains any VSAM components.

In the earlier days of MVS, clipping volumes was often done for various reasons. It is much less common today for the reasons just listed.

5.26 How to remove VSAM data set components

The normal way to delete a VSAM data set is to use an IDCAMS job such as:

```
//OGDENS JOB 1,OGDEN,MSGCLASS=X
// EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE  OGDEN.DATA.VSAM CLUSTER
/*
```

You may need to delete only an element of a VSAM cluster. You can try this (assuming the catalog and VVDSs are intact):

```
//OGDENS JOB 1,OGDEN,MSGCLASS=X
// EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE  OGDEN.DATA.VSAM.INDEX2 TRUENAME
/*
```

If the catalogs or VVDSs involved are damaged or missing, this will probably not work. If it does not work, the next IDCAMS job to try is:

```
//OGDENS JOB 1,OGDEN,MSGCLASS=X
// EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//F1 DD VOL=SER=WORK01,DISP=OLD,UNIT=3380
//SYSIN DD *
DELETE  OGDEN.DATA.VSAM.DATA FILE(F1) VVR
/*
```

This is the form you could use if you see bits of VSAM data sets in VTOC listings, but normal DELETE functions do not work.

For more information see *Access Method Services for Integrated Catalog Facility*.

5.27 How to add a system-managed volume to a storage group

OS/390, before release 9, required HFS data sets to be placed on system-managed volumes. The AD system provided a predefined storage group for this purpose. (In early releases it was the DEFAULT group; in later releases it was the HFSCCLASS group.) If you wanted to add a new volume to your system and use it for HFS, you needed to (1) initialize it as a system-managed volume, and (2) add it to the appropriate storage group. (You could also create a new storage group, but adding to an existing group was much easier.) This article describes how to add a volume (already initialized as a system-managed volume) to an existing DFSMS storage group.

“How to create and use SMS disks” on page 138 describes a more general case of adding new classes and ACS routines for DFSMS. This section describes the much more restricted case of adding additional volumes to an existing storage group.

Beginning with OS/390 V2R9, you can place HFS data sets on any volume. You no longer need to work with system managed storage volumes. *If you are a new OS/390 owner, we strongly recommend that you do not use system managed volumes (unless you have particular reasons for using them) and, therefore, do not follow the instructions in this article.*¹⁶ We include it here in case (1) you are working with an older release, or (2) you have additional reasons for working with system-managed volumes.

Older AD systems contained a storage group named DEFAULT. (This name was not well chosen. It is an assigned name, and could have been MYGROUP or anything else. It is not a default name, in the general English meaning of the word *default*.) In some releases, the DEFAULT group contains volser names such as OS39H1, OS39H2, and so forth. To add volsers, follow these steps:

- ▶ Select ISMF from the ISPF menu. (It is in the *M* group.)
- ▶ If option 6 is not shown in the primary ISMF menu, do this:
 - Select option 0, ISMF Profile
 - Select option 0, User Mode
 - Select option 2, Storage Administrator
 - Use PF3 repeatedly and completely exit from ISMF.
 - Start ISMF again and you should see option 6.
- ▶ Select option 6, Storage Groups. Enter CDS Name = 'SYS1.SCDS' and Storage Group Name = DEFAULT (or whatever group name you are using).
- ▶ Select option 1, List, from the options presented in the middle of the display. This should produce a STORAGE GROUP LIST.
- ▶ In the middle of this display should be the DEFAULT group (with type POOL). In the LINE OPERATOR field, enter the command **LISTVOL** and press enter. This will list the volsers currently defined in the DEFAULT group. You can verify that the volsers you want to add are not present.
- ▶ Press PF3 twice to return to the STORAGE GROUP APPLICATION SELECTION display. Select option 4, Volume, on this screen and press enter.
- ▶ You should have the STORAGE GROUP VOLUME SELECTION display. Select option 2, Define, and enter the volsers you want to add. Use the column marked Prefix to list your new volsers. Press enter when you have added all your volsers.
- ▶ This displays an SMS VOLUME STATUS DEFINE screen. Press PF3. You should receive the message ALL VOLUMES DEFINED.
- ▶ Press PF3 twice to return to the primary ISMF screen.
- ▶ Select option 8, Control Data Set.
- ▶ Select option 5, Activate, and enter Data Set Name = 'SYS1.SCDS'.
- ▶ Press enter.
- ▶ Enter / (a slash character) in the appropriate field on the verification screen that follows.
- ▶ DFSMS will *schedule* the activation of the updated ACDS data set. Press PF3 several times to exit from ISMF. At some point you will receive a message that the new ACDS was activated.
- ▶ You need to initialize any new DFSMS volumes with the STORAGEEGROUP attribute. See the next topic for a sample job.

¹⁶ DFSMS management functions are important for large installations with hundreds or thousands of disk volumes. A typical AD user, with fewer than, say, 30 volumes simply does not require these management functions.

5.28 How to create and use SMS disks

System Managed Storage (SMS or DFSMS) is intended to automate and ease the management of large numbers of disk volumes by making the particular volume (volser) used transparent to the user. SMS routines can analyze the characteristics of a data set and make decisions based on DCB parameters, size, and several unique SMS parameters that can be coded in JCL. SMS can also provide default values for various DCB and disk space parameters. Additional DFSMS functions can address automatic migration of data sets to various levels (such as tapes) and so forth.

A certain amount of understanding and work is needed to create and maintain an SMS environment. This is acceptable in large installations where the benefits of SMS are much needed. The large-system SMS benefits are unlikely to be needed in a small AD environment, and new OS/390 owners probably do not want another complex function that they must administer. Earlier releases of OpenEdition required the use of DFSMS-managed disks for HFS data sets; the AD administrator was required to do some DFSMS work. Starting with release 9, HFS data sets (each HFS data set is a file system, in the UNIX sense) no longer need to be placed on DFSMS-managed volumes. DFSMS must still be operational, but this has already been done in the AD system. The owner of a typical AD system can generally ignore DFSMS functions.

This section may help if you are required to use DFSMS, for whatever reason. It does not attempt to be a complete introduction to DFSMS and focuses

The example in the following pages assumes you want to:

- ▶ Create a group of system managed disks. We can name the group BILLSGRP. A storage group can have many disks in it, but we will start with just one, with the volser BILL01.
- ▶ Force any new data set with the HLQ of BILLDATA to go to this storage group.
- ▶ Use the name BILLSCLS for the data class that we must define in DFSMS to implement the other functions.

The release 10 AD system already has two DFSMS storage groups defined and a volume assigned to each one. The groups are DBCLASS (volser OS3P8A) and HFSCCLASS (volser OS3H8A). The system does not require either of these classes or volumes -- they are left over from an earlier AD release. You could alter these existing groups instead of creating a new storage group.

We need to do the following tasks (not necessarily in the order shown here):

- ▶ Define a new volume to your system. This may require DEVMAP, Logical Volume, IOCDs, hardware channel connections, or whatever is needed to make a new disk volume available. It does not matter whether it is an emulated disk, an MP3000 Logical Volume, or a "real" disk.
- ▶ Vary the device offline to OS/390 using the following system command (assuming the new volume is at address A9E):

```
V A9E,OFFLINE
```

- ▶ Initialize the device with the ICKDSF program. The JCL could be:

```
//P390A JOB 1,P390,MSGCLASS=X
//STEP1 EXEC PGM=ICKDSF
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
INIT UNITADDRESS(OA9E) NOVERIFY VTOC(0,1,29) -
INDEX(2,0,10) OWNERID('P390') VOLID(BILL01) -
STORAGEGROUP
```


/*

Note the use of the STORAGEGROUP parameter. SMS volumes also require an index for the VTOC (INDEX parameter).

- ▶ Make the volume available for OS/390 with the following system commands:

```
V A9E,ONLINE
M A9E,VOL=(SL,BILL01),USE=PRIVATE
```

- ▶ Now you must make changes to the SMS configuration in your system, so it is a good idea to take a backup of the SMS control file, the ACDS (Active Control Data Set). Ideally you will have taken backups of your system, at the volume level, before you began to manipulate SMS.

The following JCL will create a backup data set for the ACDS:

```
//P390B JOB 1,P390,MSGCLASS=X
//STP001 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE CLUSTER ( +
  NAME ( sys1.acds.backup ) +
  LINEAR +
  VOLUME ( work01 ) +
  TRACKS (15 5) +
  SHAREOPTIONS (2 3) ) +
  DATA ( +
  NAME ( sys1.acds.backup.data ) )
/*
```

Now initialize the ACDS backup with the following system command:

```
SETSMS SAVEACDS(sys1.acds.backup)
```

- ▶ After backing up the active SMS environment, you must use the TSO Integrated Storage Management Facility (ISMF) functions to make changes and additions to SMS. Use ISPF to access the ISMF primary panel shown in Figure 5-3 on page 140. (To use ISMF, you must have various ISMF libraries, such as its ISPF panel library, allocated in your TSO session. The ISPFPROC logon procedure in the distributed AD system allocates all the required data sets.)

If options 6,7,8,9,10 and C are missing, you are in User mode and you need to be in Administrator mode. Change to Administrator mode by ISMF using options: 0 - ISMF Profile, then 0 - User Mode Selection and entering a 2 for Administrator Mode. You must then restart ISMF by exiting as far as the IBM Products Panel and selecting ISMF again.

```

                                ISMF PRIMARY OPTION MENU
ENTER SELECTION OR COMMAND ====> _

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

USE HELP COMMAND FOR HELP; USE END COMMAND OR X TO EXIT.

```

Figure 5-3 ISMF Primary Option panel (for administrators)

- First you should define a storage class to allow you to direct allocations of SMS data sets to your SMS disks. We will use BILLSCLS. Select option 5 from the ISMF primary panel to display the Storage Class Application panel shown in Figure 5-4.

```

                                STORAGE CLASS APPLICATION SELECTION

Command ====>

To perform Storage Class Operations, Specify:
CDS Name . . . . . 'SYS1.SCDS'
                                (1 to 44 character data set name or 'Active' )
Storage Class Name . . billsc1s (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
  5. Cache Display - Display Storage Classes/Cache Sets

If List Option is chosen,
Enter "/" to select option      Respecify View Criteria
                                Respecify Sort Criteria

If Cache Display is Chosen, Specify Cache Structure Name . .

Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 5-4 ISMF Storage Class panel

Enter the name of the SCDS (Source Control Data Set) in this panel. This is the data set where SMS definitions are stored. The definitions are then loaded into the active SMS environment. The active environment is both the ACDS (backed up in an earlier step) and the same information in OS/390 storage (in the SMS address space). The value shown in Figure 5-4 on page 140, SYS1.SCDS, is the data set used in the AD system.

Enter the name of the Storage Class you wish to add; this is BILLSCLS in our example. Enter 3 (Define) in the option field. Press Enter to display the Storage Class Define panel shown in Figure 5-5.

You may wish to experiment with the performance objectives but they hold little meaning for a small AD system. You should enter a meaningful description in the field provided. If you wish to be able to explicitly control the volume when allocating an SMS data set, change the GUARANTEED SPACE option to Y. If you do not do this, SMS will select the volume for allocation (provided the pool contains more than one volume).

Pressing the END key will complete your Storage Class definition. Continue pressing the END key until you get back to the ISMF Primary Option panel.

```

                                STORAGE CLASS DEFINE                                Page 1 of 2
Command ==>

SCDS Name . . . . . : SYS1.SCDS
Storage Class Name  : BILLSCLS
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)
Backup . . . . . (Y, N or Blank)
Versioning . . . . . (Y, N or Blank)

Use ENTER to Perform Verification; Use DOWN Command to View next Page;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.

                                STORAGE CLASS DEFINE                                Page 2 of 2
Command ==>

SCDS Name . . . . . : SYS1.SCDS
Storage Class Name  : BILLSCLS

To DEFINE Storage Class, Specify:

  Guaranteed Space . . . . . Y (Y or N)
  Guaranteed Synchronous Write . . . N (Y or N)
  CF Cache Set Name . . . . . (up to 8 chars or blank)
  CF Direct Weight . . . . . (1 to 11 or blank)
  CF Sequential Weight . . . . . (1 to 11 or blank)

```

Figure 5-5 ISMF Storage Class Definition panel (portions of two screens shown)

- ▶ Next you must allocate a Storage Group and add the SMS volume(s) to it. Select option 6 from the ISMF primary panel to display the Storage Group Application panel shown in Figure 5-6.

```

                                STORAGE GROUP APPLICATION SELECTION

Command ===>

To perform Storage Group Operations, Specify:
CDS Name . . . . . 'SYS1.SCDS'
                                (1 to 44 character data set name or 'Active' )
Storage Group Name . . billsgrp      (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 List Option is chosen,
Enter "/" to select option      Respecify View Criteria
                                Respecify Sort Criteria

Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 5-6 ISMF Storage Group Application Selection panel

The name of the SCDS should be the same as the one used for the Storage Class definition. You must enter the name of the Storage Group you wish to add (HFS in this example), POOL for the Storage Group Type and 2 in the option field (define). Then pressing Enter will display the Storage Group Define panel shown in Figure 5-7 on page 143.

You must enter 2 in the option field and enter the name of the SMS volume in the first Prefix field (BILL01 in this example). You could also use the Prefix or Suffix along with the Range parameters to define a range of volumes, for example Prefix of BILL from 001 to 010 would define 10 volumes. Press the End key to proceed to the next panel, the Volume Status Definition panel shown in Figure 5-9.

```

                                SMS VOLUME STATUS DEFINE                                Page 1 of 2
Command ==>>>

SCDS Name . . . . . : SYS1.SCDS
Storage Group Name . : BILLSGRP
Volume Serial Numbers : BILL01

To DEFINE SMS Volume Status, Specify:

System/Sys      SMS Vol   System/Sys      SMS Vol   ( Possible SMS Vol
Group Name      Status    Group Name      Status    Status for each:
-----
P390    ==>> ENABLE    S01A    ==>> ENABLE    NOTCON, ENABLE,
S01B    ==>> ENABLE    S02A    ==>> ENABLE    DISALL, DISNEW,
S02B    ==>> ENABLE    S390    ==>> ENABLE    QUIALL, QUINEW )
==>>
==>>
==>>
==>>
==>>
==>>
==>>
==>>
* SYS GROUP = sysplex
  minus systems in the
  sysplex explicitly
  defined in the SCDS

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.

```

Figure 5-9 ISMF Volume Status Definition panel

You should accept the default of ENABLE for your system. (Ignore the extraneous system names in the example and in the AD systems.) Pressing the END key will complete your Volume Status definition. That completes the Storage Group definition so continue pressing the END key until you get back to the ISMF Primary Option panel.

- ▶ The Automatic Class Selection (ACS) routine for selecting storage groups must be modified to allow you to select files for allocation in your SMS storage group. Select option 7 from the ISMF primary panel to display the ACS Application Selection panel shown in Figure 5-10 on page 145.


```

EDIT          SYS1.SMS.CNTL(BILLSCLS) - 01.00                Columns 00001 00072
Command ==>                                           Scroll ==> PAGE
***** ***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000001 PROC STORCLAS
000002 IF &HLQ = 'BILLDATA' THEN
000003   SET &STORCLAS = 'BILLSCLS'
000004 ELSE
000005   SET &STORCLAS = ''
000006 END
***** ***** Bottom of Data *****

```

Figure 5-12 ISMF ACS Edit Entry panel (Storage Class)

```

EDIT          SYS1.SMS.CNTL(BILLSGRP) - 01.02                Columns 00001 00072
Command ==>                                           Scroll ==> PAGE
***** ***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000001 PROC STORGRP
000002 IF &STORCLAS = 'BILLSCLS' THEN
000003   SET &STORGRP = 'BILLSGRP'
000004 ELSE
000005   SET &STORGRP = 'HFSCCLASS'
000006 END
***** ***** Bottom of Data *****

```

Figure 5-13 ISMF ACS Edit Entry panel (Storage Group)

These changes allow you to direct the allocation of data sets to the new SMS volume(s) by specifying the BILLSCLS storage class. The &STORGRP variable cannot be set to nulls, and must be set to a valid name. We used HFSCCLASS, because it was already defined in the AD system (and is not used for anything useful). When you have finished the changes press END until you go back to the ACS Application Selection panel shown in Figure 5-10 on page 145. From that panel select Option 2 Translate to display the panel shown in Figure 5-14 on page 147.


```

                                TRANSLATE ACS ROUTINES
Command ==>

To Perform ACS Translation, Specify:

SCDS Name . . . . . 'SYS1.SCDS'
                                (1 to 44 Character Data Set Name)

ACS Source Data Set . . 'SYS1.SMS.CNTL'
                                (1 to 44 Character Data Set Name)

ACS Source Member . . . billscls (1 to 8 characters)

Listing Data Set . . . list
                                (1 to 44 Character Data Set Name)

Use ENTER to Perform ACS Translation;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 5-14 ISMF Translate ACS Routines panel

Fields on this panel may be primed, but check them anyway. Be certain it names the correct data set and member you just edited. When you press Enter, the source ACS routine you edited in the previous step will be translated to an internal format for use with SMS. You will then be put into ISPF Browse mode to display the results of the translation. The result should be similar to Figure 5-15. Repeat the process for the BILLSGRP routine.

```

BROWSE LIST                                ACS OBJECT SAVED
Command ==>                                Scroll ==> CSR
***** Top of Data *****
ACS TRANSLATOR ***** TIME 15:08:04 DATE 06/08/2001 PAGE 0001 *****

SCDS NAME:                                SYS1.SCDS
ACS SOURCE DATA SET:                      SYS1.SMS.CNTL
ACS SOURCE MEMBER:                          BILLSCLS

0001    PROC STORCLAS
0002    IF &HLQ = 'BILLDATA' THEN
0003        SET &STORCLAS = 'BILLSCLS'
0004    ELSE
0005        SET &STORCLAS = 'HFSCCLASS'
0006    END

TRANSLATION RETURN CODE: 0000
SIZE OF OBJECT TABLE: 0000000296
***** Bottom of Data *****

```

Figure 5-15 ISMF ACS Translation Result panel for one routine

If the translation code is not zero you must determine the error from the list and go back to the edit step, correct the source and retranslate. When the translate code is zero you can proceed to validate all your changes and implement them in the running SMS system. In either case pressing the END key will display the Output Disposition panel shown in Figure 5-16 on page 148. Both routines must be translated, of course.

```

                                OUTPUT LISTING DISPOSITION   TRANSLATION SUCCESSFUL
COMMAND ===> _

LISTING DATA SET:  P390.LIST

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.

```

Figure 5-16 ISMF Output Listing Disposition panel

Change the values on the panel to print and/or keep the output listing from the translator. When you have successfully changed and translated the Storage Group ACS routine you should continue pressing the END key until you get back to the ISMF Primary Option panel.

- ▶ You must now validate the changes you have made and implement the changes into the running SMS configuration. Select option 8 from the ISMF primary panel to display the Control Data Set Application panel shown in Figure 5-17.

```

                                CDS APPLICATION SELECTION

Command ===>

To Perform Control Data Set Operations, Specify:
CDS Name . . 'SYS1.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
6. Cache Display - Display CF Cache Structure Names for all CF Cache Sets
7. Cache Update - Define/Alter/Delete CF Cache Sets

If CACHE Display is chosen, Enter CF Cache Set Name . . *
                                (1 to 8 character CF cache set name or * for all)

Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 5-17 ISMF CDS Application Selection panel

Select Option 4 to begin the validation process. The Validate ACS Routines or Entire SCDS panel shown in Figure 5-18 on page 149 is displayed.

```

                                VALIDATE ACS ROUTINES OR ENTIRE SCDS
Command ==>

To Perform Validation, Specify:

SCDS Name . . . . . 'SYS1.SCDS'
                                (1 to 44 Character Data Set Name)

ACS Routine Type . . * (DC=Data Class, MC=Management Class, SC=Storage
                                Class, SG=Storage Group, *=Entire SCDS)

Listing Data Set . .
                                (1 to 44 Character Data Set Name)

Use ENTER to Perform Validation;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 5-18 ISMF Validate ACS Routines panel

Ensure the SCDS name is correct and an * is displayed in the ACS Routine Type field and press Enter to perform the validation. You will be put into ISPF browse mode to display the results of the validate. If there are any errors the VALIDATION SUCCESSFUL message will be replaced with a VALIDATION FAILED message and error messages will appear on the bottom of the panel. You must correct the underlying reason causing validation to fail (edit the ACS code?) and validate again before proceeding. You can ignore warnings about unused storage classes. When you press END, you will see another Output Listing Disposition panel similar to Figure 5-16 on page 148.

When you have a successful validation press END until you go back to the CDS Application Selection panel shown in Figure 5-17 on page 148 and select Option 5. This will display the Confirm Activate Request panel shown in Figure 5-19.

```

                                CONFIRM ACTIVATE REQUEST
COMMAND ==> _

TO CONFIRM ACTIVATION ON THE FOLLOWING CONTROL DATA SET:

CDS: SYS1.SCDS

SPECIFY THE FOLLOWING:
Enter "/" to select option    / Perform Activation

USE ENTER TO PERFORM OPERATION; USE END COMMAND TO EXIT.

```

Figure 5-19 ISMF CDS Confirm Activate Request panel

Every time you use this panel the *Perform Activation* option must be selected; then pressing Enter will activate your changes. SMS will copy the validated data from the Source Control Data Set (SYS1.SCDS) into the SMS address space and the current disk Active Control Data Set (SYS1.ACDS). You will see the CDS Application Selection panel redisplayed with an ISPF short message displayed in the top right hand corner of 'ACTIVATION SCHEDULED'.

If the activation is successful, you will receive a TSO line mode message a short time later:

```
IGD008I New Configuration activated from SCDS SYS1.SCDS
```

5.28.1 Allocating to SMS volumes

The example above allows ACS routines to automatically place any new data set with an HLQ of BILLDATA in an SMS class. You could also use JCL (or the ISPF allocation panels) to place a new data set in the SMS class.

To allocate a data set on the SMS-managed volume in our example, you could use the Storage Class parameter, among other methods. The values to use is BILLSCLS if you followed the example in the previous section. Use the following parameter on the DD statement if using JCL in batch to allocate the file:

```
// STORCLAS=storage-class-name
```

where storage-class-name = BILLSCLS.

You can allocate data sets on SMS volumes by using the ISPF 3.2 Data Set Utility panel, as shown in Figure 5-20.

Menu	Reflist	Utilities	Help

Allocate New Data Set			
Command ==>			
Data Set Name . . . : BILLDATA.TEST			
Management class . . .			(Blank for default management class)
Storage class	BILLSCLS		(Blank for default storage class)
Volume serial	WORK01		(Blank for system default volume) **
Device type			(Generic unit or device address) **
Data class			(Blank for default data class)
Space units	TRACK		(BLKS, TRKS, CYLS, KB, MB, BYTES or RECORDS)
Average record unit			(M, K, or U)
Primary quantity . . .	1		(In above units)
Secondary quantity . .	1		(In above units)
Directory blocks . . .			(Zero for sequential data set) *
Record format	FB		
Record length	80		
Block size	0		
Data set name type :			(LIBRARY, HFS, PDS, or blank) *
			(YY/MM/DD, YYYY/MM/DD
Expiration date			YY.DDD, YYYY.DDD in Julian form
Enter "/" to select option			DDDD for retention period in days
Allocate Multiple Volumes			or blank)

Figure 5-20 ISPF 3.2 Allocate New Data Set panel

5.29 How to select data set names

OS/390 offers much latitude in naming data sets. In this context, we are discussing *normal* OS/390 data sets, not Hierarchical Files (*OE files* or *OMVS files*). Some installations have rigid rules about data set naming conventions. If your installation has such rules, then you should ignore the comments here. Typically, a new AD installation will have no established rules for data set names.

The basic OS/390 rules are:

- ▶ A data set name cannot exceed 44 characters, including the periods that separate the sections (the *qualifiers*) of the name.
- ▶ A qualifier cannot be longer than 8 characters.
- ▶ The maximum number of qualifiers is 22; the name would be something like A.B.C.D.E..., using 21 periods and 22 characters to make the name.
- ▶ A qualifier must begin with a letter (or *national character*); in particular, it cannot begin with a number.
- ▶ There are very, very few reserved names; in general, do not create a data set with the HLQ of SYS1, unless you know what you are doing. Other than this, you can use almost any name for a data set--unless your own rules, usually via RACF or SMS, stop you.

In addition to these basic syntax rules, there are some common environmental rules:

- ▶ In more common security environments (SDSF, TSO OUTPUT command, most RACF environments), a TSO user should use his TSO userid as the first qualifier of his personal data sets. For example, userid OGDEN would have data set names such as OGDEN.MYDATA, OGDEN.X.Y.JCL, and so forth. This first qualifier is the High Level Qualifier, HLQ, and is used in many security conventions. (These are only conventions. You can define a security environment that ignores HLQs. However, unless you have unusual requirements, we strongly suggest you follow the use of HLQs outlines here.)
- ▶ Major applications should have their own HLQ, and this should correspond to a RACF userid or groupid.
- ▶ Use lower-level qualifiers that convey common information about the use of the data set. Some examples are:
 - Contains JCL: CNTL, JCL, JOBS
 - Contains COBOL: COB, COBOL, SOURCE
 - Contains assembler source: ASM, SOURCE
 - Contains object modules: OBJ, OBJECT, OBJLIB
 - Contains load modules: LOAD, LINKLIB, LOADLIB
 - Contains JCL procedures: PROC, PROCLIB
 - Contains C: C, SOURCE
 - Contains text (documentation): TEXT, DOC

A few common-sense conventions, combined with the above rules, are useful:

- ▶ Never use a data set with only a single qualifier as a name.
- ▶ Do not use two-qualifier names (such as SYS1.LINKLIB) except for system data sets.
- ▶ If you wrap JCL around your COBOL source program, for example, you would probably still consider it a COBOL data set rather than a JCL data set.
- ▶ Use three- or four-qualifier names. More than four qualifiers is awkward to remember and communicate. However, there are no system or performance restrictions for using more qualifiers.
- ▶ If a data set is a PDS (or PDSE), use the letters LIB somewhere in the name. There is no fixed way to recognize a PDS/PDSE (as opposed to a sequential data set) by the name; the LIB convention is a way to solve this.
- ▶ Some reasonable examples are:

JOE.LIB.CNTL	(a JCL library)
JOE.CNTL.LIB	(another way to name a JCL library)
JOE.PROJ1.JCLLIB	(another way to name a JCL library)
JOE.PROJ1.LINKLIB	(a load library)

JOE.PROJ1.TEST.JCLLIB (more qualifiers)
JOE.TEST.PROJ1.COBLIB (COBOL source library)
JOE.TEST23.DATA (not a PDS library) (sequential or VSAM, etc)

5.30 How to recognize a VSAM catalog

Data set names such as Z9999994.VSAMDSPC... usually are VSAM catalogs. There have been three versions of catalogs during the life of MVS: CVOLs, VSAM, and ICF. Current OS/390 (and z/OS) systems use ICF catalogs, although the older forms are still usable.

The name *VSAM Catalog* can be misleading. VSAM data sets can be cataloged in ICF catalogs. There is no need to have VSAM catalogs on any modern OS/390 system. There are good reasons *not* to have VSAM catalogs, as there were a number of practical operational problems with them. You can convert a VSAM catalog to an ICF catalog with the IDCAMS command CNVTCAT.

There are no VSAM catalogs distributed with the AD system. If you receive a volume or product with VSAM catalogs, you should question the source of the material and ask them to convert it to an ICF catalog. Unless you have very unusual requirements, you do not want VSAM catalogs in your system.

5.31 RACF and SAF

Early releases of MVS did not require a security product. IBM offered the RACF security product as an optional product for MVS. There are several other competing security products, as well. For practical purposes, the current releases of OS/390 require a security product. In particular, OpenEdition requires a number of functions from the security product.

In order to provide a defined interface for a security product, MVS defined the SAF interface. Various elements of MVS call SAF for security functions, and SAF, in turn, calls the installed security product. The IBM product, RACF, has been renamed to the *OS/390 Security Server*. The correct terminology for this discussion should be *OS/390 Security Server Administration* or, sometimes perhaps, *Administration for Security Services Through SAF*. This becomes difficult to read, so we will use the collective term *RACF*, even though this is no longer the proper product or interface name for these services.

5.32 Basic security decisions

You should decide on the level of security you require for the system. This will determine how you use the security product (RACF) and how much time you devote to security tasks in the future. Should you force users to change their passwords regularly? At what interval? Should all user data sets be automatically protected from other users? Should users be able to read, but not alter, other users' data sets? Should SYS1.PARMLIB be protected against alteration by most users? Should most RACF update authority be limited to one or two people? Should normal users be permitted to update an authorized library? Should normal users be permitted to update a procedure library? Should users be permitted to browse anyone's spooled output via SDSF? Who should have UID=0 authority for Open Edition? Should a logon be required to use additional OS/390 consoles?

There are no fixed answers to these questions of security policy; answers very much depend on the nature of your installation. As a starting point, you might consider the following table:

	Closed development environment	Closed production environment	Open production environment
Use password interval?	probably not	maybe	yes
Length of password interval	N/A or 250 days	90-180 days	30-60 days
Level of standard protection?	READ	READ or NONE	NONE
Protect PARMLIB from normal users?	yes, READ	yes, READ	yes, NONE
Limited RACF SPECIAL authority?	maybe	yes	yes!
Open <i>write</i> to authorized libs?	probably	no	never!
Browse output (SDSF) of others?	yes	maybe	no
Easy access to UID=0?	yes	no	no
Require operator logon?	no	no	maybe
Many users with OPER authority?	probably	maybe	no

The three categories in the table (closed development environment, closed production environment, and open production environment) are not well defined. The typical small AD CD-ROM system is likely to be in a closed development environment. The “closed” attribute means a small group of trusted users.

5.33 How to display RACF information

There are several ways to display RACF information:

- ▶ Use RACF TSO commands. This is easy and appropriate for *ad hoc* displays of user profiles and data set profiles, for example.
- ▶ Use RACF TSO commands in batch. This is most appropriate for a set of displays that is run, unchanged, at regular intervals.
- ▶ Use the RACF ISPF panels. These may be most appropriate for display of some of the more complex RACF general resource profiles. The panels are also very useful if you do not know the syntax for a particular command.
- ▶ Use a RACF database unload function, in batch, to unload everything (except password fields) into a flat file, printable character format.

In general, *you must have authority over a RACF entry in order to display it*. A normal TSO user can display only some of the RACF data relevant to himself. A user with SPECIAL authority can display almost anything. (We say *almost* because RACF has another authority named AUDITOR who can uniquely display certain statistical data. A SPECIAL user can create AUDITOR authority, so the SPECIAL user remains the ultimate controller of RACF.)

We will list here some of the typical RACF TSO commands that might be used to display RACF information. Any of these commands can be issued at the TSO READY prompt or from ISPF option 6. Any user can issue any of the commands, but users without SPECIAL authority may not see much output in many cases.

Listing user profile information:

```

LU or LISTUSER          (display basic data for yourself)
LU (joe ogden pete)    (list basic data for several users)

```

```

    LU martin TSO OMVS          (list RACF, TSO, OMVS data for MARTIN)
Listing data set profile information:
    LD DA('payroll.input')      (list basic protection profile)
    LD DA('payroll.input') AUTHUSER (list profile and access list (ACL))
    LD ALL                       (list my data set profiles)
    LD DA('sys1.*') AUTHUSER     (list generic profile and ACL)
Listing group profile information:
    LG or LISTGRP               (list my group information)
    LG test                      (list information about group test)
    LG sys1 OMVS                 (list OMVS information for group SYS1)
Listing general resource information:
    RLIST facility *            (all profiles in FACILITY class)
    RLIST facility cnf* AUTHLIST (ACL for profile CNF* in FACILITY class)
Use the RACF SEARCH command:
    SEARCH FILTER(sys1.***)      (all profiles protecting SYS1 datasets)
    SEARCH FILTER(pay*) CLASS(USER) (all userids beginning pay....)
    SEARCH CLASS(facility) FILTER(bpx.***)

```

These are basic examples of RACF display commands. All of these commands can have more operands to more finely control the scope of the display. Any of these commands can be used in a batch job, using the JCL for executing the TSO monitor in batch; for example:

```

//BILL23 JOB 1,P390,MSGCLASS=X
//TSOBAT01 EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSUADS DD DSN=SYS1.UADS,DISP=SHR
//SYSLBC DD DSN=SYS1.BROADCAST,DISP=SHR
//SYSTSIN DD *
LD DA('JOE.*') ALL
LU JOE
/*

```

An alternate way to display RACF data is to use the RACF ISPF panels. (This assumes that the appropriate data sets are allocated to your TSO session. The full-function ISPFPROC logon procedure provided with the AD CD-ROM does this.) In general, it is safe to experiment with the RACF ISPF panels, provided you are careful to always select DISPLAY or LIST options. The ISPF RACF panels are usually started from an option in the ISPF main menu. (The current AD CD-ROM system places the RACF option on the second page of the main menu.)

Remember that RACF will display only data you are allowed to see. In general, you need SPECIAL authority to do significant RACF displays. If you do not have SPECIAL authority, you will not receive an error message -- you simply will not see much output.

We will briefly step through one path in these ISPF panels. To start, select the RACF option in the ISPF main menu. This should produce the display shown in Figure 5-21 on page 155.


```
                                RACF - SERVICES OPTION MENU
OPTION ==> 1

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
    6 REMOTE SHARING FACILITY
    7 DIGITAL CERTIFICATES AND KEY RINGS
    99 EXIT
```

Figure 5-21 RACF Services Option menu

If we select option 1 (data set profiles), we should see the display shown in Figure 5-22. If we then select choice 8 or *D Display* in this panel we will see the panel shown in Figure 5-23 on page 156.

```
                                RACF - DATA SET PROFILE SERVICES
OPTION ==> 8 (or D)

SELECT ONE OF THE FOLLOWING:

    1 ADD           Add a profile
    2 CHANGE        Change a profile
    3 DELETE        Delete a profile
    4 ACCESS        Maintain the access lists
    5 AUDIT         Monitor access attempts (for auditors only)

    D or 8 DISPLAY  Display profile contents
    S or 9 SEARCH   Search the RACF database for profiles
```

Figure 5-22 RACF Data Set Profile Services panel

```

                                RACF - DATA SET PROFILE SERVICES - DISPLAY
COMMAND ===>

ENTER THE FOLLOWING INFORMATION:

    PROFILE NAME      ===> 'SYS1.*'

    TYPE              ===>      MODEL, TAPE, GENERIC,
                                or blank

    VOLUME SERIAL     ===>      If a discrete profile and the
                                data set is not cataloged

    UNIT              ===>      If you are adding a profile
                                and specified VOLUME SERIAL

    PASSWORD          ===>      Data set password, if the data
                                is password protected

```

Figure 5-23 RACF Data Set Profile Name example

In this panel you would enter a data set profile name, such as 'SYS1.*', or 'SYS1.**', or 'P390.open1.*', and you would probably leave the rest of options blank. *You need to enter an exact name of a profile, not just the high-level portion of a name.* The difference between * and ** operands in data set profiles is related to a RACF option named EGN (Enhanced Generic Naming). EGN is not enabled in the AD CD-ROM system, and the ** operand for data set names will not work.¹⁷ If you supply an operand and press ENTER, you should see the panel shown in Figure 5-24.

```

                                RACF - DISPLAY DATA SET PROFILE
COMMAND ===>

    PROFILE: 'SYS1*'

TO SELECT INFORMATION TO BE DISPLAYED, ENTER YES:

    ACCESS LIST ===> YES      Profile access list
    HISTORY      ===>          Profile history
    STATISTICS   ===>          Profile use statistics
    DFP          ===>          Profile DFP information
    DATA SETS   ===>          Protected data sets
    NO RACF      ===>          Limit the display to the selected
                                information.

TO LIMIT THE DISPLAY TO PROFILES FOR DATA SETS ON SPECIFIC VOLUMES,
ENTER ONE OR MORE VOLUME SERIAL NUMBERS:

    ===>          ===>          ===>          ===>          ===>
    ===>          ===>          ===>          ===>          ===>
    ===>          ===>          ===>          ===>          ===>

```

Figure 5-24 RACF Display Data Set Profile Panel

¹⁷ The difference between * and ** is subtle. With EGN enabled (which is typical in most MVS installations), * stands for a single qualifier and ** stands for zero, one, or more qualifiers. The EGN option applies only to data set profiles. General resource profiles always have the ** format enabled.

You would normally enter YES to display an access list. If you press Enter once more, you should see the displayed data. It is in an ISPF temporary file and you can scroll through it in ISPF browse mode. When you have finished browsing the output, press PF3 to return to the RACF search parameter screen.

RACF output tends to look like that shown in Figure 5-25. It contains all the required information, but is perhaps not as compactly formatted as it might be for viewing on a terminal.

```

BROWSE - RACF COMMAND OUTPUT----- LINE
COMMAND ==>
***** Top of Data *****
INFORMATION FOR DATASET SYS1.* (G)

LEVEL  OWNER    UNIVERSAL ACCESS  WARNING  ERASE
-----  -----  -
00     P390          READ      NO       NO

AUDITING
-----
FAILURES(READ)

NOTIFY
-----
NO USER TO BE NOTIFIED

YOUR ACCESS  CREATION GROUP  DATASET TYPE

```

Figure 5-25 RACF Command Output display

5.34 How to back up and unload RACF

The following job will allow you to create a backup RACF database. In addition to making a copy of the database, the IRRUT200 program will verify the internal structure of the data base. The name of a backup database is up to you. You can name it RACF.BACKUP or XXX.YYY.ZZZZ, although you should follow your naming convention for high-level qualifiers (HLQs). RACF, in the current AD systems, actively maintains a backup copy of the data base. Do not use its name for the (static) backup produced by this job.

```

//P390K JOB 1,P390,MSGCLASS=X
//STEP EXEC PGM=IRRUT200
//SYSRACF DD DSN=SYS1.RACFDS,DISP=SHR
//SYSUT1 DD DSN=BILL.RACF.BKUP,DISP=(NEW,CATLG,DELETE),
// UNIT=3390,SPACE=(CYL,(10)),DCB=(LRECL=4096,RECFM=F),VOL=SER=WORK03
//SYSUT2 DD SYSOUT=*
//SYSRINT DD SYSOUT=*
//SYSIN DD *
INDEX
MAP
END
/*

```

The following IRRDBU00 job will allow you to generate a flat file containing all the information (except the encrypted passwords) in your RACF data base. This example uses the backup RACF database for input, but you could also use the primary RACF database for input by changing the INDD1 DD statement. This *unload job* can impact performance on a large system if the active RACF database is used for input; for this reason, the RACF backup data base is typically used.

```
//P390E JOB 1,P390,MSGCLASS=X
//STEP EXEC PGM=IRRDBU00,PARM=NOLOCKINPUT
//INDD1 DD DSN=SYS1.RACFDS.BACKUP,DISP=SHR
//OUTDD DD DSN=BILL.RACF.FLATFILE,DISP=(NEW,CATLG,DELETE),
// UNIT=3380,SPACE=(CYL,(10)),VOL=SER=WORK03,
// DCB=(LRECL=4096,RECFM=VB)
//SYSPRINT DD SYSOUT=A
```

5.35 How to use a local RACF database

A complete OS/390 system, such as the AD CD-ROM systems, will contain a fully operational RACF, including a RACF database. In the normal course of usage, you will probably add users and other *profiles* to this RACF data base. When you receive a new release of the OS/390 AD CD-ROM system (or any other OS/390 packaging that is a complete, IPLable system with RACF) this new release will have its own RACF database. This new database will not reflect the changes you have made to your old RACF database.

Do not attempt to change the effective names of the RACF data base (and backup). The names are SYS1.RACFDS and SYS1.RACFDS.BACKUP in recent AD systems. The names are assembled in a RACF table and you should not try to change them unless you understand what you are doing.

There are a number of approaches to handling RACF database copying and moving, but these may assume more RACF expertise than can be found in a new AD installation. Described here is a simplistic approach to this problem. It does not solve all the difficulties of creating a local, permanent RACF data base, but it is a starting point.

While you are using your *old* OS/390 system, copy the RACF database (and backup) to local volumes—that is, to disks volumes that will not be replaced by a new AD release. In this example, we use volumes named RACCAT and WORK03. You can copy the database using IRRUT200.

```
//P390C JOB 1,OGDEN,MSGCLASS=X
// EXEC PGM=IRRUT200
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSRACF DD DSN=SYS1.RACFDS,DISP=SHR
//SYSUT1 DD DSN=SYS1.RACFNEW,DISP=(NEW,CATLG,DELETE),UNIT=3390,
// VOL=SER=RACCAT,SPACE=(CYL,(7,1)),VOL=SER=RACCAT
//SYSUT2 DD SYSOUT=*
//SYSIN DD *
MAP
/*
// EXEC PGM=IRRUT200
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSRACF DD DSN=SYS1.RACFDS.BACKUP,DISP=SHR
//SYSUT1 DD DSN=SYS1.RACFNEW.BACKUP,DISP=(NEW,CATLG,DELETE),UNIT=3390,
// VOL=SER=RACCAT,SPACE=(CYL,(7,1)),VOL=SER=WORK03
//SYSUT2 DD SYSOUT=*
//SYSIN DD *
```

MAP
/*

It is possible to copy the RACF data base with IEBGENER, but using it will not prevent logical errors caused by RACF updates while the copy is being performed. IRRUT200 provides the proper locks to ensure that the copy is created correctly.

We selected the name SYS1.RACFNEW, but you could use any name. The RACF database on the distributed AD CD-ROM systems is 285 tracks (19 cylinders), but you should verify this on your system; your copy should be the same size. Your copy of the RACF database will capture the contents of the database at the time you copy it. You should create the copy (or do the copy again) just before taking the next step.

The next step is to rename and recatalog your new copies of the RACF databases, so the system will use them after the next IPL. The steps involved are:

- ▶ Work in a single-user environment. That is, stop all batch jobs, have other TSO users log off, stop CICS and TCP/IP, and so forth.
- ▶ Uncatalog SYS1.RACFDS and SYS1.RACFDS.BACKUP. (These are the “original” RACF data bases that were supplied with your system.) You can use the ISPF 3.4 option to rename, uncatalog and catalog.
- ▶ Rename SYS1.RACFDS to SYS1.RACFDS.OLD.
- ▶ Rename SYS1.RACFDS.BACKUP to SYS1.RACFDS.BACKUP.OLD
- ▶ Rename SYS1.RACFNEW to SYS1.RACFDS.
- ▶ Rename SYS1.RACFNEW.BACKUP to SYS1.RACFDS.BACKUP.
- ▶ Catalog the new SYS1.RACFDS and SYS1.RACFDS.BACKUP data sets.
- ▶ Shut down the system and re-IPL.

You will now be using your RACF database, on your RACCAT and WORK03 volumes.

After you install a new OS/390 CD-ROM release, you will initially be using the SYS1.RACFDS¹⁸ that was provided with the new release. Your new release will have a master catalog entry that points to SYS1.RACFDS and SYS1.RACFDS.BACKUP on a volume that was part of the new release. This is volume OS39M1, in the case of the current AD CD-ROM systems. *Your* SYS1.RACDS and SYS1.RACFDS.BACKUP, on other volumes, are not cataloged in the master catalog of your new release.

At some point, after whatever initial work you need with your new OS/390, you would:

- ▶ Work in a single-user environment.
- ▶ Uncatalog SYS1.RACFDS and SYS1.RACFDS.BACKUP (on system volumes supplied with the new release).
- ▶ Catalog your SYS1.RACFDS and SYS1.RACFDS.BACKUP (on your volumes).
- ▶ Re-IPL.

You are left with duplicate data set names, but only your RACF data sets are cataloged and the system follows the catalog (at IPL time) to find the RACF data base.

¹⁸ We stress that SYS1.RACFDS and SYS1.RACFDS.BACKUP are the names used with current AD system releases. Other systems may use different names, but the concepts described here remain the same.

The disadvantage of this method is that any new profiles in the newly distributed RACF database (on your new OS/390 release) are lost. There is no simple way to merge these into your old RACF. You might randomly inspect the new distributed RACF database to look for obviously new profiles (see Figure 5.33 on page 153). You might print an unloaded, flat-file listing of the new distributed RACF database and look for new entries. Or, you might ignore the problem.

Another problem is that new *class table* entries might be defined in the new release of the RACF data base. The AD systems usually provide a migration job to update the class tables. This job can be run against your new RACF data base. The same migration job *may* include RACF updates for new profiles required in the new release.

Please note that there are other ways to switch RACF databases. An even simpler approach is to use the new RACF database, supplied with the new release of OS/390, and reenter all the userids and other profiles that you added to your previous database(s). This can work if you are very orderly about RACF updates, but not all of us are orderly enough to use this method.

5.36 How to protect the master catalog

The following RACF commands can protect the master catalog, assuming its name is CATALOG.OS390.MASTER:

```
AG CATALOG
AD 'CATALOG.OS390.MASTER' UACC(READ)
PERMIT 'CATALOG.OS390.MASTER' ID(P390A) ACCESS(UPDATE)
PERMIT 'CATALOG.OS390.MASTER' ID(OGDEN) ACCESS(UPDATE)
(... PERMITS for any other userids that require catalog entries in master catalog...)
```

Because P390 and IBMUSER have RACF SPECIAL and OPERATIONS privileges, it is not necessary to enter PERMIT commands for them, but it would not hurt. Also, if userid P390 is used to enter these RACF commands, then P390 will own the data set profile and have ALTER access to it by default.

Do not issue these commands without reading “Protecting the catalogs” on page 86.

The initial AG CATALOG command needs discussion. In normal use, RACF requires that the HLQ of a name in a data set profile must be the name of a *userid* or *groupid* in RACF. There is no purpose for this group other than to provide a name for protecting the catalog. We could have named our catalog SYSLAB.CATALOG, for example, but we would then need to define a userid or groupid named SYSLAB in order to make a RACF profile for the catalog.¹⁹ The name CATALOG has no special significance as part of a catalog name. We could have named our catalog XX.YY.ZZ.AAA.BB.

5.37 How to submit jobs under another user's ID

If user AAA needs to submit a job that will run under user BBB's identity (because BBB has the necessary RACF access to whatever data sets are needed), there are several possibilities:

- ▶ AAA can include BBB's userid and password on the submitted JOB statement. Of course, AAA must know the required password. In practice, the password would probably be

¹⁹ The user catalogs supplied with the AD system all have the HLQ USERCAT. There is a groupid named USERCAT in the AD system, solely for the purpose described here. We could have named our user catalog USERCAT.SYSLAB, and reused the existing USERCAT group name.

stored (in clear text) on several JOB statements in various libraries used by AAA. This will work, but does not represent good security practice.

- ▶ Have someone with RACF SPECIAL authority define:

```
RDEFINE SURROGAT BBB.SUBMIT UACC(NONE)           (note the spelling!)
PERMIT BBB.SUBMIT CLASS(SURROGAT) ID(AAA) ACC(ALTER)
```

With this permission, AAA can submit a job with BBB's userid in the JOB statement, but without specifying (or knowing) BBB's password. In this case AAA is a surrogate for BBB. If multiple people need to submit jobs for BBB, then multiple PERMIT commands would be needed.

- ▶ Have someone with RACF SPECIAL authority define:

```
RDEFINE SURROGAT *.SUBMIT UACC(NONE)
PERMIT *.SUBMIT CLASS(SURROGAT) ID(AAA) ACC(ALTER)
```

With this permission, AAA can submit a job using *any* user's identity, by coding the desired userid in a JOB statement. This is a powerful authority and would probably not be used in a larger installation.

Note that when a job is submitted, JES checks the JOB statement for USER and PASSWORD operands. If none are found (which is the normal case), JES inserts the userid and password of whoever submitted the job. (You will not see these operands in the JCL listing of the job, but they are effectively there.)

Before it can be used, the SURROGAT class must be active. If you have RACF SPECIAL authority, you can use the TSO command SETROPTS LIST to determine if it is active. This produces a long list, but the active classes are listed near the beginning of the list. If SURROGAT is not among the active classes, issue the command SETROPTS CLASSACT(SURROGAT).

5.38 How to grant RACF access to manipulate DASD

In reading various MVS manuals, you will find references to RACF controls for DASDVOL and STGADMIN controls in RACF. These two classes provide controls for unusual access to DASD volumes:

- ▶ *DASDVOL* applies to non-SMS managed volumes.
- ▶ *STGADMIN* applies to SMS-managed volumes.

By unusual access we mean any of the following: (1) dumping or restoring a volume (which requires access to all the data sets on the volume), or (2) manipulating the VTOC or VTOC index, or (3) working outside defined data set extents or similar activities.

These two RACF classes, *if activated*, can provide fine controls over who is allowed to perform these activities, and for which volumes. In the AD systems, these classes are not activated.

If these classes are *not* activated, controls are as follows:

- ▶ A user with RACF OPERATIONS authority can dump or restore any volume.²⁰ (You may need to specify the ADMINISTRATOR keyword in the ADRDSSU job).
- ▶ Any job (usually with Superzap) that attempts to manipulate a VTOC will request permission from the MVS operator.

²⁰ There is an exception. If the userid with OPERATIONS authority has access level NONE to any profile, he cannot dump data sets protected by the profile. The normal RACF search (PERMITS, group membership, and so forth) is used to determine if NONE access exists to a relevant profile.

We suggest that these controls are sufficient for most AD installations and suggest you do not use DASDVOL and STGADMIN controls unless you have a specific need for them.

If you do need STGADMIN controls, a profile that grants all privileges might be defined this way:

```
RDEFINE FACILITY STGADMIN.** UACC(NONE)

PERMIT STGADMIN.** CLASS(FACILITY) ID(goodguy) ACC(ALTER)
SETR RACLIST(FACILITY) REFRESH
```

5.39 How to allocate data sets with HLQ of revoked user

The situation described here occurs only with DFSMS-managed volumes. If the userid that is to be used as the HLQ of a new data set is in a *Revoked* state (by RACF) and if that userid is the RESOWNER of the Storage Class or Management Class (both these are DFSMS controls) then the allocation will fail. The simple solution is to use RACF to RESUME the userid.

This is a complex situation and is generally described in the DFSMS documentation. An older redbook (unfortunately out of print) describes the situation clearly. The book may be available in larger installations; the title is *DFSMS Implementation Primer Series: DFSMS and RACF Usage Considerations*, GG24-3378.

5.40 RACF and daemons

In traditional UNIX systems, a daemon runs with UID=0 and often temporarily changes to various effective userids to perform its function on behalf of end users. In the MVS environment, a program would need to be an *authorized program* to do the equivalent functions. An authorized program must reside in an authorized library, and RACF is used to closely control which libraries are authorized.

The ability of rogue programs to acquire UID=0 status has been a longstanding security problem for UNIX systems. The OS/390 designers of the OpenEdition UNIX functions added a more rigorous requirement for programs wishing to use the effective UID of other users. This is provided by the BPX.DAEMON profile in the FACILITY class of RACF. *If this profile exists*, then a program attempting to switch to an effective UID of another user must:

- ▶ Run as UID=0.
- ▶ Have been loaded from a *program controlled* library (MVS) or file (HFS).
- ▶ Be permitted to the BPX.DAEMON profile.

The BPX.DAEMON profile can be created²¹ with:

```
RDEFINE FACILITY BPX.DAEMON UACC(NONE)

PERMIT BPX.DAEMON CLASS(FACILITY) ID(daemonuserid) ACC(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

²¹ The OS/390 R10 and z/OS R1.1 AD systems already have this class and profile active.

The RDEFINE needs be done only once. The userid defined in the PERMIT command is the userid of the daemon. This userid is probably defined with UID=0. The access level of the PERMIT can be READ or a higher level access; it acts as a simple switch. You need to issue similar PERMIT commands for all userids used by daemons. (This has already been done for the AD system.)

The FACILITY class definitions of RACF are normally RACLISTed. This means they are kept in main storage to improve performance. Whenever you change a FACILITY class definition (with a new profile or with a new PERMIT command) you need to REFRESH the in-storage copy using the command shown above.²²

If the BPX.DAEMON profile does not exist in RACF, then any program with UID=0 can change its effective UID to that of any user. That is, it works like traditional UNIX systems. The BPX.DAEMON profile is already defined in the AD system. We strongly suggest you do not write programs or procedures that depend on the absence of this profile.

See “How to program control libraries and files” on page 168 for instructions to make a library or file program controlled.

5.41 How to merge RACF databases

It would be very convenient to be able to merge the RACF databases from an older release of OS/390 into the RACF database of a newer release. IBM does not provide a program to merge RACF databases, unfortunately. The IRRUT400 utility (provided with RACF) may appear to merge databases, but it simply uses the first profile (with a given name) that it finds. It does not merge the characteristics of two profiles with the same name. (Doing this can be quite complex if the two profiles contain conflicting specifications.) IRRUT400 will merge profiles (from two databases) that do not have duplicate profile names, but this is unlikely to be what you want.

You have two options:

- ▶ Buy a third-party product that performs RACF database merges. (There is little magic in these; you will need to manually adjust the results of profiles with severely conflicting specifications.)
- ▶ Download DBSYNC from www.s390.ibm.com/racf and use it to list the *differences* between two RACF databases. You can then manually update one of the databases to match the important entries in the other one. This is likely to be sufficient for AD-class installations. Note that DBSYNC is not an IBM-supported program. It is an informal tool provided by Walt Farrell, a well-known member of IBM's Security Server team.

5.42 How to allow a user to su to root

Controlling access to UID zero has been a problem for UNIX systems.

The OpenEdition designers decided to add an additional control for the `su` command, and the control is external to the OS/390 UNIX functions.

In order to `su` to any userid with UID=0, an OpenEdition user must:

- ▶ Know the target userid.
- ▶ Be permitted to the RACF profile BPX.SUPERUSER in the FACILITY class.

²² If the class is not RACLISTed, the REFRESH will do no harm. We suggest you always issue the REFRESH command; it is a good habit, and is usually needed.

The BPX.SUPERUSER profile is almost certain to already exist in any current OS/390 system. (In the R10 AD system, the profile exists, but only IBMUSER is *permitted* to it.) To PERMIT a user to it, someone with RACF SPECIAL authority would issue the command:

```
PERMIT 'BPX.SUPERUSER' CLASS(FACILITY) ID(joe) ACC(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

Once permitted to BPX.SUPERUSER, another user can `su` to root without knowing the password of a root user.

If the BPX.SUPERUSER profile does not exist, the traditional UNIX rules for using `su` apply; that is, the caller must know the password of the target userid.

5.43 How to see detailed JES2 security messages

An undocumented operator command (which is probably unsupported) can be used to display detailed JES2 security error messages. You should not need this function during normal operation, but it might be useful in special cases. The JES2 command is:

```
$T DEBUG,SECURITY=YES
```

5.44 How to delete a user with a digital certificate

Digital certificates are a relatively new addition to RACF. If you attempt to delete a user who has a digital certificate and obtain an error message about the certificate (message ICH04014I), it is possible your RACF templates are down level. If this happens, you need to run the RACF IRRMIN00 utility with PARM=UPDATE. Once this is done, you should be able to delete users having certificates without performing any special actions.

See the standard RACF documentation for a description of the IRRMIN00 program and its usage.

5.45 How to set rules for RACF passwords

You can set a variety of rules for forming valid RACF passwords. This is done with the SETROPTS command; you must have RACF SPECIAL authority to use this command. For example:

```
SETROPTS PASSWORD(HISTORY(15) REVOKE(4) WARNING(4) RULE1(LENGTH(5:8)))
```

In this example:

- ▶ RACF will maintain a history of the last 15 passwords used by every user, and prevent any user from reusing the same password.
- ▶ A userid will be REVOKED after four failed attempts to supply a password. (If a RACF SPECIAL user exceeds the maximum password attempts, an operator message is issued and the operator can permit the user to try again. This is to prevent SPECIAL users from being locked out of the system.)
- ▶ RACF issues a message (at logon) four days before a password will expire.
- ▶ RACF requires a password to have 5 - 8 characters.

Another useful example is:

```
SETROPTS PASSWORD(INTERVAL(60))
```

which sets the maximum number of days a password can be used. That is, the user must change his password within 60 days in this example.

You can specify the internal structure of a password; for example, you may require or limit the number of numeric, consonants, and vowels used. The RACF Command Language Reference or RACF Administrator's Guide contains more details.

If much more sophisticated RACF password rules are needed, then someone needs to write a RACF *exit* program.

5.46 How to use RACF Global Access Control

You can use an optional RACF function known as the Global Access Control (GAC). This defines an in-storage table that is consulted for access permissions before the RACF data base is searched. Normally, this in-storage table is relatively small and contains mostly generic profiles. Some key points are:

- ▶ The GAC only grants permissions. It does not deny permission. If the GAC does not grant a requested permission, then the RACF data base is consulted (by normal RACF processing) for more profiles.
- ▶ The GAC normally contains DATASET profiles, but it can contain some other classes of profiles, as well.
- ▶ No audit records are generated if access is allowed by a GAC profile.
- ▶ GAC profiles are tested in the order they are defined. This can be important. The first profile that grants access ends a search.
- ▶ The R10 AD system has no GAC defined.
- ▶ The GLOBAL class is a grouping class.
- ▶ Global profiles require some thought. Consult the RACF manuals for more information before creating a GAC from scratch.

You can list your existing GAC entries with this RACF TSO command (if you have SPECIAL authority):

```
RLIST GLOBAL *
```

You could create a GAC as follows:

```
RDEFINE GLOBAL DATASET ADMEM('&RACUID.*'/ALTER)          note 1
RALTER GLOBAL DATASET ADMEM('SYS1.RACF'/NONE)           note 2
RALTER GLOBAL DATASET ADMEM('SYS1.*'/READ, 'CEE.*'/NONE)
SETROPTS GLOBAL(DATASET)                                note 3

if you later perform more RALTER commands, follow with
SETROPTS GLOBAL(DATASET) REFRESH
```

Note 1: A user always has ALTER access to data sets that have his userid as the HLQ. Placing this entry in the GAC simply helps performance. This is the only place the &RACUID symbolic parameter is used. (&USERID is used in certain other RACF profiles.)

Note 2: This entry is unusual. It specifies NONE access, but the GAC cannot deny access to anything; it only grants access. Placing this entry before the next one, which grants READ access to SYS1.*, limits the effect of the global READ access for SYS1.*. In effect, the two entries say “grant global READ access to SYS1.* except for SYS1.RACF. For access to

SYS1.RACF, consult the access profiles in the RACF data base.” There is nothing special about SYS1.RACF in this case; you can limit the effect of any following profile this way. SYS1.RACF is a good example because you are very unlikely to want global READ access to it.

Note 3: You need to issue this command once, to enable GAC processing. Do not issue it more than once. Any time you change the GAC, you must issue a REFRESH command.

General Note: Using the GAC to grant READ access for all SYS1 data sets is not advisable in a production operation. In addition to SYS1.RACF, there are many other SYS1 data sets that should not have global READ access in any installation concerned about security. In a small development environment, READ access to SYS1 is more appropriate.

5.47 How to find the name of the current RACF database

From an TSO session, issue the command

```
RVARY LIST
```

to find the name of the current RACF database(s). (If a backup RACF database is active, there will be two names.) Note that this is a TSO command, not an MVS operator command.

5.48 How to define a default OpenEdition userid

An MVS user (TSO or batch) may have an indirect requirement to access OpenEdition facilities. For example, a C program that uses TCP/IP sockets requires OpenEdition access, even when the program is used under TSO. In order to use OpenEdition facilities, the user normally needs to have an OMVS segment in his RACF definition. In particular, a UID is required before any OpenEdition function can be used, even indirectly.

You can create a *default* OpenEdition userid. The system will use this default userid information to supply a UID (and a GID) for valid system users that do not have an OMVS segment in their RACF definition.

The following RACF commands (issued by a RACF SPECIAL user) will establish a default userid and groupid for OpenEdition:

```
ADDGROUP odefg OMVS(GID(1000))

ADDUSER odefu DFLTGRP(odefg) OMVS(UID(1000) HOME('/') PROGRAM('/bin/sh'))

RDEFINE FACILITY BPX.DEFAULT.USER APPLDATA('odefuf/odefg')
```

This will permit any TSO user without a defined OMVS segment to use OMVS. If you want to permit such users to use secondary OE functions (such as TCP/IP sockets) but to not permit such users to access the OMVS shell, then use PROGRAM('/bin/echo') in the ADDUSER command.

(The specific userid and groupid used in these commands is not important. It is the APPLDATA in the RDEFINE that makes them become the default OpenEdition user/group. Likewise, you can use any otherwise unused UID/GID.)

5.49 What is a RACF user

The term *RACF user* generally means a TSO user who is defined via RACF. The alternative is to define the user through SYS1.UADS, and have no RACF definition for the user. Such a user is sometimes known as a *UADS user*. In earlier MVS times, there were situations where both types of users existed, sometimes for extended periods.

A UADS user, completely unknown to RACF, is rare today. You should not define such users in your system unless you have quite unusual requirements. Note that you may have a few users defined both in RACF and in UADS. In the AD system, IBMUSER is defined this way. These are often defined as emergency userids, intended for use only if RACF is not usable for some reason.

5.50 Understanding the difference between UACC and ID(*)

You may sometimes find documentation that discusses the advantage of ID(*) protection versus UACC protection. The UACC protection you define for a data set profile or a general resource profile specifies the level of protection for any user that does not have a specific defined level of access to the resource.

The key point is that the UACC applies to *any* user, including those defined through UADS instead of being defined to RACF. That is, a user completely unknown to RACF will have the UACC level of access to resources.

An alternative is to define UACC(NONE), allowing no access, and then define PERMIT ID(*) ACCESS(READ) (or whatever general level of access is desired). The ID(*) matches any defined RACF user, but will not match a user unknown to RACF. This might be important in narrowly defined cases.

5.51 How to locate HFS files with invalid UID

All HFS files have a numeric UID and a numeric GID stored to indicate the ownership of the file. Internal RACF functions are used to translate UIDs and GIDs to equivalent userids and groupids. For example, UID 101 might be userid P390A. If a UID or GID number stored with a file does not have a RACF identity, two things happen:

- ▶ Directory lists, with the `ls -al` command for example, list the numeric UID and/or GID instead of a name, and
- ▶ Performance suffers due to excessive RACF searching. (Newer OS/390 releases have mechanisms to partly reduce this extra overhead, but the same considerations apply.)

For these reasons, your files systems should not contain UIDs or GIDs that are unknown to RACF. You can find such files with this OpenEdition command:

```
find / -nouser -o -nogroup
```

To prevent many *access denied* messages, you should issue this command from a userid that has UID=0. If it finds files without known UIDs or GIDs, you should consider using the `chown` command to change ownership. If the files or directories are seldom accessed, the need to make changes is less important.

The most common source of unknown UIDs and GIDs is imported tar files, but there are many other sources.

5.52 How to find data set names protected by RACF profile

You can use a command such as:

```
LD DA('ogden.*') ALL DSNS
```

to list the names of all cataloged data sets that are protected by the indicated profile. This can be very useful, but remember these points:

- ▶ Only cataloged data sets are listed. Furthermore, only currently active catalogs are searched.
- ▶ Use a single asterisk for generic names if EGN is not active. You will receive an error message if you use a double asterisk. (The AD systems, through R10, have not activated EGN.)
- ▶ You must know the exact name of the defined RACF profile you want to check. In the example, this is `ogden.*`. Only data sets protected by the exact profile you specify are listed. This can be confusing for new users. In this example, the exact name of the data set profile is `'ogden.*'` (the single quote marks are not part of the profile name). This profile protects all the data sets with an HLQ of `ogden`. The *name of the profile* is a different thing than the *names of all the data sets* that might be protected by the profile.

5.53 Modifying RACF control in OpenEdition processes

When a traditional MVS program first invokes an OpenEdition service, the system will *dub* the environment, turning it into a UNIX process and/or thread (if a subtask). At this time, it looks up UIDs and GIDs and generally builds various control blocks that are not required by traditional MVS programs.

If your program then modifies its own security controls, using traditional MVS pointers and control blocks, these changes may not be seen by OpenEdition services. The most common case involves modifying the ACEE associated with a TCB. One symptom is a 4C6 ABEND.

Once a process is dubbed, you should use only C or UNIX-callable services to modify the security environment. (This discussion generally assumes you are working with authorized programs and are directly modifying MVS control blocks.)

5.54 How to *program control* libraries and files

RACF *program control* is a complex subject and we recommend you read the *RACF Security Administrator's Guide*, SG28-1915 for a better understanding. *Program control* can be used in a number of ways; we will briefly describe one here.

You (the local system administrator) can designate load modules, load libraries, or HFS executable files as *program controlled*. Commands to do this might be:

```
RDEF PROGRAM * ADDMEM('sys1.linklib'//NOPADCHK) UACC(READ)

RDEF PROGRAM mymod ADDMEM('my.load.lib'//NOPADCHK) UACC(READ)

extattr +p /u/ogden/mypgm
```

The first example makes all modules (indicated by use of the asterisk) in SYS1.LINKLIB program controlled. The second example makes only module MYMOD in the indicated library program controlled. The third command (issued from the OpenEdition shell) makes a single file program controlled. Other parameters are available when program controlling a module or library; the forms shown here are the most common forms.

For the first two examples, (1) the two libraries must also be protected by normal RACF data set protection, and (2) the WHEN(PROGRAM) attribute of RACF must be activated. (This is done by a SETROPTS WHEN(PROGRAM) command.) After changes to the PROGRAM class (such as the two commands shown here) SETROPTS WHEN(PROGRAM) REFRESH must be issued.

The R10 AD systems has the WHEN(PROGRAM) activated. If you use the TSO command SETROPTS LIST, the first few lines show the status of WHEN(PROGRAM).

An important point is that the PDS (or PDSE) containing one or more program controlled modules must also be protected by a normal RACF data set protection profile. Any user must have at least EXECUTE access to the data set before he can attempt to access the program controlled modules in the data set. In this sense, two profiles are needed: one for data set protection and one for program control. (HFS files do not require a data set profile, of course, and only the extended attribute bit plus an execute bit in the appropriate permissions field is needed.)

You must have special RACF permission to use the **extattr** command. This is explained in “How to set and display extended attributes” on page 230.

What does program control do? In the basic case we are describing here, it indicates that all the controlled modules are considered acceptable. This is often described in a number of ways. “These modules have been blessed by the systems programmers.” “These modules are trusted by the installation.” “These are the official modules of the installation.” The implication is that if these modules are used, *from the indicated libraries*, then there is no chance of a rogue module being included in the execution of a job that intends to use only program controlled modules. The whole concept depends on the local installation selecting only trusted libraries or modules to mark as *program controlled*.

The mechanism works like this:

- ▶ When an address space is created, it is *clean*. (*Clean* and *dirty* are key words in the discussion of program control.)
- ▶ When any new module is loaded, the program fetch mechanism of MVS checks whether it is *program controlled*. At the macro level, this means any LINK, LOAD, or XCTL function causes this check.
- ▶ When the first module that is not *program controlled* is loaded, the address space is marked *dirty* and remains this way for the life of the address space.
- ▶ A program can check the clean/dirty state of the address space and make logical choices based on the results. The implication is that a non-controlled module (which, by implication, is not a fully-trusted module) might alter the address space environment in some way as to corrupt the processing of later modules. If the address space is still *clean* when a later module gets control, that module can assume the address space environment is correct.

5.55 How to permit users to view OMVS segments

The OMVS segment in RACF contains a user's UID, his home directory path, and his initial program when he uses an OpenEdition shell. It may also contain various process limits. A normal user cannot view his (or anyone else's) OMVS segment. Knowing another user's UID is not normally seen as a security exposure. However, knowing all the userids that have UID=0 might be an exposure. In any event, a normal user cannot display OMVS segment data in the RACF data base.

The following RACF commands (issued by someone with RACF SPECIAL authority) will permit any user to display OMVS segment information for any user. You would probably not want to do this in a large production shop, but it might be useful in a small development environment.

```
SETROPTS GENERIC(FIELD) CLASSACT(FIELD) RACLIST(FIELD)
RDEF FIELD USER.OMVS.* UACC(READ)
SETROPTS RACLIST(FIELD) REFRESH
```

This will permit any user to list the OMVS segment data for any/all users. Usage might be (assuming the current user is P390Z, who does not have any special RACF authority):

LU	Note 1
LU *	Note 2
LU P390Z OMVS	Note 3
LU P390A OMVS	Note 4
LU * OMVS	Note 5

- ▶ A simple LU command lists basic RACF (but not OMVS) data for the current user.
- ▶ The LU * command requests a listing of all base RACF user information that the current user is permitted to see. This will normally be only his own RACF base information. Again, LU does not list OMVS segment data by default.
- ▶ LU P390Z OMVS lists the current user's base RACF data plus his OMVS segment information. (We assumed the current user was P390Z). In this case he sees additional information because he requested his OMVS data to be displayed and he has authority to display it because we granted all users authority to see OMVS segment data when we entered the RDEF command earlier.
- ▶ LU P390A OMVS asks to see the base RACF data plus the OMVS segment for P390A. The current user, P390Z, is not permitted to see the base RACF data for another user but he is permitted to see the OMVS data. This command will display only the OMVS data for P390A.
- ▶ LU * OMVS asks to list all base user + OMVS segment data that the current user is permitted to see. He is permitted to see only his own base data, but he is permitted to see OMVS segment data for all users. This command will produce a long list of OMVS segment data, with the current users base RACF information in his entry in the list.

Be careful with the last listing. Careless reading will associate the wrong userid with a given OMVS segment listing. (The userid for the following OMVS segment appears to be associated with the previous OMVS segment.)

This is a small example of using RACF FIELD controls. You can consult the *RACF Security Administrator's Guide* for much more extensive examples.

5.56 How to use TSO authorities profiles

A TSO user can be granted a number of additional authorities. For example, *JCL authority* is needed before the user can SUBMIT a job to MVS batch. The names of these authorities are not always logical and are carryovers from SYS1.UADS design. (If a TSO user does not have a TSO segment in RACF, but does have an entry in SYS1.UADS, the UADS authorities are used. You should ensure all your TSO users have TSO segments defined in RACF.)

RACF has a number of classes that manage various TSO authorities. (These classes must be active before they do anything. You can use the SETROPTS LIST command to display active RACF classes for your system. The list is near the beginning of the output from the SETROPTS LIST command.) Within each RACF class are various profiles that offer more specific control.

For example, class TSOPROC must contain a profile for every TSO logon procedure. The profile name corresponds to the member name in the procedure library. Using the AD system for examples, you could define several procedure authorizations:

```
RDEFINE TSOPROC ISPFPROC UACC(NONE)
RDEFINE TSOPROC BILLPROC UACC(READ)

PERMIT ISPFPROC CLASS(TSOPROC) ID(OGDEN)
PERMIT ISPFPROC CLASS(TSOPROC) ID(PATTI)
```

Here we have authorized two TSO logon procedures, ISPFPROC and BILLPROC. ISPFPROC has UACC(NONE); this means that each user intended to use this procedure must be individually (or by groups) permitted to that profile. Two sample PERMIT commands are shown. BILLPROC has UACC(READ), meaning that any TSO user can use this profile; no PERMIT commands are needed. (Most of these profiles involve binary decisions. Any access level other than NONE is treated as permitting access.)

Which should you use, UACC(NONE) or UACC(READ)? If you use NONE, you will need to issue PERMIT commands for every new user, to allow him to use one or more logon procedures. (A TSO user can select a logon procedure by changing a field on the password screen during logon.) Using NONE provides more administrative control, but makes more administrative overhead. For small AD development installations, we suggest READ is an appropriate level. The release 10 AD system uses READ for its TSO logon procedures.

Other examples of TSO authorities from the AD system are:

```
RDEFINE ACCTNUM A1          UACC(NONE)          (this is not in the AD system)
RDEFINE ACCTNUM ACCT#      UACC(NONE)
RDEFINE TSOAUTH JCL        UACC(NONE)
RDEFINE TSOAUTH OPER       UACC(NONE)
RDEFINE TSOAUTH ACCT       UACC(NONE)

PERMIT ACCT# CLASS(ACCTNUM) ID(PATTI)
PERMIT A1     CLASS(ACCTNUM) ID(PATTI)
PERMIT JCL   CLASS(TSOAUTH) ID(PATTI)
PERMIT OPER  CLASS(TSOAUTH) ID(PATTI)
```

Any account number (such as A1 or ACCT#) used for TSO logon must be authorized in the ACCTNUM class with a profile name matching the account number. Here we have defined two account numbers. Since the UACC is NONE, each user must be PERMITTED to whichever account numbers he is authorized to use.²³

²³ The system demands that each TSO user have an account number, even if you are not using any accounting functions. The AD system uses 'ACCT#' as an account number for all TSO users.

There are a number of preassigned profile names for the TSOAUTH class. These include JCL (permission to SUBMIT jobs), OPER (permission to issue MVS commands), and ACCT (permission to use the ACCOUNT command to manipulate UADS; it also can affect SDSF controls). As shown, each TSO user would need to be PERMITTED to the appropriate profiles.

The RALT command is used to change existing general resource profiles. If needed, we could change our system to make some of these profiles have UACC(READ). After doing this, we would not need to issue PERMIT commands whenever we added a new TSO user. We could use these commands:

```
RALT ACCTNUM ACCT# UACC(READ)          (let anyone use ACCT#)
RALT TSOPROC ISPFPROC UACC(READ)      (let anyone use this proc)
RDEF TSOPROC BILLPROC UACC(READ)     (we added this procedure)
RALT TSOAUTH JCL UACC(READ)           (let anyone submit jobs)
RALT TSOAUTH RECOVER UACC(READ)
SETROPTS RACLIST(ACCTNUM TSOPROC TSOAUTH) REFRESH
```

We did not change TSOAUTH OPER or TSOAUTH ACCT to UACC(READ) because you probably do not want these authorities to be universally available. However, you might want to provide OPER authority for all your users if you do not mind the potential exposures. With the R10 AD system, OPER plus ACCT will allow all users to view all SDSF output:

```
RALT TSOAUTH OPER UACC(READ)
RALT TSOAUTH ACCT UACC(READ)
SETROPTS RACLIST(TSOAUTH) REFRESH
```

You can read more about TSO authorization controls in *Security Server (RACF) Security Administrator's Guide (SC28-1915)*.

5.57 Accessing your own data sets

If the RACF option PROTECTALL is enabled, every data set must be protected by a RACF profile. (Unprotected data sets can be accessed only by RACF SPECIAL users.) The AD CD-ROM system does not use PROTECTALL. In general, considerable planning and preliminary work must be done before using PROTECTALL.

If PROTECTALL is not in use, then any user normally has ALTER access to all data sets that have his userid as the high level qualifier, whether there is a RACF profile for these data sets or not.

5.58 How to collect SMF data

The OS/390 CD-ROM systems are delivered with SMF recording disabled. This is a good default, since SMF recording can produce noticeable system overhead. The release 10 CD-ROM systems do have SMF output data sets defined (SYS1.MAN1, SYS1.MAN2, and SYS1.MAN3) with a reasonable amount of space defined. If you want to collect a small amount of SMF data, this may be sufficient. If you want to routinely operate with full SMF recording, you may need to allocate more or larger SMF data sets. If you do not know if you want SMF data, we suggest you leave recording disabled.

To enable SMF recording, edit SYS1.PARMLIB(SMFPRM00) and change the first parameter from INACTIVE to ACTIVE.

If you have added an additional SMF data set, change the DSNAME in this member parameter to include it. For example,

```
DSNAME(SYS1.MAN1,SYS1.MAN2,SYS1.MAN3,SYS1.MAN4))
```

SMF data sets traditionally have names based on SYS1.MANx, such as SYS1.MAN2, SYS1.MAN3, SYS1.MANX, and so forth. You could use this job to create another SMF data set:

```
//OGDENS JOB 1,OGDEN,MSGCLASS=X
//      EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN   DD *
        DEFINE CLUSTER (NAME(SYS1.MAN4),VOLUME(LOCAL2), +
                        NONINDEXED,CYLINDERS(75),REUSE, +
                        RECORDSIZE(4086,32767),SPANNED, +
                        SPEED,CONTROLINTERVALSIZE(4096), +
                        SHAREOPTIONS(2))
/*
```

Changes to SMF parameters will be effective with the next IPL, or you can make them effective immediately (after saving the changes in PARMLIB(SMFPRM00)), by issuing the OS/390 command SET SMF=00 from the MVS console.

Please note that IPLing can be difficult if you have an invalid SMFPRM00 parameter. We suggest that you copy the original SMFPRM00 member into SMFPRM99. If there is an IPL problem, you can specify that the alternate SMF member be used. (You will need to perform a *manual IPL* to do this.)

SMF can create considerable overhead, and small “MAN” data sets fill very quickly, if too much data is collected. You can control which SMF record types are recorded by using the SYS parameter in the SMFPRM00 member. One option is to specify SYS(TYPE(0:255)), which would record all SMF records. We recommend you do not use this parameter, unless you have a very specific reason for doing so.

The CD-ROM system has this SYS parameter:

```
SYS(NOTYPE(14:19,62:69),NOINTERVAL,NODETAIL)
```

This may be a useful specification if you need *some* SMF data. It collects all SMF records except records:

- ▶ 14 - Input data sets
- ▶ 15 - Output data sets
- ▶ 16 - SORT statistics
- ▶ 17 - Temporary data sets
- ▶ 18 - Rename operations
- ▶ 19 - A list of online DASD at IPL
- ▶ 62 - 69 - Various VSAM and catalog activities
- ▶ NODETAIL - A record is not produced for each TSO command executed
- ▶ NOINTERVAL - Intermediate records are not produced for long-running jobs

5.59 How to remove RMF automatic start

Earlier AD CD-ROM system automatically started RMF during the IPL process. Other prepackaged OS/390 versions may do the same. (The current AD system does not automatically start RMF.) Some RMF setups produce automatic RMF reports and these tended to fill the spool volume, especially if left running over a weekend, for example. You can prevent the automatic start of RMF by editing the PARMLIB member COMMNDxx (or member VTAMAPPL (or another script used by the VTAMAPPL program), depending on your setup) and deleting the line:

```
          COM='S RMF.RMF,,,MEMBER(09)      AUTOMATIC START OF RMF'
or       COM='S RMF.RMF'
or       S RMF                          (in VTAMAPPL)
```

Before you attempt this you should understand the implications and dangers involved with updating PARMLIB, see “How to work with PARMLIB” on page 110 for details on updating PARMLIB and how to prevent the situation where your change prevents OS/390 from starting. More information about using RMF is in “Getting started with RMF” on page 251.

5.60 How to increase SPOOL space

The AD systems have a limited amount of JES2 SPOOL space. It is sufficient for reasonable use, but is not sufficient if you (like many people) tend to leave HELD output in JES2 for long periods.

You must decide how much JES2 SPOOL space you need. Larger installations usually have at least one 3390 volume. You can place your new JES2 SPOOL data set on any volume, if there is space. We elected to create a volume just for SPOOL and the new volume was given the volser SPOOL1. Traditionally, the JES2 checkpoint data set is placed on a different volume to improve performance. For a typical small AD installation, this is probably not necessary.

It is possible to have multiple SPOOL volumes, for additional space and/or for performance. We think there is very little reason to do this with an AD system.

After selecting (or creating) a volume for SPOOL, and deciding how much space to allocate, follow these steps:

- ▶ Allocate new data sets. We used ISPF option 3.2 to create SYS2.HASPCKPT (30 cylinders, record format U, sequential (no directory blocks)), and SYS2.HASPACE (1080 cylinders, same attributes), both on SPOOL1. These spaces assume we are using a 3390-1 volume, which has 1113 usable cylinders. A small VTOC was placed on the first cylinder.
- ▶ Copy PARMLIB(JES2PARAM) to PARMLIB(JES2BACK), so you can start JES2 if something goes wrong with your new JES2 definitions. You can start JES2 with this backup parameter by using the command:

```
S JES2,HASPPARM=ALTPARM
```
- ▶ Change the SPOOLDEF parameters of JES2PARAM (change DSNAME=SYS2.HASPACE and VOLUME=SPOOL). Note that the VOLUME= parameter in SPOOLDEF is only five characters, even though the volser is six characters. Specify the first five characters in this parameter. (You should avoid having any other volumes that have the same first five characters of their volsers. JES2 will look for additional spool data sets on any volumes that have the same first five characters in their volsers.)

- ▶ Change the CKPTDEF parameters of JES2PARM (change to DSN=SYS2.HASPCCKPT and VOLSER=SPOOL1). (These names, SYS2..., are arbitrary but we recommend using easily recognized names such as the ones shown.
- ▶ Stop OS/390 and re-IPL with a manual mode that lets you control the start of JES2.²⁴ When asked to SPECIFY OPTIONS for JES2, enter COLD. This will cause JES2 to format the new spool data sets for use. Reply to the messages asking if you really want to format the spool data set.
- ▶ When formatting is complete, enter a \$\$ command to start normal JES2 functions. JES2 will be unresponsive until you issue this command. (The NOREQ parameter you normally enter to start JES2 implies the \$\$ function.)

5.61 How to print & clear SMF data sets

When an SMF data set is full, OS/390 will switch to another SMF data set and issue messages about the full data set. The following JOB can be used to copy SMF data to a user data set and clear the SMF data set for reuse:

```
//OGDENC JOB 1,OGDEN,MSGCLASS=X
//      EXEC PGM=IFASMFDP
//SYSPRINT DD SYSOUT=*
//DUMPIN  DD DSN=SYS1.MAN1,DISP=SHR
//DUMPOUT DD DISP=OLD,DCB=(BLKSIZE=27000,LRECL=32767,RECFM=VBS),
//      DSNAME=OGDEN.MY.SMF
```

The output data set (DUMPOUT) should be large enough to hold one of the SMF files. Using the AD CD-ROM system, it should be at least 30 cylinders. You might have multiple jobs like this, one for each of your SMF data sets. There are much more sophisticated ways to handle SMF output, and these are discussed in the formal SMF documentation (GC28-1457).

Note that the DCB parameters in the sample JCL are unusual. The LRECL is larger than the BLKSIZE. This is because spanned records (VBS) are used. Application programs that read SMF data must accept spanned records. The BLKSIZE is not important; half-track blocking on whatever disk you use is suggested.

5.62 How to clear SYS1.LOGREC

OS/390 automatically records hardware (and some software) error data, plus various statistical data, in SYS1.LOGREC. Eventually this data set becomes full, and console messages appear requesting the operator to take action. The hardware data recorded is not relevant for emulated devices or for a P/390 adapter itself. The data may be relevant for channel-attached devices.

The following job will print a brief summary of LOGREC data and then clear the data set. If you need LOGREC data (because you have channel-attached devices, or because you have a use for the software data, or perhaps you want the IPL and EOD information), you should refer to the EREP User's Guide and Reference (CG28-1378, or later) for information about printing detailed records and keeping history files

```
//OGDENC JOB 1,OGDEN,MSGCLASS=X
//      EXEC PGM=IFCEREPI,PARM='CARD'
//SERLOG  DD DSN=SYS1.LOGREC,DISP=OLD
//ACCDEV  DD DUMMY
//EREPPPT DD SYSOUT=*
```

²⁴ Any of the *cold start* IPL parameters supplied with the AD system could be used.

```

//TOURIST DD SYSOUT=*
//SYSIN   DD *
          SYSUM
          ACC=Y
          ZERO=Y
/*

```

5.63 How to change the system's IDs

OS/390 has a number of parameters where a system ID is specified. Some of the identifiers are:

- ▶ The SMF identifier. You can change this by editing the SID(name) parameter in PARMLIB(SMFPRMxx). This is a four-character identifier that appears in SMF data, console messages, and a number of other places. The name is specified in the SID parameter, for example SID(SYS1).
- ▶ The identifier in PARMLIB(IEASYSxx). This four-character identifier is used by GRS, in particular. (GRS is a function that allows ENQ/DEQ functions to be shared among several OS/390 systems connected by channel-to-channel adapters or sysplex connections.) The name is specified in the SYSNAME parameter, for example SYSNAME=P390. This can also be specified in the PARMLIB(IEASYMxx) member. If specified both places (IEASYSxx and IEASYMxx), the IEASYMxx name is used.
- ▶ The JES2 member name. You can change this by editing the MEMBER NAME=cccc field in PARMLIB(JES2PARM). This value is the name is specified in the MEMBER(1) NAME=xxxx parameter.
- ▶ The VTAM name used for NJE connections to the system. This is set in VTAMLST. In the AD CD-ROM system it is the name of the APPL statement for JES2 in VTAMLST(A0600).
- ▶ The TCP/IP host name. This is set in the TCP/IP parameters data set, which does not have a standard name. The AD CD-ROM system uses TCPIP.TCPIP.DATA for this parameter.
- ▶ The TCP/IP IP address, which is set in the TCP/IP PROFILE data set(s).
- ▶ The DFSMS configuration name, which may be relevant in a shared DASD environment.
- ▶ The VTAM subarea number (which can be considered a different type of system name), the SSCP name, and the APPN name are all set in SYS1.VTAMLST(ATCSTR00).
- ▶ The VTAM application names used for SNA access, such as NJE access, are set in various SYS.VTAMLST members.

There are an endless number of ways to combine the above names. In general, the IEASYSxx, SMFPRMxx, and JES2 names should be the same (and this name is limited to four characters). The TCP/IP host name and the VTAM NETID name might be the same. We suggest that you do not attempt to change your system name(s) unless there is a significant reason for doing so.

5.64 How to alter TSO timeout logoffs

After a period of inactivity, TSO will automatically log off a user. This period is set in PARMLIB(SMFPRMxx), by the parameter JWT(hhmm). JWT means Job Wait Time. The value in earlier AD CD-ROM systems is JWT(0030), which means that a user is forced off (with a 522 ABEND) after 30 minutes of no activity. Current AD CD-ROM systems have JWT(0400), which is 4 hours.

You can edit this PARMLIB member and change this value. Note that the value is normally four digits, in the form HHMM. A new value of JWT(0800) would establish a timeout of eight hours. After saving the altered PARMLIB member, the new value will be effective at the next IPL, or if you issue the OS/390 command SET SMF=xx, where xx is the suffix for the SMFPRMxx member name, to immediately activate the new parameters.

The JWT parameter also applies to batch jobs, if one should encounter a very long WAIT for some reason. Be certain you use a valid HHMM parameter. The value 0099, for example, is invalid and would cause problems the next time you IPL.

TCP/IP also has a TN3270 idle time parameter, specified in the TELNETPARMS section of the TCP/IP PROFILE. TCP/IP will drop the connection if this idle time is exceeded. TSO will continue the session, with the user disconnected, until the JWT time is exceeded.

The SMF job wait time does not apply to a program using OpenEdition services if the process is in a signal-enabled wait and was created through a *fork* function.²⁵ This permits parent processes and daemons to run indefinitely, but it also prevents normal timeouts of inactive users because an idle UNIX shell prompt is in a signal-enabled wait. You can use the TMOU environmental variable (placed in /etc/profile, for example) to provide a timeout function for idle shell users.

5.65 How to make a logon CLIST

```

BILL'S P/390 TOYSHOP

                                White System
                                Help 293-1715

IIIII TTTTT SSSSS 000000      XX  XX      XX
 I   TT  S    0 00      XX  XX      XX
 I   TT  SSSSS 0 00      XX  XX      XX
 I   TT  SSSSS 0 00      XX  XX      XX
 I   TT  S    0 00      XX  XX      XX
iiii  TT  SSSSS 000000      XX  XX  XX
                                XX  XXXX

-----
1. Remember to use BILLPROC for faster logons.
2. Next full backups scheduled for Friday evening.
3. Enter PAYRC, ACCTRCV, or ACCTPAY to start relevant applications.
4.
READY

```

Figure 5-26 Display produced by CLIST

In some circumstances, you might want to provide additional logon functions for a group of users. You can do this by (1) providing a unique logon procedure (in SYS1.TSOPROC, in an AD system) for these users, and possibly (2) automatically executing a CLIST from this procedure. A CLIST might produce a screen like the one shown Figure 5-26. This section addresses only the use of the PARM option in the EXEC PGM=IKJEFT01 statement that is normally used in a TSO logon procedure. The core statement of a logon procedure is something like this:

```
//IKJADMIN EXEC PGM=IKJEFT01,TIME=1440,
//          DYNAMNBR=25,PARM='%LP390',REGION=4096K
```

²⁵ If the environmental value _BPX_SHAREAS=YES is set when the user enters OMVS, the user's shell will be in the original TSO address space and the SMF JWT time will apply to it.

The CLIST you automatically execute may contain anything you want. The example here contains:

- ▶ A small, easily-recognized logo.
- ▶ A place for important announcements. (You would need to edit the CLIST to change the announcements.)
- ▶ CLIST code to automatically create an ISPF profile data set if this user does not have one.

The process for creating and using the CLIST is straightforward. Making a small, graphic logo might take more time than the rest of the actions. (The logo is optional, of course, but it helps people immediately identify the screen.) Your design should not use more than about 20 lines on the screen, since the default logon screen mode will be 24x80. The steps involved are:

- ▶ You will find that the standard user logon procedure on the AD CD-ROM system, SYS1.TSOPROC(ISPFPROC) already executes CLIST SYS1.CLIST(ISPFCL). This CLIST is very similar to the one listed in the next step. You might simply modify SYS1.CLIST(ISPFCL) to contain your messages (and logo), or you can create a new CLIST and leave the original one intact.
- ▶ Write your CLIST and store it in SYS1.CLIST, using a member name that does not exist, such as LP390. A sample CLIST is:

```

PROC 0 VOL(USER01)          <---- change as needed
  CONTROL NOMSG NOFLUSH
  WRITE BILL's P/390 TOYSHOP                               White System
  (.....WRITE statements to create your logo.....)
  WRITE -----
  WRITE 1. Remember to use BILLPROC for faster logons
  (.....more WRITES for announcements .....)
  PROFILE NOMODE MSGID PROMPT INTERCOM WTPMSG
  /*                                                              */
  /* ALLOCATION CLIST FOR ISPF PROFILE DATA SET                  */
  /*                                                              */
  FREE FILE(ISPPROF ISPTABL)
  SET &DSNAME = &STR(&SYSUID..ISPF.ISPPROF)
  ALLOC DA('&DSNAME') OLD FILE(ISPPROF)
  ALLOC DA('&DSNAME') OLD FILE(ISPTABL)
  IF &LASTCC ``= 0 THEN DO
    FREE FILE(ISPCRTE)
    DELETE '&DSNAME.'
    DELETE '&DSNAME.' NOSCRATCH
    ATTRIB ISPCRTE DSORG(PO) RECFM(F B) LRECL(80) BLKSIZE(3120)
    ALLOC DA('&DSNAME.') SPACE(2 4) TRACKS DIR(2) VOL(&VOL) -
      USING (ISPCRTE) FILE(ISPPROF)
    IF &LASTCC = 0 THEN -
      ALLOC DA('&DSNAME.') OLD FILE(ISPTABL)
    ELSE DO
      WRITE %%% UNABLE TO ALLOCATE OR CREATE ISPF PROFILE DATA SET "&DSNAM
      FREE FILE(ISPPROF)
      EXIT CODE(12)
    END
    FREE FILE(ISPCRTE)
  END
ELSE DO
  CONTROL MSG
  WRITE
  EXIT CODE(0)
  END
END

```

- ▶ Test your CLIST, using the TSO command:


```
EX 'SYS1.CLIST(LP390)' CLIST
```

- ▶ Decide if you want all TSO users to use your CLIST. If so, edit your standard logon procedure (which is SYS1.TSOPROC(ISPFPROC) in the AD system, unless you have changed it), and change the PARM field to point to your new CLIST. Place a percentage sign before the member name, for example %LP390. (This is TSO's way of indicating that the following is the name of a CLIST.)
- ▶ If you do not want all TSO users to use your CLIST, make a different logon procedure (which can be an exact copy of your standard logon procedure with only the PARM field changed), and have your selected users use this procedure. The procedure used during TSO logon is selected by a field in the logon screen. A user can overwrite this field (and the change is saved in RACF) and name your new logon procedure.
- ▶ The body of this CLIST is intended to allocate an ISPF profile data set if the user does not already have one. You are not required to do this. For example, you might decide that all your users should already have preallocated ISPF profile data sets and do not need this logic in the CLIST.

5.66 How to change the ISPF menu

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

0 Settings      Terminal and user parameters      User ID . . : P390
1 View          Display source data or listings      Time. . . . : 14:22
2 Edit          Create or change source data    Terminal. . : 3270
3 Utilities     Perform utility functions              Screen. . . . : 1
4 Foreground   Interactive language processing        Language. . . : ENGLISH
5 Batch         Submit job for language processing        Appl ID . . . : ISR
6 Command      Enter TSO or Workstation commands          TSO logon . . : ISP
7 Dialog Test  Perform dialog testing                    TSO prefix: P390
8 LM Facility  Library administrator functions          System ID . . : P390
9 IBM Products IBM program development products MVS acct. . . : ACCT#
10 SCLM        SW Configuration Library Manager        Release . . . : ISP
11 Workplace  ISPF Object/Action Workplace
M More         More IBM Product Options
```

Figure 5-27 ISPF Primary Options menu used with the AD system

ISPF panels (menus) are created by ISPF definitions. Products that use ISPF, such as SDSF or RACF, usually provide their own ISPF panel libraries. In release 10, the primary panel library for ISPF itself is ISP.SISPPENU. This library contains member ISR@PRIM, which is the primary ISPF menu panel.

ISPF has its own language for describing panels, and it is this language that appears in ISP.SISPPENU. (A starting reference is *Dialog Tag Language Guide and Reference*, SC28-1219.) Earlier version of ISPF had a simpler dialog language such that you could readily modify panel definitions without much effort. The current ISPF offers much more functionality than the earlier versions, but has a more complex dialog language. If you want to modify the standard ISPF panels, you should probably obtain the appropriate ISPF reference documentation.

Your release 10 AD system concatenates another library, SYS1.LOCAL.ISPFPNLS, before ISP.SISPPENU, and this local library has a modified version of ISR@PRIM. It is this modified version that provides the “M” option on the primary ISPF menu of the AD system. If you are planning to change it, you should keep this member name. Otherwise you must change the name in many CLISTs and logon procedures. This is the panel shown in Figure 5-27 on page 179. (You can usually find the name of the ISPF panel currently displayed by entering the command PANELID on the command line of the current panel; enter PANELID OFF to stop the display.) The ISR@PRIM panel references another panel, SYS1.LOCAL.ISPFPNLS(IBMPRODS), for the *second page* of the primary menu. If you want to change the ISPF primary menu, you will need to work with one of these two members.

If you want to change the primary menu panel, you should copy it (ISR@PRIM and possibly IBMPRODS) to another member in SYS1.LOCAL.ISPFPNLS (or any other library that is concatenated to these panel libraries) with a new, unique member name--such as ISR@LOCL--and experiment with changes to this member. You can test your new panel by exiting ISPF and using the command:

```
ISPSTART PANEL(panel_name)
ISPSTART PANEL(ISR@LOCL)                for example
```

assuming your new panel is in one of the libraries already allocated for ISPF use. There is also an option in the ISPF Primary Option Menu to test and validate a new ISPF menu; this is the Dialog Test option.

When you are ready to use your modified panel for production, copy ISR@PRIM to a backup name, such as ISR@BACK, and then rename your new member to ISR@PRIM. If your new menu fails, you can issue the command ISPSTART PANEL(ISR@BACK) from a TSO READY prompt to restart ISPF and repair whatever was wrong with your menu.

The core elements of a menu are the BODY section, containing the displayed menu, and the PROC section, containing the command associated with each menu item. For example:

```
)PANEL
.
.
)BODY
.
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
K OMVS         Start OMVS
.
.
)PROC
0,'PGM(ISPISM) SCRNAME(SETTINGS) '
1,'PGM(ISRBRO) PARM(ISRBRO01) SCRNAME(VIEW) '
2,'PGM(ISREDIT) PARM(P,ISREDM01) SCRNAME(EDIT) '
3,'PANEL(ISRUTIL) SCRNAME(UTIL) '
K,'CMD(OMVS) '
.
)END
```

The distributed panel descriptions contain many lines of code in addition to these core functions. Much of this additional code deals with color controls, CUI (action bar at top of screen), and PC interfaces to ISPF.

5.67 Understanding basic SDSF security

SDSF, used primarily to display SYSOUT data sets, has a number of different security options that are explained in *System Display and Search Facility: Customization and Security*, SC23-3807. Briefly, you have the choice of one of the following:

- ▶ Using an assembled module to control access characteristics for predefined groups of users,
- ▶ Using a PARMLIB member to control access characteristics for groups of users defined in the member, or
- ▶ Using RACF controls for very fine-grain control of access characteristics for individual users and groups.

Very fine resolution of control is possible, especially with the RACF option, but the setup and administration can be complex. If you need detailed controls for SDSF, we suggest you obtain the indicated manual and start from there.

The distributed OS/390 release 10 AD system uses a PARMLIB member (in conjunction with a started task named SDSF) to control SDSF security. The started task (named SDSF) contains a parameter that specifies a suffix for an ISFPRMxx member in PARMLIB. The AD system uses suffix 00. Extracts from PARMLIB member ISFPRM00 are shown here; you should examine the PARMLIB member for a full listing.

```

/*****/
/* OPTIONS Statement - Global SDSF Options */
/*****/

OPTIONS ATHOPEN(YES),      /* Use authorized open for datasets */
DCHAR('?'),               /* Command query character */
DSI(NO),                  /* Bypass ENQ for dynamic allocation */
FINDLIM(5000),            /* Maximum lines to search for FIND */
IDBLKS(4096),             /* HASPINDEX blocksize */
INDEX(ISF.HASPINDEX),    /* HASPINDEX dataset name */
JESDATA(VERSIONS),       /* Use checkpoint versioning */
LINECNT(55),              /* Print lines per page */
LOGLIM(0),                /* OPERLOG search limit in hours */
MENUS(ISF.SISFPLIB),     /* Panels dataset name for TSO */
NIDBUF(5),                /* Number of haspindx buffers */
NSPBUF(5),                /* Number of spool buffers */
SCHARS('*%'),            /* Generic and placeholder characters */
SCRSIZE(1920),            /* Maximum screen size */
SYSOUT(A),                /* Default print sysout class */
TRACE(C000),              /* Default trace mask */
TRCLASS(A),               /* Default trace sysout class */
UNALLOC(NO)              /* Do not free dynalloc data sets */

/*****/
/* GROUP ISFSPROG - System Programmers */
/*****/

GROUP NAME(ISFSPROG),     /* Group name */
TSOAUTH(JCL,OPER,ACCT),  /* User must have JCL, OPER, ACCT */
ACTION(ALL),              /* All route codes displayed */
ACTIONBAR(YES),           /* Display the action bar on panels */
APPC(ON),                 /* Include APPC sysout */
AUPDT(2),                 /* Minimum auto update interval */
AUTH(LOG,I,O,H,DA,DEST,PREF, /* Authorized functions */
  SYSID,ABEND,ACTION,INPUT,
  FINDLIM,ST,INIT,PR,TRACE,
  ULOG,MAS,SYSNAME,LI,SO,NO,PUN,RDR,JC,SE,RES),
```

```

CMDAUTH(ALL),          /* Commands allowed for all jobs */
CMDLEV(7),             /* Authorized command level */
CONFIRM(ON),          /* Enable cancel confirmation */
CURSOR(ON),           /* Leave cursor on last row processed */
DADFLT(IN,OUT,TRANS,STC,TSU,JOB), /* Default rows shown on DA */
DATE(MMDDYYYY),      /* Default date format */
DATESEP('/'),         /* Default datesep format */
DFIELD2(DAFLD2),     /* Sample alternate field list for DA */
DISPLAY(OFF),         /* Do not display current values */
DSPAUTH(ALL),         /* Browse allowed for all jobs */
GPLEN(2),             /* Group prefix length */
ILOGCOL(1),          /* Initial display column in log */
ISYS(LOCAL),         /* Initial system default for DA */
LANG(ENGLISH),       /* Default language */
LOGOPT(OPERACT),     /* Default log option */
OWNER(NONE),          /* Default owner */
UPCTAB(TRTAB2),      /* Upper case translate table name */
VALTAB(TRTAB),       /* Valid character translate table */
VIO(SYSALLDA)        /* Unit name for page mode output */

```

```

/*****/
/* GROUP ISFOPER - Operators */
/*****/

```

```

GROUP NAME(ISFOPER), /* Group name */
TSOAUTH(JCL,OPER),  /* User must have JCL and OPER */
ACTION(ALL),        /* All route codes displayed */
ACTIONBAR(YES),     /* Display action bar on panels */
APPC(ON),           /* Include APPC sysout */
AUPDT(2),           /* Minimum auto update interval */
AUTH(LOG,I,O,H,DA,PREF,DEST, /* Authorized functions */
      SYSID,ACTION,FINDLIM,ST,
      INIT,PR,ULOG,MAS,SYSNAME,LI,
      SO,NO,PUN,RDR,JC,SE,RES),

```

```

CMDAUTH(ALL),          /* Commands allowed for all jobs */
CMDLEV(7),             /* Authorized command level */
CONFIRM(ON),          /* Enable cancel confirmation */
CURSOR(ON),           /* Leave cursor on last row processed */
DADFLT(IN,OUT,TRANS,STC,TSU,JOB), /* Default rows shown on DA */
DATE(MMDDYYYY),      /* Default date format */
DATESEP('/'),         /* Default datesep format */
DISPLAY(OFF),         /* Do not display current values */
DSPAUTH(USERID,NOTIFY,AMSG), /* Browse authority */
GPLEN(2),             /* Group prefix length */
ILOGCOL(1),          /* Initial display column in log */
ISYS(LOCAL),         /* Initial system default for DA */
LANG(ENGLISH),       /* Default language */
LOGOPT(OPERACT),     /* Default log option */
OWNER(NONE),          /* Default owner */
UPCTAB(TRTAB2),      /* Upper case translate table name */
VALTAB(TRTAB),       /* Valid character translate table */
VIO(SYSALLDA)        /* Unit name for page mode output */

```

```

/*****/
/* GROUP ISFUSER - General Users */
/*****/

```

```

GROUP NAME(ISFUSER), /* Group name */
TSOAUTH(JCL),        /* User must have JCL */
ACTION(11,12,USER), /* Default route codes in log */
ACTIONBAR(YES),     /* Display action bar on panels */
APPC(ON),           /* Include APPC sysout */

```

```

AUPDT(10),           /* Default auto update interval */
AUTH(I,O,H,DA,ST,SE), /* Authorized functions */
CMDAUTH(USERID,NOTIFY), /* Command authority */
CMDLEV(2),           /* Command level */
CONFIRM(ON),         /* Enable cancel confirmation */
CURSOR(ON),          /* Leave cursor on last row processed */
DADFLT(IN,OUT,TRANS,STC,TSU,JOB), /* Default rows on DA */
DATE(MMDDYYYY),     /* Default date format */
DATESEP('/'),        /* Default datesep format */
DISPLAY(OFF),        /* Do not display current values */
DSPAUTH(USERID,NOTIFY), /* Browse authority */
ILOGCOL(1),          /* Initial display column in log */
LANG(ENGLISH),       /* Default language */
LOGOPT(OPERACT),     /* Default log option */
OWNER(USERID),        /* Default owner */
PREFIX(USERID),       /* Default prefix */
UPCTAB(TRTAB2),      /* Upper case translate table name */
VALTAB(TRTAB),       /* Valid character translate table */
VIO(SYSCALLDA)      /* Unit name for page mode output */

```

These parameters define three groups of users, with decreasing authority to display output with SDSF. These are:

- ▶ The ISFSPROG group, which can display anything and use all SDSF facilities, including issuing OS/390 commands through SDSF,
- ▶ The ISFOPER group, which can do almost all the same things, except display other users' SYSOUT data sets,
- ▶ The ISFUSER group, which can display only the jobs belonging to the user and whose jobname begins with the appropriate userid.

These specific names, ISFSPROG and so forth, are arbitrary. If you elect to change the parameters you could use any names. These names have no external significance outside SDSF and are not related to RACF groups.

Look at the TSOAUTH line of each of these groups. It has a field such as TSOAUTH=(JCL,OPER). If you look at the standard job we suggest you use to add new users, see “Adding new userids” on page 88, you will notice commands such as PERMIT *userid* JCL CLASS(TSOAUTH). This is the link between RACF and the SDSF controls. In the example here, if a user has JCL *and* OPER authority (via the TSOAUTH class in RACF), he will have all the SDSF authority of the ISFOPER group.

In practice, almost all TSO users should have JCL authority; it allows them to SUBMIT jobs from TSO. The questionable control, in regard to SDSF display authority, is with TSO OPER and ACCT authority. If a user has OPER and ACCT authority, he can display output for anyone's jobs. If he does not have OPER and ACCT authority, he can display output only for his own jobs. If he has OPER but not ACCT, he can see the JES2 output of other users' jobs, but not see the contents of SYSOUT data sets.

If you strongly encourage your TSO users to always use jobnames that begin with their userids, the OPER and ACCT authorities will not be an issue for looking at SDSF output.

This OPER authority is the same authority that permits a user to enter OS/390 operator commands. (These commands can be entered through direct TSO functions, as well as through the SDSF LOG display.) The ACCT authority is the same authority that permits a user to use the TSO ACCOUNT command. It is a little unfortunate that SDSF controls are keyed to TSO authorities that were meant for other purposes.

If you want to (1) allow users to examine SYSOUT data sets belonging to other users' jobs, but (2) you do not want to allow OPER authority, and (3) you do not want to spend much time learning about other ways to control SDSF security, then you might key the ISPOPER class only on JCL and ACCT. There is less practical exposure this way than allowing many users to have OPER authority.

SDSF uses the following libraries.

ISF.SISFJCL	/* Installation JCL	*/
ISF.SISFLOAD	/* Load modules library	*/
ISF.SISFLPA	/* Link Pack Area modules library	*/
ISF.SISFMLIB	/* Messages library	*/
ISF.SISFPLIB	/* Panels library	*/
ISF.SISFPSHF	/* Book shelves library	*/
ISF.SISFPUBS	/* Book library	*/
ISF.SISFSRC	/* Modules source code library	*/
ISF.SISFTLIB	/* Tables library	*/

Converting SDSF fully to RACF controls could provide much finer control of individual privileges, but takes some time and planning. If you need to do this, you must start with the manual (SC23-3808). In general, you will be working with these RACF classes:

SDSD, GSDSF, WRITER, JESSPOOL and OPERCMDS

and a number of profiles within these classes.

5.68 How to permit SDSF users to display all jobs

This section provides a quick solution to a problem; see “Understanding basic SDSF security” on page 181 for more details about SDSF security. Assuming the SDSF environment that is present in the release 10 AD system is present, the following RACF commands should enable the target user to see SDSF output for all users:

```
PERMIT OPER CLASS(TSOAUTH) ID(userid) ACC(ALTER)
PERMIT ACCT CLASS(TSOAUTH) ID(userid) ACC(ALTER)
SETROPTS RACLIST(TSOAUTH) REFRESH
(the target TSO user should logoff and logon again)
```

Unfortunately, this permission also permits the userid to issue any MVS command. Wide use of this authority is probably acceptable in a smaller, closed environment, but should not be acceptable in a production environment.

5.69 How to change JES2 definitions

JES2 uses member JES2PARM of PARMLIB to initialize its normal environment in the OS/390 system. Some of the parameters specify relatively static parameters, such as initiators or SYSOUT class definitions. Others specify controls that may require some tuning, such as the number of track groups, that influences how fast your spool will fill up. JES2 will display messages if it exhausts these dynamic elements, but you may need to change JES2PARMS to address the problem.

You can make many temporary changes to JES2 by using console commands. Other changes will require you to restart JES2, and a few will require you to cold start JES2. *Cold start* means that JES2 will format (erase) all the spool area, and you will lose anything stored in spool.

The following parameters can only be changed by a cold start:

```
JOBDEF    JOBNUM=  
NJDEF     OWNNODE=  
OUTDEF    JOENUM=  
SPOOLDEF  BUFSIZE=, DSNAME=, SPOOLNUM=, TGNUM=, TRKCELL=, VOLUME=
```

For example, expanding spool space (as described in “How to increase SPOOL space” on page 174) will require a cold start.

A cold start begins at the message:

```
$HASP426 SPECIFY OPTIONS - JES2 m.n.x
```

If you reply COLD, then JES2 will reformat the spool area and completely reset its contents. (If you reply COLD, JES2 will ask you several times to verify that you want to do a cold start. After processing the cold start, JES2 will ask you to ENTER OPTIONS. You must enter \$\$ (which is not a reply to an outstanding message) to start normal JES2 processing.)

JES2 is critical to OS/390 operation. If you cannot start JES2 (because of an error in JES2PARM, perhaps), then you cannot use TSO to fix your error. You must use an alternate set of JES2 parameters, or restore PARMLIB from a backup tape, or IPL another OS/390 system that can access the volume containing a good JES2PARM in order to continue. If you are careful to always keep a functional JES2PARM in PARMLIB(JES2BACK), you should be able to bypass any errors you make updating JES2PARM. Note that the member in JES2BACK need not be kept current. It simply must work to the extent that you can start JES2 and TSO, and then use them to fix whatever problem exists in JES2PARM.

To start JES2 using JES2BACK, stop JES2 and restart it with the command:

```
S JES2,HASPPARM=ALTPARM
```

Before doing this, be certain you have a working set of JES2 parameters in member JES2BACK. The release 10 AD system does not have a JES2BACK member. Assuming this release, you should copy the JES2PARM member of SYS1.ADCD10.PARMLIB to member JES2BACK (which you must create).

You will almost certainly want to change the distributed JES2PARM. Among the changes you might consider are:

- ▶ Define a 1403 printer for PRT1 and a 2540 card reader for RDR1.
- ▶ Reduce the number of initiators, or the number default active initiators.
- ▶ Reduce the SYNCTOL value, to reduce the waiting time to bypass artificial shared DASD delays.
- ▶ Reduce the number of job classes, SYSOUT classes, and SYSOUT priorities defined, although the existing definitions do no harm.

For further reference, see *OS/390: JES2 Initialization and Tuning Reference*.

5.70 How to work with MIH

This section will describe the *Missing Interrupt Handler* in your OS/390 System. You may need access to PARMLIB to customize and adjust MIH parameters. The IECIOSxx member is used by MIH, and you may need to add this member to PARMLIB if it is not already there. If you do not specify MIH controls, OS/390 will use default settings. These default settings are not always appropriate.

When OS/390 sends channel commands to a control unit, it waits for a response. If no response is received after a certain amount of time, a missing interrupt situation exists. The following conditions qualify as missing interrupts if the specified time interval has elapsed:

- Primary status interrupt pending
- Secondary status interrupt pending
- Start pending condition
- Idle with work queued
- Mount pending

The IECIOSxx member can also specify timing limits for devices. When the value that you specify for the I/O timing limit is greater than the value that you specify for MIH, normal MIH recovery will be in effect until the I/O timing limit is reached. Once the I/O request exceeds the I/O timing limit that you have specified, the system will abnormally end the I/O request.

Hot I/O refers to a hardware malfunction that causes repeated, unsolicited I/O interrupts. If those I/O interrupts are not detected, the system can loop or the system queue area (SQA) can be depleted. IOS tries to detect the hot I/O condition and perform recovery actions before the system requires a relPL. Recovery actions taken for a hot device depend on the type of device and its reserve status. IECIOSxx can specify hot I/O controls.

An example of an IECIOSxx member that specifies MIH, hot I/O, and timing controls is:

```
HOTIO DVTHRS=200
MIH DEV=(2E8-2FF,7300-7370),TIME=00:30
HOTIO DFLT111=(CHPK,CHPF),DFLT112=(CHPK,OPER)
MIH IOTIMING=00:12,DEV=(2E8-2FF,730-737)
MIH TIME=00:00,DEV=(180-187,230,B10-B17)
```

An IECIOSxx member in PARMLIB is ignored unless it is referenced in the IEASYSxx PARMLIB member you are using. This parameter should be added to IEASYSxx; be certain to include the comma:

```
IOS=xx,
```

The two alphanumeric characters (xx) are appended to IECIOS to identify the IECIOSxx PARMLIB member. If the IOS parameter is not specified, the system uses defaults for MIH processing and defaults for hot I/O processing.

The IECIOSxx member is normally read only during IPL. If you change it after IPL, you can issue a SET IOS=xx command to cause the revised values to be processed. You can also change MIH values through operator commands, but these are lost after the next IPL. A command example is:

```
SETIOS MIH,DEV=(560),TIME=10:00
```

For early releases of OS/390, the LAN emulated devices required the MIH time to be set to zero. OS/390 TCP/IP now does this automatically, so there is no need to create or alter IECIOSxx for this purpose. The 4mm tape drive used in MP3000 and P/390-based systems sometimes requires an unusually long time to rewind. This does not happen often, but it could trigger MIH action that is unwanted. We suggest setting the MIH time for a tape address that corresponds to a 4mm drive to 10 minutes. A PARMLIB(IECIOS00) member might be:

```
MIH TIME=10:00,DEV=(560)
```

If you add an IECIOSxx member, remember to insert IOS=xx lines in all the IEASYSxx members that might be used.

5.71 How to provide more paging space

OS/390 can support a large amount of virtual storage by using paging data sets on disk. Data sets named Page Spaces are used to contain this virtual storage. Another term often used to collectively describe these Page Spaces is Auxiliary Storage. The Auxiliary Storage Manager (ASM) manages paging operations.

As you increase the workload of your system you may run out of available Auxiliary Storage. If this occurs, you will see messages on the system console indicating Auxiliary Storage Shortages and OS/390 will attempt to remedy the situation (messages IRA200I and IRA201I). OS/390 will reject LOGONS and START commands and may delay the starting of certain initiators. OS/390 may or may not be successful in fixing the situation; in the worst case no work will be happening in the system and you may have to cancel some of the workload to allow the remainder of the system to proceed.

The solution to this problem is to increase the Auxiliary Storage available for OS/390 use by defining more Page Space. The status of Auxiliary Storage on your system can be determined using the system command:

```
D ASM
```

which will return a display like this:

```
IEE200I 13.38.34 DISPLAY ASM 109
TYPE FULL STAT DEV DATASET NAME
PLPA 82% OK 0122 SYS1.PLPA.PAGE
COMMON 28% OK 0122 SYS1.COMMON.PAGE
LOCAL 17% OK 0122 SYS1.LOCAL.PAGE
PAGEDEL COMMAND IS NOT ACTIVE
```

To define more Page Space you should use IDCAMS:

```
//P390D JOB 1,P390,MSGCLASS=X
//STP001 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE PAGESPACE ( +
NAME ( SYS1.LOCAL2.PAGE ) +
VOLUME ( PAGE01 ) +
CYLINDERS ( 700 ) )
/*
```

You should change the volume serial and number of cylinders to suit your requirements (and possibly the Page Space name). The new Page Space can be added to the present system immediately. The 700 cylinders in the example, assumed to be on a 3390 volume, represents about 500 MB of additional virtual storage paging space.

After creating the new paging data set, you can tell OS/390 to use it immediately by issuing an operator command. The following example assumes the same Page Space name as used in the above example:

```
PAGEADD PAGE=SYS1.LOCAL2.PAGE
```

To ensure the new Page Space remains a permanent part of your OS/390 system you should add it to the PAGE= parameter in all the IEASYSxx members you use in PARMLIB. Remember to change all the IEASYSxx members in all the PARMLIBs that you normally use. If you do not require the Page Space immediately, make the change to IEASYSxx and the Page Space will become active following the next OS/390 IPL.

Modern S/390 and z900 machines typically have enough memory such that paging performance is not as critical as it was a few years ago. Nevertheless, adding additional paging data sets (on separate disks, on separate channels, if possible) helps system performance to some degree.

5.72 Controlling virtual memory

Virtual memory is not “free” if it consumes too much paging space on your disks. This consideration is the primary reason you should define default region sizes for TSO users and batch jobs.²⁶ A *region size* limits the virtual memory available in the address space. The limit applies only to virtual memory that can be allocated by the user; it does not apply to commonly addressable storage such as PLPA or CSA. Also, a region size limitation has no direct effect on real memory use.

In earlier years, real memory was expensive and limited. Region size limitations were a moderately strong enforcement mechanism to ensure reasonable sharing of a valuable resource. The primary purpose of region size limitations today is to prevent runaway allocation of virtual memory.

OS/390 has 2 GB address spaces. If a user actually allocates (malloc, GETMAIN) all this space and does something with it, most of it will be paged out at some point. This consumes paging space on disks. A complete 3390-3 provides about 2.4 GB paging space; it would not take too many users with large address spaces to consume all the paging space available on typical OS/390 systems. Region size controls can prevent unintentional situations.

You can determine how much of your disk paging space is currently used by issuing the MCS operator command D ASM.

A region size limitation is not a security control. In a basic OS/390 system, a user can simply ask for whatever region size he wants. The system administrator can set default sizes, and these are usually large enough for 99% of whatever applications are run. An installation can install *exits* to enforce region size limitations. This is not uncommon, but has not been done in the AD system.

5.72.1 How to specify a TSO region size

Specifying a region size²⁷ can be confusing since 24-bit *below the line* and 31-bit *above the line* memory are involved. As we understand it, the rules are these:

- ▶ If you specify less than 16 MB, then the value applies only to storage below 16 MB. You will get the storage requested (if it is available, of course) plus 32 MB located above the 16 MB line.
- ▶ If you specify a value between 16 MB and 32 MB, you will get all the storage available below 16 MB, plus 32 MB above the 16 MB line.
- ▶ If you specify a value greater than 32 MB, you will get all the storage available below the 16 MB line, plus the requested amount above the 16 MB line.

There is one obvious problem with this method: you may not know how much memory (*all that is available*) exists below the 16 MB line and a GETMAIN for 24-bit storage may fail.

²⁶ You can also specify a region size in JCL (for batch jobs) or in the TSO logon password panel.

²⁷ This is old terminology, used before MVS provided multiple address spaces. A more correct terminology would be TSO Address Space Size, but the *region* term is still widely used.

In the case of typical AD development environments, there is usually no reason to have very restrictive TSO region sizes. For example, a 40 MB region size might be appropriate for most users. A *small* region size (a few megabytes) might be useful for stopping a program that loops with GETMAIN requests, but this is normally a rare case. Restrictive sizes (often 1 MB, with 4 - 6 MB for “large users”) were common for earlier MVS systems because real memory was much smaller on these systems, and paging rates were a primary performance concern. This is no longer the case with current machines, where the amount of real memory has increased faster than other performance factors.

Note that installation exits can be used to limit or restructure the region sizes specified for a TSO user (or for a batch job).

The TSO region size does not apply to an OpenEdition address space. In this case, the MAXASSIZE parameter from the BPXPRMxx member of PARMLIB specifies the maximum size.

The basic TSO region size parameters are normally set in the RACF TSO segment when a userid is defined. The SIZE() parameter sets the default size and the MAXSIZE() parameter set a maximum size. Both values are expressed in K bytes. The default SIZE parameter can be overridden at logon time, by changing the value in the TSO password panel. The override value cannot be larger than the MAXSIZE value. You can inspect the SIZE and MAXSIZE values (if you have SPECIAL authority) by the TSO command:

```
LU userid TSO
```

The largest possible value for SIZE and MAXSIZE (in the RACF TSO segment) is 2096128, which corresponds to a 2 GB region.

5.72.2 Understanding MAXASSIZE

MVS batch and TSO address spaces work with REGION parameters (on JOB and EXEC statements, and in the RACF TSO segment definition) to determine the maximum address space a user can obtain. OpenEdition does not use these controls. A Maximum Address Space Size (MAXASSIZE) parameter is specified in BPXPRMxx (in PARMLIB) and this sets the maximum size of the OpenEdition address space. TSO segment parameters are not involved. The default MAXASSIZE is 200 MB if it is not specified. The R10 AD system specifies MAXASSIZE as approximately 1 GB in the supplied BPXPRMxx members.²⁸

Changing MAXASSIZE in BPXPRMxx changes the maximum address space size for all OpenEdition users. (Be certain to change all the BPXPRMxx members you might use.) You can specify MAXASSIZE values for individual userids (in their OMVS segment definition) and this overrides the general BPXPRMxx specification.

Batch jobs run under BPXBATCH are limited by MAXASSIZE, and not by any REGION size specified in their JOB or EXEC statements.

5.72.3 How to select the number of page data sets

There are two issues here: performance and capacity. There are always at least three paging data sets: (1) PLPA, (2) COMMON, and (3) everything else. The AD R10 system supplies these three. See “How to provide more paging space” on page 187 for help in adding more paging data sets.

²⁸ We could not determine why this size was specified. It seems excessive, in that it would not stop runaway mallocs before they severely impacted the available paging space.

Conventional wisdom, for a small OS/390 system, calls for at least five paging data sets (PLPA + COMMON + three more), spread over as many channels as possible, and placed on volumes that are not often accessed for other purposes. However, informal observation of emulated I/O systems indicates there is little, if any, gain from spreading out the paging data sets--especially if RAID 5 disks are used on the underlying server.

P/390-based systems, using their emulated I/O, are not very suitable for high paging-rate situations. Informal observations indicated that sustained paging rates over 10-20 per second are "high" for these systems. EFS systems, using a different emulated I/O design, can apparently tolerate much higher paging rates and benefit from having paging data sets spread over several volumes.

The capacity or amount of paging space (regardless of how many paging data sets and volumes are used to hold this amount) has three areas of concern:

- ▶ There must be sufficient capacity for your normal usage. We needed to add additional paging space to the AD system when we multiple students compiling and running Java programs while DB2 and several CICSs were running, for example.
- ▶ A large SVC dump can temporarily consume a large amount of paging space, perhaps up to 2 GB. If this occurs when your paging data sets are already heavily used, you may have problems.
- ▶ If a program loop keeps acquiring virtual storage, it can tie up much of your paging space. Using appropriate region size parameters (or MAXASSIZE for UNIX System Services) can help control this exposure.

It is possible to add paging data sets while the system is running, but only if sufficient space is available on suitable volumes, of course. If OS/390 runs out of paging space, the operator may be able to recover the situation but this may be difficult. In general, running out of paging space is a serious situation for OS/390 and must be avoided if possible.

The OS/390 R10 and z/OS R1.1 AD systems, as distributed, have sufficient paging space for a modest number of TSO users (or batch jobs) doing traditional work. If you are working with applications exploiting large virtual memory (such as DB2, Java, and perhaps applications ported to UNIX System Services) we suggest you add several additional paging data sets. It is difficult to recommend the total amount of paging space needed, but 2 or 3 GB would certainly not be excessive.

5.73 How to compress a PDS

Partitioned Data Sets (PDSs, libraries²⁹) are fairly primitive structures. They consist of smaller internal files, each ending with an end-of-file marker (which is not readily visible, but is a special disk record). The first internal file is a *directory* that contains the names of the library members and a pointer to each one. The directory can also contain a limited amount of additional information for each member, and this is used for various purposes by different applications.³⁰ When you first allocate a PDS, you must specify the fixed size of this initial directory internal component.

²⁹ PDSEs are also known as libraries. PDSEs have a more complex internal structure and do not require compressing. In some cases (such as some of the ISPF panels) IBM attempts to distinguish between PDSs and *libraries*; in these contexts a *library* means a PDSE. However, the use of the term *library* for a PDS is common.

³⁰ ISPF statistics, for example, are kept in directory entries.

In general, you cannot directly update a member of a PDS. (This can be done by special programming, but such usage is not common.) If you edit a member, the altered version is written at the end of the PDS, the directory is changed to point to this updated member, and the disk space used for the original member is lost. Every time you edit or otherwise replace a PDS member the changed member is written at the end of the PDS and the previous space is lost. (By *lost*, we mean that no directory entry points to it and there is no easy way to use that space.) Eventually, you will exhaust all the space allocated to the library and you will get an ABEND when you try to save into the PDS.

You can recover the lost space in a PDS by using IEBCOPY to *compress* it. This is a standard function of IEBCOPY, and consists of *moving up* all the members in the PDS so that they reuse the lost gaps in the data set. One way to do this is to use the ISPF 3.1 option. On this panel, enter the Data Set Name you want to compress and C in the Option field. Press Enter and wait for the completion message.

The ISPF option 3.4 provides another way to compress a PDS. In any data set listing produced under 3.4, enter the letter Z as a line command. ISPF will call IEBCOPY to compress the PDS.

Compressing a large PDS that has been actively updated for some time can cause considerable disk activity and take some time. In principle, there is some risk during a compress operation. If the PDS directory has been corrupted, the compress operation may leave the whole PDS unusable. This is rare, but experienced systems programmers ensure that they have sufficient backups before compressing a major, critical library.

If the PDS is large, you may prefer to compress it by using a batch job such as this:

```
//P390COMP JOB 1,MARK,MSGCLASS=X,NOTIFY=*
// EXEC PGM=IEBCOPY,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(10,10))
//DDCOMP DD DSN=MY.PDS,DISP=OLD
//SYSIN DD *
        COPY INPUT=DDCOMP,OUTPUT=DDCOMP
/*
```

You should *always* use DISP=OLD when compressing a PDS. This can sometimes be a problem with system data sets that are allocated as DISP=SHR by some long-running system component. You will need to devise a way to unallocate these before compressing them.

Any PDS that is changed (members altered or replaced) will accumulate lost space. This applies to system data sets, such as SYS1.PARMLIB, and to load libraries, source libraries, procedure libraries, and any other kind of PDS. If you make enough changes to SYS1.PARMLIB and SYS1.PROCLIB, they will run out of space, the same as your own TSO libraries. You should compress these system libraries after a substantial number of changes.

You can use ISPF option 3.4, followed by an S line command, to see how much of its allocated space is being used by any PDS. If the allocated space is nearly full, you should compress the PDS. Of course, if new material is being added to the PDS, even the compressed version can fill the allocated space.

If you copy a PDS to a new PDS, the copy operation produces, in effect, a compressed new copy.

You may need to be careful compressing system data sets that are possibly in use by other jobs. In general, the DISP=OLD locks (in JCL or the TSO equivalent) will prevent problems, but not always. It is fairly common practice to compress system data sets when there are no other users, and to follow this by an immediate re-IPL.

Note that the term *compress*, in this context, does not mean compressing data. It means removing wasted space in the data set, and is sometimes informally known as *de-gassing* a PDS. Also note that PDSEs, as opposed to PDSs, do not require compressing. The PDSE internal structure is more complex, and automatically recovers space left by updated members.

5.74 How to allow more TSO sessions

By default, the release 10 AD CD-ROM system permits 10 concurrent TSO sessions. There are two primary elements that can limit the number of sessions. You can change them as follows:

- ▶ Edit PARMLIB(TSOKEYxx)

You may find a USERMAX=nn parameter. If it is not present, the default is USERMAX=40. If the value or default is not what you want, change it. The AD systems use the TSOKEY00 member. (The release 10 system has no values defined in this member, which means that all the default values are taken.)

- ▶ Edit VTAMLST(A0600)

You will find stanzas named A06TSO01, A06TSO02, up to A06TSO10. Create more of these stanzas, named A06TSO11 through A06TSOnn. In each stanza, change the ACBNAME to contain the appropriate count number. Be certain you copy the continuation character (column 72) and comma when creating the stanzas. Note that there are two fields that must be updated in each stanza: the label field and one other.

Be certain to edit the right PARMLIB and VTAMLST. (In the AD release 10 system, these are SYS1.ADCD10.PARMLIB and SYS1.LOCAL.VTAMLST.) You can stop TSO and VTAM and restart them, or simply re-IPL the system, to have the increased number of TSO users.

Note that the number of TSO users (and CICS/IMS users) connected via TCP/IP TN3270 is limited by the number of LUs defined for TCP/IP. Check the TCP/IP PROFILE (TCPIP.PROFILE.TCPIP for the AD system) and find the DEFAULTLUS stanzas. The current AD system has 30 defined. You can add more, using the same pattern. These must correspond to LUs defined in VTAMLST (or alternate). For the AD system, member A0TCP in SYS1.LOCAL.VTAMLST contains the matching LUs; you can increase the number by following the existing pattern.

If you are uncertain about which VTAMLST libraries are being used, inspect SYS1.PROCLIB(VTAM) and check the data sets defined in the VTAMLST DD statement. The last several AD releases have SYS1.LOCAL.VTAMLST concatenated before SYS1.VTAMLST.

5.75 How to manage dump data sets

Dump data sets are used by OS/390 to take *system* dumps, also called SVC dumps. The information dumped to these data sets is typically used by IBM technical support to debug system problems, although it is also sometimes used for local debugging, as well.

Do not confuse these system dumps with application program dumps. Dumps associated with an application go into data sets with DDnames such as SYSABEND, SYSUDUMP, or SYSMDUMP. System dumps are requested by OS/390, on behalf of itself, when it detects a possible problem in system code.

Two methods are available for allocating and managing system dump data sets. The two methods are preallocated dump data sets and automatically allocated dump data sets. Traditional MVS systems used preallocated dump data sets, and sometimes also provided automatically allocated dump data sets in case the preallocated ones were full. *The release 10 AD system uses only automatically allocated dump data sets.*³¹ Overviews of both methods are presented here.

5.75.1 Preallocated dump data sets

Preallocated dump data sets must be allocated prior to use by the system dump routines. This allocation is usually done, once, when the OS/390 system is built from the IBM distribution tapes. The allocation is normally not altered after this; the data sets are reused many times. They use a naming convention of *SYS1.DUMPnn* where nn can be any decimal number from 00-99.

You can determine the status of the dump data sets by using the operator command:

```
DISPLAY DUMP,STATUS
D D,S                (abbreviated format)
```

You can determine to some degree what is in the dump data sets by displaying the titles of the dumps with the operator command:

```
DISPLAY DUMP,TITLE
D D,T
```

At some stage you will want to clear one or more of these dump data sets. You may have examined the dump, with IPCS, or simply do not want it.

To clear a single dump data set:

```
DUMPDS CLEAR,DSN=nn
DD CLEAR,DSN=nn        (abbreviated form)
```

To clear multiple data sets (using the abbreviated form of DUMPDS):

```
DD CLEAR,DSN=(nn,nn)
DD CLEAR,DSN=(nn-nn)
DD CLEAR,DSN=ALL
```

Clearing one of the preallocated dump data sets does not delete the data set; it deletes the logical contents, making it ready for another dump. If all the dump data sets are full (and you are not using the other alternative of automatically allocating dump data sets), and OS/390 wants to write another dump, the operator will receive messages about the full dump data sets and asking permission to discard the new dump that the system is attempting to create.

5.75.2 Automatically allocated dump data sets

Automatically allocated dump data sets (as opposed to the preallocated data sets just described) are allocated at the time the system dump routines take the dump. There can be any number of these, since OS/390 will simply allocate another one for the next dump. (This has obvious implications for disk space management; you must manually delete unwanted dumps.) They are named according to a pattern determined by the installation or using a default name pattern. The pattern uses system symbols to resolve to a data set name that is unique and meaningful to the installation. For example name pattern for the z/OS R1.1 AD system is:

³¹ The R10 system (in PARMLIB) specifies that preallocated dump data sets are to be used, but does not provide any preallocated dump data sets.

```
SYS1.&SYSNAME..DMP&SEQ  
SYS1.P390.DMP00000 (example)
```

Where &SYSNAME and &SEQ are system symbols (&SEQ must be used in the name pattern). Full details for using this facility can be found in *OS/390 System Commands*. There are no operator commands to list automatically allocated dump data sets, or to list their titles.

If you use automatic dump data sets, the initialization commands are usually in PARMLIB(COMMNDxx), although an operator can enter the commands directly at the OS/390 console. The AD system COMMNDxx members include the following:

```
DD ADD,VOL=OS39M1  
DD NAME=SYS1.&SYSNAME..DMP&SEQ  
DD ALLOC=ACTIVE
```

To display and manage these data sets you must use the facilities of TSO, ISPF and IPCS. For example, to determine what dump data sets have been taken and assuming you are using the AD name pattern you would use ISPF option 3.4 with a Dsname level of SYS1.P390 to list the data sets.

When you have determined that a dump is not required, it can be cleared by deleting the data set. Checking for the existence of new dump data sets, and deleting unwanted ones is one of a system programmers normal chores.

Earlier versions of the AD system used a name pattern that included the date and time. This made the data set names more complex, but was useful in immediately determining the date of the dump.

5.75.3 Security implications of system dump data sets

A system dump (like an application dump, which is a different topic) might contain confidential information. You might want to create a generic RACF profile to protect all dump data sets, with UACC(NONE). This is seldom done for smaller development systems, but you should be aware of the potential security exposure.

5.76 How to use job and print classes

Job classes and output classes are arbitrary names used to segregate jobs to be run and output that may be printed. The classes have names such as A, B, C, 1, and so forth; single-character names that have no intrinsic meanings. They are defined in PARMLIB(JES2PARM), or an alternate member you have selected, and you are likely to want to change the definitions.

A batch job always has a job class. It is included in the JOB statement. If no class is selected, the default is class A. A job has only a single class. Initiators, the OS/390 functions that run batch jobs, service one or more job classes. Multiple initiators can service overlapping sets of classes. For example, initiator 1 might handle job classes D, E, and A; initiator 2 might handle job classes A, C, and X. Again, the names have no intrinsic meanings, other than class A being the default class.

Some installations have elaborate definitions for job classes. For example, there might always be two initiators with class K assigned as the first choice (highest selection priority) and class K is reserved for urgent accounts-payable jobs. Smaller installations, such as a typical AD user, might have no special definitions and default all jobs to class A, with two to four initiators running class A jobs. Other than using class A for the default, there are no standards or conventions for job classes.

The R10 AD system does not define specific job classes. Instead, it permits the use of all possible job classes. (There are only 36, if you avoid special characters.) Key characteristics are:

All Job Classes:

```

AUTH = ALL    (jobs can include all operator commands)
ACCT = NO     (no accounting field is required)
COMMAND = VERIFY
BLP = NO      (you might need to change this)

```

INITIATOR	Job Classes	Status.....
1	A	Started automatically
2	AB	Started automatically
3	ABC	Started automatically
4	ABCDE	Started automatically
5	ABCD	Started automatically
6	ABCD	Started automatically
7	ABCD	Started automatically
8	ABCD	Started automatically
9	ABCD	Started automatically
10	ABCD	Started automatically
11	ABCD	Drained
12	ABCD	Drained

This means, for example, that initiator 3 is started when JES2 is started; it will run any job class A jobs that are in the job input queue. If there are no class A jobs, it will run class B jobs, and so forth. Ten active initiators is excessive for a P/390 system but might be sometimes suitable for a larger MP3000; the drained initiators do no harm.

Earlier systems provided several specific job classes with BLP authority. The R10 system does not do this. You may need to define (by editing JES2PARM) a specific job class that permits BLP. (BLP means Bypass Label Processing. A BLP parameter can be used in JCL statements; the JCL parameter is ignored unless the initiator also is authorized for BLP processing. BLP is normally used with specialized tape utility programs.)

You can use the following operator commands to work with initiators:

COMMAND	Purpose.....
\$DI	Display initiator status
\$PI3	Drain (stop) initiator 3
\$SI7	Start initiator 7

You can also use operator commands to change the job classes assigned to an initiator. Permanent changes are made by editing the JES2PARM member.

Job classes, initiator names (1, 2, 3,.. in the above example), and SYSOUT classes all use single-character names, but these names do not imply any relationships. For example, a job with job class J might run in an initiator named J and produce SYSOUT=J output, but this is all coincidence. Another job with job class J might run in initiator K and produce SYSOUT=X output. *There is no relationship between job class names, initiator names, and SYSOUT class names.*

SYSOUT classes are typically used to direct output to certain printers, or certain types of printers. For example, SYSOUT=A might be used for line data, while SYSOUT=J might be used for AFP (LIST3820)page printing. SYSOUT=A is a default class; JCL is often printed in SYSOUT=A. SYSOUT=B is usually defined as a punch; very few installations have card punches, but this format is sometimes used for RJE/NJE purposes. By convention, SYSOUT=Z is often a throwaway class that is immediately discarded. (This is often used by started tasks.)

The release 10 AD system has these definitions:

OUTCLASS	Characteristics....
A	write, print
B	write, punch
C	write, print
D	write, print
H	hold, print
J	write, print
K	hold, print
L	write, print
O	hold, print
X	hold, print
Z	dummy, hold
5	dummy, purge
9	dummy, purge

Printers: no printers are defined in the R10 AD CD-ROM JES2PARAM

A small AD installation might use only three SYSOUT classes:

```
SYSOUT=A    to actually print jobs (probably via a PC printer)
SYSOUT=X    for HELD output; to view via SDSF
SYSOUT=Z    to discard output for started tasks
```

You can redefine SYSOUT classes any way you like, but we suggest retaining the definitions for A, B, and Z. The AD CD-ROM, as distributed, cannot print anything. To define and start a line printer, do the following:

1. Define an emulated 1403 printer, if appropriate. We assume address 00E in this example.
2. Copy PARMLIB(JES2PARAM) to PARMLIB(JES2BACK)³².
3. Edit PARMLIB(JES2PARAM).
4. Find the PRT(1) line, which is a comment line.
5. Remove the comment characters (before and after the line), to make the line *active*.
6. Remove the FCB parameter.
7. Change the class field to CLASS=C. (Or whatever you like. Class A tends to accumulate unwanted files that you probably do not want to print.)
8. Change the unit to UNIT=00E (which is generated as a 1403 printer in the AD system).
9. Save the changed member.
10. Restart JES2. (The easiest way to do this is to shutdown the system and restart it.
11. When the system restarts, use the \$SPRT1 command to start the JES2 printer.
12. Submit a job with SYSOUT=C (or whatever class(es) you specified for the printer).
13. The SYSOUT=C output should be sent to your server for printing, if you are using an emulated printer.

A batch job can use multiple SYSOUT classes, although this is usually done only if there are special printer characteristics involved. For example:

```
//JOB1 JOB 1,BILL,MSGCLASS=A
..
```

³² This is to ensure that you have a usable JES2 parameter member. If you already have copied a working JES2 parameter member to JES2BACK you do not need to do it again.

```
//CHECKS DD SYSOUT=K
//REPORT DD SYSOUT=A
```

might be used if the printer associated with class K has preprinted checks (or some other special form) ready for use. A very common usage for the AD system is:

```
//JOB2 JOB 1,BILL,MSGCLASS=X
..
//SYSPRINT DD SYSOUT=*
//REPORT DD SYSOUT=*
//XYZ DD SYSOUT=*
```

where SYSOUT=* means to use the same SYSOUT class as used by the MSGCLASS operand in the job statement. SYSOUT class X, in the AD system, is a HELD class. You would run your job, and then inspect the HELD output with SDSF. You could then use SDSF facilities to change the X output to some other class if you want to print it.

For further reference please see *JES2 Initialization and Tuning Guide* and *JES2 Initialization and Tuning Reference*. See “How to change JES2 definitions” on page 184 for comments about changing JES2PARM.

5.77 How to obtain JOB ENDED messages

JES2 produces a JOB ENDED message whenever a batch job ends. This message, along with many other nonessential messages, is often suppressed on larger MVS systems in order to reduce total console traffic. On a small system, such as a typical AD system, you may find these messages useful.

Edit PARMLIB(MPFLST00). Find the line with \$HASP395, and delete that line.³³ Save the altered member. When you next IPL, the Job Ended messages will be displayed. You can also activate the altered MPFLST by issuing the OS/390 command:

```
SET MPF=00
```

5.78 How to use TSO Edit

This section will describe briefly the process you need to follow to use the EDIT TSO/E command. ISPF offers a much better editing function than the TSO EDIT command, and the TSO EDIT command is seldom used. However, there may be cases when ISPF is not working correctly for some reason, and you need an editor to repair the problem. The EDIT command may be useful in such circumstances.

EDIT is a fairly simple line editor and supports data sets that have one of the following formats:

- ▶ Fixed blocked, unblocked, or standard block with or without ASCII or machine records format.
- ▶ Variable blocked, or unblocked, without ASCII or machine control characters.

Using the EDIT command and its subcommands you can create, modify, store, submit, retrieve, and delete data sets with sequential or partitioned data set organization. You can not use TSO edit with PDSE data sets.

To create a new member into a partitioned data set use the following command:

³³ More recent AD releases have done this already, so the JOB ENDED messages are displayed.

```
EDIT 'yourid.tso.cntl(newmemb)' EMODE TEXT
```

The above command³⁴ can be entered from the TSO foreground panel or the READY prompt. After you press Enter you will be notified that newmemb was not found in the partitioned data set and the assumption will be NEW; at that time you will be prompted with the first line number. After entering text lines, the screen will appear similar to that in Figure 5-28.

```
edit 'p390.tso.cntl(editmore)' imode text
ikj52320i data set or member not found, assumed to be new
input
00010 This is a test member to illustrate the use of the edit command
00020 and its subcommands. To create this member I used the following
00030 command:
00040     edit 'p390.tso.cntl(editmore)' imode text
00050 After that I got the information message IKJ52320I saying
00060 the data set or member was not found and the assumption to be
00070 new. At the upper left corner I have an "input"
00080 indicator that indicates I'm working in edit input mode and the
00090 first line prompt (number) was displayed. After this, I .....
```

Figure 5-28 Sample of TSO EDIT command input

At this time you can start entering text; note that we used the operand TEXT to specify to the system we are introducing text data, there are other alternatives.

Key elements for using EDIT include:

- ▶ There is the concept of the *current line*; the line number * (asterisk) refers to the current line.
- ▶ A command without a line number, or with an asterisk for the line number, refers to the current line. Commands with explicit line numbers do not refer to the current line.
- ▶ You are usually in either INPUT or EDIT mode.
- ▶ In INPUT mode, the system will prompt you to enter the next line by displaying the number of the next line.
- ▶ You enter INPUT mode by using the INPUT command.

```
INPUT          (at end of current file)
INPUT *        (after current line number)
INPUT nnn      (after line nnn)
```

- ▶ Press Enter twice (with no input) to exit from INPUT mode.
- ▶ Useful commands (in EDIT mode are)

```
BOTTOM          (go to last line of file)
CHANGE 30 /old/new (change text in line 30)
CHANGE 30 'some text' 'other text' (blank characters included)
CHANGE 10 999 'old' 'new' (change all lines in range 10-999)
COPY 30 60 90    (copy lines 30-60 to line 90 (or after line
                  90, if 90 already exists))
DELETE 10        (delete line 10)
DELETE 500 2000 (delete all lines in range 500-2000)
DOWN            (move current pointer to next line)
DOWN 25         (move down 25 lines)
FIND 'this data' (start search from current line; use
                LIST * to display found line)
INPUT *         (input after current line)
```

³⁴ As with many TSO commands, you can omit the single quotes around the data set name if the data set name begins with your current PREFIX. Your PREFIX is usually your userid.

INPUT * R	(input and replace current line)
LIST *	(display current line)
LIST 10 99999	(list lines 10-99999)
MOVE 12 30 500	(move lines 12-30 to (or after) 500)
RENUM	(renumber file)
SAVE	
SAVE 'another.data.set'	(save under a different name)
SUBMIT	(submit batch job)
UNNUM	(unnumber lines -- limited use)
END	

There are many more commands and variations of commands for EDIT. For more information see *TSO/E V2R5 Command Reference*.

5.79 How to make a trivial IDCAMS job

IDCAMS is the OS/390 program that works with catalogs and VSAM data sets. You will need to use IDCAMS for many purposes. You can issue IDCAMS commands as TSO commands (from the READY prompt, or from ISPF option 6). For example, the LISTCAT command you may have already used is an IDCAMS command. Some IDCAMS tasks are suitable for use through TSO IDCAMS commands. However, many IDCAMS tasks that an administrator needs are better done as batch jobs. One reason is that IDCAMS output is often lengthy and not well formatted for viewing as the output from a TSO command. If you use batch jobs, you can scroll and view the output as often as you like—and even print it if necessary.

A batch job for IDCAMS can be very small. For example:

```
//P390I JOB 1,OGDEN,MSGCLASS=X
// EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
LISTCAT UCAT ALL CAT(CATALOG.OS390.MASTER)
/*
```

Note that the IDCAMS command(s) must not start in column 1. An additional reason for using IDCAMS through batch is that a batch job does not attempt to use your TSO userid as an automatic prefix for data set names. Many TSO commands do this, and this is seldom what you want when working with IDCAMS. (The automatic use of a userid as the HLQ of a data set name can be controlled by the PREFIX parameter of the TSO PROFILE command, but changing this can be inconvenient and the current prefix in use is sometimes not obvious.)

We suggest you create a small batch job, like the one shown, in one of your TSO libraries. You can overwrite the command contained in the job and submit the job whenever you need to use IDCAMS functions.

5.80 How to view SYSOUT with ISPF

ISPF provides an alternative to SDSF for viewing output. This is the “3.8” option. That is, select “3 Utility” and then select option “8 Outlist”. It produces the screen shown in Figure 5-29 on page 200.

```
Menu Utilities Help
-----
                                Outlist Utility

Option ==>

    L List job names/id(s) via the TSO STATUS command
    D Delete job output from SYSOUT hold queue
    P Print job output and delete from SYSOUT hold queue
    R Requeue job output to a new output class
blank Display job output

For Job to be selected:
  Jobname  . . P390A
  Class    . . .
  JobID    . . . JOB00473

For Job to be requeued:
  New Output class  . .
```

Figure 5-29 Outlist utility screen

In order to proceed you must select one of the options presented in the panel; if you choose option L then a list (in TSO line mode) will appear showing all the jobs for your user ID that are in the JES2 spool. Remember, the jobs must have originated from your user ID or must start with your user ID. A list similar to the following will appear when you select L (assuming you are userid P390):

```
IKJ56192I JOB P390Z(JOB00068) ON OUTPUT QUEUE
IKJ56192I JOB P390A(JOB00069) ON OUTPUT QUEUE
IKJ56192I JOB P390B(JOB00070) ON OUTPUT QUEUE
IKJ56192I JOB P390C(JOB00071) ON OUTPUT QUEUE
```

To display output for a job, enter the Jobname or the JobID in the appropriate field. The JobID is useful if there are several jobs with the same job name. You can use the Status function to determine the IDs for all the jobs with a given name.

If you select option P a panel like the one shown in Figure 5-30 on page 201 is displayed. You can also use the option to requeue the job. You would normally move it from a HELD class (such as class X in the AD system) to a printed class (such as class A).

```

                                Foreground Print Options
Option  ==>

    PK Print data set and keep      K Keep data set (without printing)
    PD Print data set and delete    D Delete data set (without printing)

    If END command is entered, data set is kept without printing.

Data set name . : P390.SPF113.OUTLIST

Print mode . . . BATCH              (BATCH or LOCAL)

SYSOUT class . .
Printer id . . .                    (For local printer)

Job statement information:          (Required for system printer)
==>
==>
==>

```

Figure 5-30 ISPF - foreground print options

You will be able to view all jobs run with your user ID or all those whose name begins with your user ID. You will not get access to other jobs.

5.81 How to make a simple HCD change

A step-by-step walkthrough using HCD to alter a device definition may be helpful. This was originally provided by Gordon Chamberland, of Interprocess Systems, Inc. It uses a particular example of redefining a tape drive, but the general process applies to most HCD usage.

The R10 version of the AD system uses SYS1.IODF01 as the supplied IODF. You cannot modify a production IODF. You must create a “work” IODF (based on a production IODF), modify the work IODF, and then create a new production IODF. When doing this, you will need to specify a data set name for the work IODF and the new production IODF. The following convention is often used when dealing with IODF names:

- ▶ Begin with a working IODF, such as SYS1.IODF01.
- ▶ When altering this IODF, create a working IODF (as outlined in the steps later in this section) by appending the name WORK to the original name. For example, you would create SYS1.IODF01.WORK when modifying SYS1.IODF01.
- ▶ When you create a production IODF from your modified IODF (again, the steps are outlined later in this section), you would create the next name in numeric order. In our example, this would be SYS1.IODF02.
- ▶ Any IODF name ending with WORK is not a production IODF.

Using this plan, the progression of names is:

```

    SYS1.IODF01 --> SYS1.IODF01.WORK --> SYS1.IODF02 -->
    (production)                               (production)

--> SYS1.IODF02.WORK --> SYS1.IODF03 --> etc
                               (production)

```

With this convention, the SYS1.IODFnn with the highest numeric suffix is the most current production IODF. The SYS1.IODFnn.WORK where nn is one less than the highest suffix is the working IODF used to create the current production IODF.

(Not everyone uses this plan. Another common alternative is to have SYS1.IODFnn.WORK used to build SYS1.IODFnn. If an alteration is needed, then SYS1.IODFnn is used to create SYS1.IODFnn+1.WORK, which is then used to build SYS1.IODFnn+1. The plan you does not matter, as long as you remember your plan.)

If you have a SCSI-attached 3480 tape drive which has the IDRC feature and you are unable to read or write 3480 tapes to process compacted or compressed data (IDRC), one probable cause is that the I/O definition for your MVS system does not specify that the COMPACT feature is enabled for the tape unit at the address you are using (typically address 560). This has been a common problem, and we use it as the basis of the following example.

To alter your I/O definition file (assuming SYS1.IODF01 is the current production IOD), use the following steps:

1. Under ISPF, select option M and then option 4 (using the AD system ISPF menus). You could also use ISPF option 6 and enter CBDCHCD on the command line. This will invoke the HCD facility.
2. Ensure that the IODF file name is set to 'SYS1.IODF01' (or whatever is the name of your current production IODF).
3. Select option 1 (Define, Modify, etc.).
4. Select option 5 (I/O devices).
5. On the next panel, page down to device address 560, place a slash (/) in the prefix area and press ENTER. Device 560 is a 3480 tape drive; the goal here is to add a feature to the definition of this drive.
6. Select option 2 (Change).
7. HCD will now ask you to assign a name for an IODF work data set. If you started with SYS1.IODF01, we suggest you assign the name SYS1.IODF01.WORK. (You will need to specify it within single quotes.)
8. The next panel gives you the option to specify or revise several fields for device 560. Just press ENTER to continue.
9. On the next panel, place a slash (/) in the prefix area to select the CBIPO OS and press ENTER.
10. Select option 1 (Connect, Change)
11. On the next panel (Define Device Parameters / Features), change the setting for the COMPACT feature from NO to YES. (You may need to scroll forward to find the COMPACT feature line.)
12. On the next panel (Define Device Parameters/Features), note that a new generic will be shown (3480X). This is what you should specify in your JCL (UNIT=3480X) to specifically select a 3480 drive which has IDRC enabled. You could also set the option for the other generics shown for this device (CART, CTAPE, TAPE) to NO. Press ENTER twice to activate.
13. Press PF3 until you return to the main HCD menu.
14. Verify that the I/O definition file shown (bottom of screen) is SYS1.IODF01.WORK. Select option 2 (Activate).
15. Select option 1 (Build new IODF). Verify the work IODF name is 'SYS1.IODF01.WORK'. Set the new production name to 'SYS1.IODF02'. Set the volume serial to OS39M1. Select option 1 to use the work IODF in use at present. Press Enter.
16. At the next panel (Specify or Revise), just press Enter. Success is near when you see the message: Production IODF SYS1.IODF02 created.
17. Press Enter.

18. Press PF3 until you exit HCD and return to the main ISPF menu. Now you must change the MVS parameter controls so that the IPL process will pick up the new IODF file. Complete the following steps:
19. From the main ISPF menu select option 2 (Edit).
20. Set the data set name to 'SYS1.IPLPARM'. (This example assumes you are using SYS1.IPLPARM for your LOADxx members.) This example assumes you normally use LOAD00 to start your system. You should modify the following instructions to reflect whatever LOAD parameter you normally use.)
21. On the command line enter S LOAD00.
22. On the command line of the new (empty) edit member, enter COPY LOADT1. (The name LOADT1 is arbitrary, but it must be in the in the format LOADxx.)
23. Change the IODF parameter in column 10 from 01 to 02.
24. Save the new LOADT1 member, exit completely and shutdown MVS.
25. Now IPL with the new T1 parameter. The typical complete IPL parameter might be 0A82T1 if your OS30M1 volume is at address 0A82.

When you IPL OS/390, the new IODF file will be activated and you can use UNIT=3480X to direct tape mount requests to device 560 which now has the COMPACT feature enabled. Verify that it is working correctly. Then edit and change all the LOADxx members you use to reflect the new IODF number. (You might keep one old member, under a new name, as a fallback if something goes wrong.)

There are many ways to do this IO change (almost as many as there are MVS system programmers). One advantage of this approach is that if you have trouble with your new IODF file, you can change your IPL load parameter back to 0122WS and revert to your previous IODF file.

You can modify a current production IODF, or delete old IODFs. However, we *strongly* recommend that you never modify your only production IODF. Keep at least one older production IODF, and keep at least one LOADxx member that points to it. This provides a fallback position if your modified IODF does not work.

5.82 How to set up an ISPF administrative environment

As a system administrator, you must have access to products such as RACF, SMP/E, ISMF and others that normal users do not need to access routinely. Normal TSO users do not need access to these functions, and you can make the logon process for normal TSO users much faster by deleting unnecessary allocations, as described in “ISPFLITE procedure” on page 45 and “ISPFLITE procedure” on page 75.

If you remove the allocations for RACF, SMP/E, and so forth, from the normal logon procedure, you must devise a way to make these allocations available to yourself and other administrators. There are two general approaches:

- ▶ Use a different TSO logon procedure (such as the supplied ISPFPROC) when you want a large number of preallocated functions, or
- ▶ Use the normal (smaller, faster) TSO logon procedure, and execute a CLIST to perform the additional allocations, when needed.

The use of separate logon procedures is probably the most common approach, although we think the CLIST approach can be more convenient. Remember that, by default, the AD system provides a large logon procedure (with the related allocations for RACF, SMP/E, and so forth included) for everyone. The discussion in this section is relevant only if you decide to provide a smaller (but faster) logon procedure for your users.

As mentioned, using different logon procedures is the most common approach. This involves:

- ▶ Creating a new procedure. This is usually done by copying an existing procedure and adding or deleting DD statements as required. The AD system keeps TSO logon procedures in SYS1.TSOPROC, but they can be in any procedure library in the JES2 procedure concatenation.
- ▶ Defining any new logon procedure to RACF:

```
RDEFINE TSOPROC newproc UACC(READ)    or

RDEFINE TSOPROC newproc UACC(NONE)
PERMIT newproc CLASS(TSOAUTH) ID(myuser)

SETROPTS REFRESH RACLIST(TSOPROC)
```

If the new procedure has UACC(READ) then all TSO users can use it. If it has UACC(NONE) then appropriate users must be given access to it through a PERMIT (PE) command. The SETROPTS command, as shown, should be issued (from TSO) after the TSOPROC is changed, otherwise the change may not be recognized until the next IPL.

- ▶ Changing the logon procedure for a session, using a field in the LOGON panel of TSO. The changed value will be remembered for future logons.

The disadvantage of using several logon procedures is that a user must logoff and logon again if he needs to use another procedure.

The CLIST approach has fewer logon procedures, and might have only one. It uses a CLIST (or a selection of CLISTS) to add allocations to whatever was allocated by the logon procedure. You would execute a CLIST instead of logging off and logging on again with a different logon procedure. In this case you should use the same basic logon procedure for all users in your system and then define a number of CLISTS to allocate additional libraries to perform administrator tasks.

In general, you would place the CLISTS where all your users can access them. This should not be a security exposure, because security should be based on other things than a CLIST full of ALLOC statements. (Any TSO user can issue ALLOC statements from his terminal.) The AD system provides SYS1.CLIST as a convenient place for common clists.

An example CLIST to allocate additional libraries is shown below. This CLIST simply allocates all the data sets that are normally allocated in the full logon procedure (which is ISPFPROC in the AD system). It does not matter that some of the data sets may already be allocated by the user's logon procedure. (The assumption is that this CLIST is used only during a small percentage of user sessions, so a minor bit of inefficiency does not matter.)

You can create this CLIST in SYS1.CLIST(ADMINDS), or in any other CLIST library you choose.

```
PROC 0 VOL(0S39M1)
CONTROL MSG NOFLUSH
PROFILE NOMODE MSGID PROMPT INTERCOM WTPMSG
/* ----- */
/* Alloc clist for all functions 0S/390 R3 AD */
/* ----- */
FREE F(SYSUADS)
FREE F(SYSLBC)
FREE F(SYSPROC)
FREE F(SYSEXEC)
FREE F(SYSHELP)
FREE F(ISPLLIB)
```

```

FREE F(ISPLIB)
FREE F(ISPTLIB)
FREE F(ISPEXEC)
FREE F(ISPPLIB)
FREE F(DITPLIB)
FREE F(ISPSLIB)
FREE F(SDSFMENU)
FREE F(ISPTABL)
FREE F(SMPTABL)
ALLOC F(SYSUADS) DA('SYS1.UADS') SHR
ALLOC F(SYSLBC) DA('SYS1.BROADCAST') SHR
ALLOC F(SYSPROC) DA('ISP.SISPCLIB',
                   'SYS1.CLIST',
                   'SYS1.DGTCLIB',
                   'SYS1.HRFCLST',
                   'SYS1.SICECLIB',
                   'ICQ.ICQCLIB',
                   'SYS1.SBLSCLIO',
                   'SYS1.SBPXEXEC',
                   'SYS1.SCBDCLST',
                   'SYS1.SEDGEXE1',
                   'GIM.SGIMCLSO',
                   'SYS1.SERBCLS',
                   'MQM.V1R2MO.SCSQCLST',
                   'CSQ.SCSQCLST',
                   'EOY.SEOYCLIB',
                   'IOE.SIOEEXEC',
                   'ASU.SASUEXEC',
                   'FAN130.SVSC.PROCLIB',
                   'IEL111.SVSC.PROCLIB',
                   'IGY210.SVSC.PROCLIB',
                   'CBC.SCBCPRC') SHR
ALLOC F(SYSEXEC) DA('ISP.SISPEXEC',
                   'SOMMVS.SGOSREXX',
                   'SOMMVS.SGOSEXEC',
                   'EUV.SEUVEXEC',
                   'SYS1.SBPXEXEC',
                   'CSQ.SCSQEXEC',
                   'IOE.SIOEEXEC',
                   'ASU.SASUEXEC',
                   'FAN130.SVSC.CMDLIB',
                   'IGY210.SIGYCLST',
                   'EOY.SEOYCLIB') SHR
ALLOC F(SYSHELP) DA('SYS1.HELP',
                   'ISP.SISPHELP',
                   'SYS1.SEDGHLP1',
                   'TCPIP.SEZAHLP',
                   'FAN130.SFANHENU',
                   'FAN130.SEAGHENU') SHR
ALLOC F(ISPLLIB) DA('GDDM.SADMMOD') SHR
ALLOC F(ISPLIB) DA('ISP.SISPMENU',
                   'SYS1.DFQMLIB',
                   'SYS1.DGTMLIB',
                   'ICQ.ICQMLIB',
                   'SYS1.SICEMENU',
                   'SYS1.HRFMSG',
                   'SYS1.SBPXMENU',
                   'SYS1.SCBDMENU',
                   'SYS1.SBLSMSGO',
                   'CSQ.SCSQMSGO')

```

```

        'SYS1.SISTMSG0', -
        'SYS1.SEDGMENU', -
        'TCPIP.SEZAMENU', -
        'GIM.SGIMMENU', -
        'ISF.SISFMLIB', -
        'SYS1.SERBMENU', -
        'CBC.SCBCIMG', -
        'SOMMVS.SGOSMSG0', -
        'EOY.SEOYMENU', -
        'ASU.SASUMENU', -
        'FAN130.SFANMSEU') SHR
ALLOC F(ISPEXEC) DA('ISP.SISPEXEC', -
        'SYS1.SBPXEXEC', -
        'CSQ.SCSQEXEC', -
        'EUV.SEUVEXEC') SHR
ALLOC F(ISPPLIB) DA('SYS1.LOVAL.ISPPNLS', -
        'ISP.SISPPENU', -
        'SYS1.DFQPLIB', -
        'SYS1.DGTPLIB', -
        'ICQ.ICQPLIB', -
        'SYS1.SICEPENU', -
        'SYS1.HRFPANL', -
        'SYS1.SBLSPNLO', -
        'SYS1.SBPXPENU', -
        'SYS1.SCBDPENU', -
        'CSQ.SCSQPNLE', -
        'SYS1.SEDGPENU', -
        'EUV.SEUVPNL', -
        'TCPIP.SEZANPNL', -
        'TCPIP.SEZAPENU', -
        'GIM.SGIMPENU', -
        'ISF.SISFPLIB', -
        'SYS1.SISTPNLO', -
        'SYS1.SISTPNL1', -
        'SYS1.SERBPENU', -
        'CBC.SCBCPNL', -
        'SOMMVS.SGOSPMLS', -
        'EOY.SEOYPENU', -
        'ASU.SASUPENU', -
        'FAN130.SFANPENU') SHR
ALLOC F(DITPLIB) DA('DIT.V1R3MO.SDITPLIB') SHR
ALLOC F(ISPSLIB) DA('ISP.SISPSLIB', -
        'GIM.SGIMSENU', -
        'ISP.SISPSENU', -
        'ISF.SISFSLIB', -
        'SYS1.DGTSLIB', -
        'SYS1.HRFSKEL', -
        'SYS1.SICESLIB', -
        'SYS1.SBLSKELO', -
        'CSQ.SCSQSKL', -
        'CBC.SCBCSKL', -
        'SOMMVS.SGOSSKEL') SHR
ALLOC F(ISPTLIB) DA('ISP.SISPTEU', -
        'SYS1.DGTTLIB', -
        'SYS1.ICQTLIB', -
        'ICQ.ICQTABLS', -
        'SYS1.SICETLIB', -
        'ASU.SASUTLIB', -
        'SYS1.SBLSTBLO', -
        'SYS1.SBPXTENU', -

```

```

        'SYS1.SCBDTENU',           -
        'GIM.SGIMTENU',           -
        'CSQ.SCSQTBLE',           -
        'ISF.SISFTLIB',           -
        'SYS1.SERBT',             -
        'SYS1.SERBTENU',          -
        'EOY.SEOYTENU',           -
        'SYS1.SMP.OTABLES') SHR
    ALLOC F(SDSFMENU) DA('ISF.SISFPLIB') SHR
    ALLOC F(ISPTABL) DA('SYS1.SMP.OTABLES') SHR
    ALLOC F(SMPTABL) DA('SYS1.SMP.OTABLES') SHR

```

The above CLIST is executed by entering this TSO command:

```
EXEC 'SYS1.CLIST(ADMINDS)'
```

assuming the CLIST was stored as SYS1.CLIST(ADMINDS). (This CLIST is not provided by the AD system. You need to make your own.)

It is fairly easy, although a bit tedious, to make the above list. You would start with the distributed SYS1.TSOPROC(ISPFPROC) procedure and transform it to the CLIST. Do not reallocate the temporary data sets that are usually allocated by whatever shortened logon procedure you normally use. When making this CLIST, be especially careful with the continuation marks. The logon procedures distributed with the AD system contain the volsers of all the libraries; this speeds processing because there is no need to go to the catalog for every DD statement. The above CLIST does not include volsers, so it will be slower than the equivalent logon procedure. This should not matter, if this CLIST is used only occasionally by an administrator.

You might want to make another CLIST to FREE the data sets allocated by the allocation CLIST, but this is seldom necessary.

5.83 How to interpret system data set names

TSO logon procedures and CLISTs contain many library names whose use is not obvious from their names. These libraries are (almost) all involved with ISPF usage of various functions, such as RACF, ISPF, and so forth. The following table may be helpful.

ISP.SYS.....	ISPF libraries
ISF.SIS.....	SDSF functions
....DGT.....	ISMF functions
EOY.SEOY....	BookManager functions
....HRF.....	ISPF libraries for RACF functions
...SICE....	Sort functions
ICQ.ICQ.....	Libraries for TSO/E extended functions
....SBLS....	ISPF libraries for IPCS functions
SYS1.SBPX...	OMVS functions
SYS1.SCBD...	HCD functions
....SEDG....	DFSMS RMM (removable media)
GIM.GIM.....	SMP/E and CBIPO functions
....SERB....	RMF functions
CBC.SCBC....	C and C++ functions
ASU.SASU....	DCE functions
DFS.SIOE....	DCE/DFS functions
SOMMVS.SGOS....	ISPF libraries for SOM functions
IEL.SIEL....	PL/1 functions
TCP/IP.....	TCP/IP functions
....SEZA....	ISPF libraries for TCP/IP functions

....DFQ.....	ISPF libraries for DFSMS HSM functions
SYS1.CLIST	Nonstandard; provided with AD system

Other data set prefixes that might be useful:

AJV.....	Java
ASMA.....	High Level Assembler
ASMT.....	High Level Assembler Toolkit
ABJ.....	CCCA
ANF.....	Print Server
AOP.....	Print Server
API.....	NetSpool
APK.....	(for AFP printing)
ASU.....	DCE
BFS.....	Lan Server
CBC.....	C, C++ compiler
CSF.....	Cryptographic APIs and tools
CEE.....	Language Environment (LE) data sets
CMN312.....	Net.Commerce
CMX.....	(DB2 related)
COT.....	Payment Server (eBusiness)
CSF.....	ICSF (crypto, smart cards, ...)
CSQ.....	MQ Series
DIT.....	DITTO
EDC.....	C/370 compiler
EMW.....	NetQuestion
EOY.....	BookManager
EQAW.....	Cooperative Development
EUV.....	DCE
EWX.....	LANRES
EZM.....	NetView
FAN.....	REXX
FFST.....	FFST (Service Tool)
GIM.....	SMP/E
GLD.....	LDAP
GSK.....	System SSL
HLA.....	High Level Assembler
HOU.....	Millennium conversion tool
HPJ.....	Java compiler
ICA.....	Firewall
ICE.....	OS/390 Sort
ICQ.....	TSO/E extended functions
IEL.....	PL/1 Compiler
IGY.....	Cobol Compiler
IEL.....	PL/1
IGY.....	COBOL for MVS & VM
IMO.....	Text Search product
IMW.....	Web Server related
IOA.....	OSA/SF
IOE.....	DFS (DCE)
ISP.....	ISPF libraries
MQM.....	MQ Series
SOMMVS.....	System Object libraries
SYS1.SIST...	VTAM materials

You may notice there are two general formats to system data set names. For example:

```
DCE.SEUVEXEC
DCE.V1R1M0.SEUVEXEC
```

MVS traditionally used the longer form, with a product's version number, release number, and modification level placed in the second data set name qualifier. This had advantages and disadvantages; over time the disadvantages (need to change too much JCL for every new release) overcame the advantages. Most MVS data sets appear to be moving to the shorter name that does not contain the release level. You will find both formats in current OS/390 systems, including the AD system.

5.84 How to backup and restore OS/390

There are a number of aspects to consider for disk backup planning:

- ▶ Disk hardware errors. P/390, MP3000, and Netfinity/EFS systems normally use RAID adapters, making the unrecoverable loss of a disk unlikely. However, good planning should consider the possibility of the complete loss of a disk or RAID array.
- ▶ Logical errors. A change to a system library (a PARMLIB, for example) might make a system unusable, or an application error may corrupt a database beyond recovery. There are two levels of concern here:
 - ▶ You cannot IPL the system, and must use external (standalone, OS/2, UNIX) functions to recover the system to where it can be IPLed.
 - ▶ You can IPL and perform your recovery under OS/390.
- ▶ You must be realistic about the resources needed to take backups. It is not reasonable, for example, to backup every volume every night -- especially using a 4mm tape drive that is standard on many of these systems.
- ▶ Do you have sufficient disk space to keep two OS/390 systems? One of these might be a minimal system that is used only for recovery purposes. The P/390 AD CD-ROMs do not provide such a system. Having a second system makes recovery of a non-IPLable system much easier.

You have more backup options with a P/390 and Netfinity/EFS system than with a "real" mainframe, because you can use OS/390 backup facilities or OS/2, AIX(R/390), or UNIX (EFS) facilities. Considering OS/2 functions, you can:

- ▶ Simply copy the OS/2 file that is an OS/390 emulated volume to another file. For example:

```
OS2 C:> copy D:OS39M1_1.A82 E:OS390M1_1.BAK
OS2 C:> copy D:OS39M1_2.A82 E:OS390M1_2.BAK
```

This will take several minutes, because the files are large, but it is very simple to do. You can recover the OS390M1 volume by copying it back to the original location, or by altering the P/390 configurator to point to the new name. If you have sufficient OS/2 disk space, this is a very easy way to take volume-level backups. The same concept applies to EFS systems, using the underlying UNIX facilities.

- ▶ You can use a compression product to do the same thing; for example, use one of the PKZIP products or UNIX/Linux equivalents. The advantage is that this will take much less disk space, although compression will take longer than a simple copy. You can expect a 2 - 4 compression of a typical OS/390 volume.
- ▶ You can use an OS/2 tape backup product; the most common is the informal XTAPE program that is now supplied with P/390 systems.

The disadvantage of these backups is that they are always volume-level backups. That is, you must restore a complete OS/390 volume. You cannot restore only a single data set, because OS/2 or UNIX programs are unaware of the internal format of an emulated 3380/3390 volume.

You can use the normal OS/390 dump/restore program, ADRDSSU, which is documented in *DFSMSdss Storage Administration Reference*, order number SC26-4929. Your backup device can be the 4mm tape or any other tape drive you have connected to your system, such as SCSI-attached 3490-type drives. The advantage of backups under OS/390, using ADRDSSU, is that you can restore a single data set if needed. A disadvantage is that ADRDSSU is a bit more complex to use than XTAPE, for example.

To restore a single data set using ADRDSSU, you must have a workable OS/390 system. If you cannot IPL your OS/390, you can perform a complete volume restore of an appropriate ADRDSSU dump, using the standalone restore program that is included with the AD CD-ROMs. At this level, an ADRDSSU backup has no advantages over a backup taken using an OS/2 or UNIX product.

If you use ADRDSSU, you can plan the full range of typical OS/390 backups:

- ▶ Full or Complete
All data in the system is backed up regardless of type, content or access pattern.
- ▶ Incremental
An incremental backup only backs up data that has been changed since a previous backup. Managing data this way requires the operating system to maintain some form of change control for individual files or data sets.
- ▶ Selective
Data is backed up under control of the user. Specific volumes, files or data sets are selected for backup based in its importance or access pattern.

Such planning, especially a well-designed incremental backup scheme, is typical of a production environment. It may not be as typical of a smaller development environment, because it requires order and discipline to make it work.

More extensive backup methods, such as HSM (DFSMSHsm) and other advanced OS/390 backup products, can be used, but these are likely to be overkill for an AD environment.

A major consideration when deciding on backup/restore strategy is the hardware available. The following devices are good candidates:

- ▶ All systems incorporate a 3.5 inch diskette drive.
- ▶ All systems incorporate a SCSI attached 4 mm DAT drive.
- ▶ One or more SCSI attached tape drives. These may or may not be OS/390 compatible (3420/3480/3490 compatible and supported by the P/390 or R/390 drivers).
- ▶ With a 370 Channel Adapter/A card you may have 'real' IBM or compatible 3420, 3480 or 3490 tape or cartridge drives.
- ▶ Other system bus or SCSI attached devices (Zip drives for example).

While a diskette drive is not normally considered as an OS/390 backup device, we have found it useful for downloading (IND\$FILE) PARMLIB and JCL members. If you choose to backup to 4mm DAT, you should be wary of the life of the tapes you are using, particularly if you are backing a large amount of data to each tape. This technology is somewhat hard on the media, with the result that the tapes have a limited *reliable* usage. (Comments by users in the electronic forums suggest 50 uses or 50 hours use. This is not a lot of uses if you are backing up a large amount of disk, for example to backup and compare 10 gigabytes of data will take approximately 10 hours.) Also, the drives need to be cleaned (using the cleaning cartridge) frequently; two passes of the cleaning cartridge are sometimes needed. In general, 4mm backups have not been as dependable as, for example, 3480/3490 backups.

The authors use a mixture of backup techniques, with OS/2 disk copies being the most common, followed by XTAPE backups written to 4mm, and followed by ADRDSSU backups written to a SCSI-attached 3490E drive. This is not an ideal arrangement, but it meets the needs of a diverse number of users.

The OS/2 backups (disk copy, XTAPE) are considerably faster than any OS/390 backups. For OS/390 backups, the 4mm drive (used to emulate a 3480) is slower than either SCSI-attached or channel attached 3480/3490 drives. There is not much performance difference between channel-attached and SCSI-attached 3480-3490 drives.

5.85 How to continue TSO command lines

You can continue a TSO command line by using either plus or minus signs. The location of the + or - sign is important, and location of the first character in the following (continuation) line can be significant.

A minus sign means to concatenate the following line, starting at the beginning of the line, with the previous line. A plus sign means to concatenate the following line, starting at the first non-blank character, with the previous line. Using the LISTDS command as an example:

```
+----- first column of terminal input
|
V

LISTDS 'OGDEN.LIB.ASM' MEMBERS      normal command

LISTDS 'OGDEN.+
LIB.ASM' MEMBERS                    works. Blanks before LIB skipped

LISTDS 'OGDEN.-
LIB.ASM' MEMBERS                    fails due to two blanks before LIB

LISTDS 'OG-
DEN.LIB.ASM' MEMBERS                works because continuation line starts in col 1

LISTDS 'OG -
DEN.LIB.ASM' MEMBERS                fails due to space after OG

LISTDS 'OGDEN.LIB.ASM'-
MEMBERS                              works. Blanks required before MEMBERS
```

5.86 How to copy an ADRDSSU dump

The ADRDSSU program, the standard OS/390 program used to back up (*dump*) a disk volume, produces tapes that are difficult to copy. By default, it writes tape blocks longer than 32K bytes, and such blocks cannot be processed by common utility programs. (The advantage in doing this is that more data will fit on a tape and the average transfer rate of a dump or restore job is higher.) You can use a special function of ADRDSSU to copy such tapes:

```
//OGDEN5 JOB 1,OGDEN,MSGCLASS=X
//      EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//TAPEIN  DD UNIT=570,VOL=SER=TAPE01,LABEL=(1,SL),DISP=SHR,
//          DSN=DUMP.HFS001
//TAPEOUT DD UNIT=560,VOL=SER=TAPE02,LABEL=(1,SL),DISP=(NEW,KEEP),
```

```
//          DSN=DUMP.HFS001
//SYSIN    DD  *
COPYDUMP  INDD(TAPEIN)  OUTDD(TAPEOUT)
/*
```

You must tailor the JCL to your situation, of course, but the basic job structure should be clear. We used the specific example shown here to copy a backup on a 3480 cartridge to a 4mm tape on a P/390 system.

5.87 How to set your TSO prefix

From a TSO prompt (either a READY prompt, or ISPF option 6), enter the command:

```
PROFILE
```

This will display a few lines of text with your current TSO profile settings. You can probably ignore most of the settings if you are using ISPF for your normal TSO interface, but the PREFIX setting can be important.

TSO (including ISPF) will normally use the PREFIX value to automatically create a HLQ for data sets that you name. For example, if my PREFIX is OGDEN, then the TSO commands:

```
LISTDS SYS1.PARMLIB      will try to use OGDEN.SYS1.PARMLIB
LISTDS 'SYS1.PARMLIB'   will try to use SYS1.PARMLIB
```

That is, TSO will automatically use your prefix as the high-level qualifier (HLQ) of any data set that you name unless you place the data set name in quotes. This use of quotes (*single quotes*) to avoid prefixing is very common. You might receive instructions with a software product, for example, to *enter the quoted name of library X* in a certain field.

The TSO prefix value is normally your userid, but you can change this using the PROFILE command.

```
PROFILE NOPREFIX        disables use of prefix
PROFILE PREFIX(joe)
PROFILE PREFIX(sys1)
```

A system administrator sometimes finds NOPREFIX to be useful. Normal TSO users almost always want their PREFIX equal to their userid, and this is the default value when a new user is defined.

Changing your PREFIX does not create any new security authorities. It simply provides a more convenient syntax for many commands, for normal TSO users.

5.88 How to escape from a command or program

You may need to escape from a command if, for example, it prompts for a parameter and you do not know the correct format. 3270 terminals use the PA1 key (Programmed Action 1) as an *escape*, *break*, or *attention* key. The normal TSO terminology is to *attention out* of a command or program. In a simple program structure, PA1 can interrupt a running program (in a loop, perhaps), but the effect in a complex multitasking program structure can be more complicated.

The full function of PA1 is seldom used. Under certain circumstances, you can reenter an interrupted program, or call the TSO TEST function to work with the interrupted program.

3270 terminals have a PA1 key on the keyboard. Emulated 3270 keyboards, usually PC keyboards, do not have a key marked PA1. Various emulators and customizations may make the emulated PA1 key difficult to find. The most common location, on a PC keyboard, is Alt-Insert.

3270 terminals also have a PA2 key (often located at Alt-Home on emulated keyboards), but this key has no well-defined function for TSO.

5.89 How to use VIO

VIO (Virtual I/O) is a mechanism for using virtual memory instead of real disks for temporary data sets. For example, compiler work files would normally be allocated to VIO. In general, VIO is faster than using disks. The VIO process emulates a real CKD disk, with track length, cylinder size, and so forth, the same as a real CKD disk. A normal program is not aware of whether it is using VIO or disks for temporary data sets. There is no programming difference, and no easy way to sense the difference.

For P/390 OS/390, VIO is very important. In proportion to its processor speed, and to its I/O bandwidth, a P/390 OS/390 system has an unusually large amount of main storage. (This comment assumes you have 256MB of P/390 storage, and that none of it is allocated to S/390 expanded memory.) Also, in proportion to its processor speed and the amount of main storage, P/390 OS/390 has rather limited CKD disk performance, due to the need to emulate CKD/ECKD functionality. The availability of main storage should be used whenever possible to replace CKD disk I/O. VIO does this.

When an OS/390 system is defined, a number of esoteric³⁵ device names can be made eligible for VIO processing. In the OS/390 CD-ROM systems, names SYSDA, DASD, TSO, and VIO are all eligible for VIO processing. If all the following conditions are met, OS/390 will use VIO for a data set:

- ▶ If a DSNAME is coded (in JCL) it must indicate a temporary name, such as &&tempname. Typically, a DSNAME is not given, unless it must be PASSED to a later step of the job.
- ▶ If the DISP parameter is coded it must be (NEW,DELETE), (NEW,PASS), or (,PASS).
- ▶ The UNIT parameter must specify a VIO-eligible name.
- ▶ A volser must not be coded.
- ▶ The data set organization cannot be IS, PDSE, or VSAM.

If any of these conditions are not met, OS/390 will assign the temporary data set to a real (or emulated) disk. The allocation log, printed at the beginning of most batch jobs, will indicate whether VIO was used.

5.90 How to add a program to OS/390

There is no single way to *add a program* to the system. You can execute a program from any load library, including your private TSO-related library, by using a JOBLIB statement (JCL for batch) or a CALL statement (TSO). If you want to add a program so that you do not need to use JOBLIB/STEPLIB or CALL functions, you need to add it to a data set in the LNKLST or to a data set in LPA.

³⁵ Do not let this word confuse you. These are arbitrary names assigned to hardware devices; these names can be used in JCL. For example, SYSDA is an esoteric name. Esoteric names can be locally assigned, using an HCD option. The other ways to name devices are by address (A80, for example) or by device type (3390, for example).

You normally place a load module (the executable form of a program) in a load library by either of these methods:

- ▶ Link edit the module, with the SYSLMOD DD statement in the link edit step pointing to the target library and specifying the member name to be used:
- ▶ `//LKED.SYSLMOD DD DISP=OLD,DSN=SYS1.P390.LINKLIB(myname)`
- ▶ Copy the module using ISPF 3.3 or a batch job using IEBCOPY.

In the AD system, all programs in the LNKLST and LPA are potentially authorized programs. (They must be link edited with AC=1 and explicitly use authorized functions if they are to take advantage of this potential.) Programs in LPA must be reentrant; programs in LNKLST need not be reentrant. If you are not certain whether your program is reentrant, assume it is not.

For LNKLST, you can link edit or copy your program into any library in the LNKLST. All the data sets in LNKLST are defined in SYS1.PARMLIB(LNKLSTxx) or in SYS1.PARMLIB(PROGxx). The AD system uses PROGxx. In the AD system, a library named SYS1.P390.LINKLIB is included, and is a good place to link small applications you may want to *add to the system*.

A program added to an LPA library is not available (in LPA) until after an IPL that specifies CLPA (Create Link Pack Area). The cold start IPLs provided with the AD system include a CLPA, although they also cold start JES2 (which you might not want). A program added to a LNKLST library is not available (via LNKLST) until the operator issues an F LLA,REFRESH command.

Selecting a name (that is, the member name in the library that will hold your program) can be a problem when *adding your program to the system*. There are a large number of programs already in the system. You must not duplicate any existing name. The system has a standard search path it uses to find programs. It stops searching as soon as it finds a matching program (member) name. If you select a name that duplicates a system name, there are two possibilities:

- ▶ If the system finds your program first (in its search order), the system module with the same name is, effectively, deleted from the system.
- ▶ If the system finds the system program first, then your program will never be found using the standard system search mechanism.

For these reasons, almost all applications (as opposed to system programs) are placed in private libraries and accessed through JOBLIB/STEPLIB functions or explicit TSO CALL commands.

5.91 How to start a JCL library

The JCL Reference manual describes many JCL options and statements. Most jobs, however, can be run using a very small subset of these control statements. Once you are familiar with the characteristics of the jobs you typically run, you may find that you need to know the details of only a few of the control options.

Within a job, the control statements are grouped into job steps. A job step consists of all the control statements needed to run one program. If a job needs to run more than one program, the job would contain a different job step for each of those programs.

Every job must contain a minimum of the following two types of control statements:

- ▶ A JOB statement, to mark the beginning of a job and assign a name to the job. The JOB statement is also used to provide certain administrative information, including security,

accounting, and identification information. Every job has one and only one JOB statement (usually called a job *card*).

- ▶ An EXEC (execute) statement, to mark the beginning of a job step, to assign a name to the step, and to identify the program or procedure to be executed in the step. You can add various parameters to the EXEC statement to customize the way the program executes. Every job has at least one EXEC statement. A job can consist of up to 255 job steps, including all steps in any procedures that the job calls.

In addition to the JOB and EXEC statements, most jobs usually also contain:

One or more DD (data definition) statements, to identify and describe the input and output data to be used in the step.

Before creating JCL for a job, you need to know:

- ▶ Installation conventions. Some installations require accounting information in a JOB card, for example. (The AD CD-ROM has no requirements.) You may need to ask someone more familiar with your installation to help you identify the conventions used at your installation.
- ▶ How to allocate and edit a data set. We are assuming you know the process to allocate and edit a data set to save, modify and reuse jobs as required.
- ▶ The job to be done and the resources needed. You need to determine what work you plan to have OS/390 perform:
 - What programs and data sets you will use.
 - What DD names are used by your application.
 - Where the output, if any, should go when the job completes. You will either dispose of the output or hold it for later printing or viewing.
- ▶ How to view and understand held output. Running your job will produce three types of held output:
 - System messages (JES and MVS)
 - Your JCL code with procedures expanded, overrides applied, and symbols resolved.
 - Output from the application.

Almost all users should have at least one TSO data set for JCL. By convention, these data set names usually contain JCL or CNTL as a qualifier. For example, you can use ISPF option 2 to allocate a library (PDS) with:

```
userid.JOBS.JCL on USER01 volume
15 tracks primary space
5 tracks secondary space
20 directory blocks
FB record format
80 logical record size
8000 block size
```

You can create the first member in this data set by editing (ISPF 2) `userid.JOBS.JCL(MYJOB)`. Enter JCL statements as required. Remember:

- ▶ Every JCL statement begins with `//`.
- ▶ A continued statement ends with a comma.
- ▶ Everything is uppercase (with a few exceptions).
- ▶ The continuation statement begins with `//`.

We suggest you copy examples from a known text. The Utilities manual (*DFSMS/MVS Utilities*, SC26-4926 or later) is a good starting place.

When you have finished entering the JCL into the data set, submit the job by entering the SUBMIT command from the ISPF EDIT command line, the ISPF command shell, or following a READY mode message. See “Submitting a job while editing the data set” on page 216. Or, at a TSO READY prompt, you could enter:

```
SUBMIT JOBS.JCL(MYJOB)
```

When entering the command from the ISPF COMMAND SHELL or after a READY message, you must surround the data set name with single quotation marks if the HLQ (high-level qualifier, or first part of the data set name) is not the same as your PREFIX. See 5.87, “How to set your TSO prefix” on page 212 for a discussion of your PREFIX.

```
EDIT ---- userid.ADMIN.JCL----- LINE 00000000 COL 001 080
COMMAND ==> SUBMIT                                SCROLL ==> CSR
***** TOP OF DATA *****
//OGDEN JOB 1,OGDEN,MSGCLASS=X
// PGM=IEBGENER
//SYSPRINT DD SYSOUT=X
//SYSUT1 DD DSN=SYS1.P390.CNTL(USSTAB)
//SYSUT2 DD SYSOUT=X
/*
```

Figure 5-31 Submitting a job while editing the data set

5.92 How to use ALLOC and FREE

The ALLOC command is the TSO equivalent of a JCL DD statement. You can associate a data set (DSname) with the DDname coded in a program. This could be an existing data set, or a new data set that is created by the ALLOC statement—just as a new data set can be created by a JCL DD statement. You can specify DCB parameters, and many (but not quite all) of the parameters that can be specified through JCL.

Due to OS/390's automatic locking mechanisms, a data set allocated to a user may not be available for use by other users. This is controlled by the OLD and SHR parameters of the allocation. The FREE command unallocates a data set, making it available for other users. The batch equivalent of FREE is the end of a job step.

If you create a new data set with the ALLOC command,³⁶ you can specify parameters for DCB attributes, record format and so forth. If you are allocating many data sets, this can become tedious; the ATTRIB command can be used to define common attributes used for a number of allocations. An alternative is the LIKE parameter, to reference the attributes of an existing data set. For initial usage of ALLOC, you should not need either the ATTRIB, LIKE, or USING (which specifies the name of an ATTRIB function) parameters.

ALLOC and FREE commands can be entered from a READY prompt, from ISPF option 6, within a CLIST, or with practically any other general TSO interface.

You can allocate an existing cataloged data set by the TSO command:

```
ALLOCATE DA('JOE.MY.DATA') OLD F(DDA1)
ALLOC DA(MY.FILE) F(SYSUT1) SHR
```

³⁶ In practice, few users create new data sets with ALLOC or use the ATTRIB command. Instead, they use ISPF functions or batch jobs to create data sets, and ALLOC to later refer to those data sets.

This allocates JOE.MY.DATA to DDname DDA1. We assume the user plans to CALL a program that requires a DDA1 DD allocation. Note the single quotes around the name. If the quotes are not used, TSO will append your PREFIX (usually your userid) as the first qualifier. If your PREFIX/userid is JOE, then the second allocation above will use the data set JOE.MY.FILE. You need to use the quotes when the HLQ of the data set does not match your userid. See 5.87, "How to set your TSO prefix" on page 212 for more information about your PREFIX.

You can allocate an existing uncataloged data set with the command:

```
ALLOCATE DA('PROD1.LIB.LOAD') FILE(SYSLIB) VOLUME(USRVOL)
```

You can allocate a new data set with different space allocation and different DCB of an existing model data set by:

```
ALLOCATE DA('ID.NEW1.DATA') SPACE(10,5) TRACKS LRECL(80) -
BLKSIZE(8000) RECFM(F,B) VOLUME(WORK01) NEW
```

Note the use of a continuation character when using two input lines. Also note the subtle differences in syntax between ALLOC and JCL, as seen in the RECFM parameter, for example. The JCL equivalent would be RECFM=FB. You can use the command HELP ALLOC to obtain a list of the proper syntax for the command.

You can specify data set attributes for non-VSAM data sets that you intend to allocate in several ways:

- ▶ Using the LIKE operand to obtain attributes for an existing data set:

```
ALLOCATE DA('ID.COPIED.DATA') LIKE('USERID.MODEL.DATA')
ALLOCATE DA('ID.NEW2.DATA') SPACE(10,5) TRACKS LIKE('ID.MODEL.DATA')
```

- ▶ By identify the data set and describe its attributes in explicit way in the ALLOC command.
- ▶ Using the ATTRIB command to build a list of attributes, during your TSO session you can have the system making reference to this list of data set attributes by the USING operand.
- ▶ With Storage Management Subsystem installed and active you can specify a DATACLAS operand.
- ▶ For HFS files, you can specify the following operands:

```
PATH, PATHDISP, PATHMODE, PATHOPTS, DSNTYPE(HFS) and DSNTYPE(PIPE)
```

When you enter the LOGOFF command, all data sets allocated and any attribute lists created during the session are released by the system.

UNALLOC is the alias name for the FREE command.

To concatenate more than one data set you can follow this example:

```
ALLOC FI(SYSLIB) DA(A.CLIST,B.CLIST,C.CLIST) SHR REUSE
or
ALLOC FI(SYSLIB) DA(A.CLIST, -
                   B.CLIST, -
                   C.CLIST) SHR REUSE
```

If you want to add another data set to the concatenation, there are two ways:

- ▶ Use the FREE command to unallocate the data sets in the concatenation. Then use the ALLOCATE command to allocate the entire number of data sets.
- ▶ Specify the REUSE operand when you use the ALLOCATE command.

An HFS allocation (for use with MVS programs, of course, since no ALLOC functions are needed within OpenEdition shells) might be:

```

ALLOC PATH('/u/ogden/hello.c')      +
    PATHDISP(KEEP,KEEP)             +
    PATHOPTS(OWRONLY,OCREAT)        +
    PATHMODE(SIRWXU)                 +
    FILE(OUTPUT)

```

You can free (unallocate) a data set by specifying the DSname or the DDname.

```

FREE DATASET('ID.MY.LOAD')          DSname
FREE DATASET(MY1.LOAD,MY2.LOAD,MY3.LOAD) DSnames
FREE FI(SYSUT1,SYSUT2,SYSUT3)       DDnames

```

You can free Hierarchical File System (HFS) allocation with:

```

FREE PATH('/U/USERID/OUTPUT.DBP')  +
    PATHDISP(DELETE)

```

You can free of all dynamically allocated files, attribute lists, and data sets:

```
FREE ALL
```

5.93 How to execute a CLIST or REXX script

There are two main methods of executing a CLIST or REXX in the TSOE environment.

- ▶ You can explicitly execute a CLIST or REXX by using the TSOE EXEC command. The easiest way to do this is from ISPF option 6 the ISPF Command Shell. Assuming you have written a CLIST or REXX called CLISTXYZ and it is stored as member CLISTXYZ in data set P390.TSO.CLIST

```
COMMAND ==> exec 'p390.tso.clist(clistxyz)'
```

Alternately a CLIST/REXX can be executed from any command line (====>) in ISPF by prefixing the command with the string TSO as illustrated below using the same CLIST example as shown in Figure 5-32 on page 219.

```
COMMAND ==> tso exec 'p390.tso.clist(clistxyz)'
```

A CLIST/REXX can also be executed in native TSO (READY mode) using the EXEC statement. The TSO prefix is not required in native TSO:

```
exec 'p390.tso.clist(clistxyz)'
```

- ▶ If you intend to use the CLIST/REXX often or want to make it available for use by other users it can be copied to a data set in the SYSPROC DD concatenation of LOGON procedures. A data set named SYS1.CLIST has been provided in the AD system for this purpose. This method has the advantage that the EXEC command is not required because TSO searches the SYSPROC DD data sets when looking for a command or program. By putting your CLIST/REXX in a SYSPROC data set it becomes a command for TSO sessions using the SYSPROC data set. So assuming you copied your CLISTXYZ to SYS1.CLIST you can execute it directly from ISPF option 6 as illustrated in Figure 5-33 on page 219.

Alternately direct from an ISPF command line as:

```
COMMAND ==> tso clistxyz
```

or in native TSO (READY mode):

```
clistxyz
```



```

Menu List Mode Functions Utilities Help
-----
                          ISPF Command Shell
Enter TSO or Workstation commands below:

===> exec 'p390.tso.clist(clistxyz)'

Place cursor on choice and press enter to Retrieve command

=> ishell
=> exec 'p390.tso.clist(allocall)'
=> lista st h
=>
=>

```

Figure 5-32 ISPF Command Shell panel with CLIST/REXX execution

```

Menu List Mode Functions Utilities Help
-----
                          ISPF Command Shell
Enter TSO or Workstation commands below:

===> clistxyz

Place cursor on choice and press enter to Retrieve command

=> exec 'p390.tso.clist(clistxyz)'
=> ishell
=> exec 'p390.tso.clist(allocall)'
=> lista st h
=>
=>

```

Figure 5-33 ISPF Command Shell panel with direct CLIST/REXX execution

5.94 How to transfer data between OS/2 and OS/390

There are three common methods of transferring data between OS/2 and OS/390:

If you are using the native display attached to your P/390, then using the OS/2 Clipboard is by far the quickest and easiest way to transfer small amounts of data between the two environments. This method is limited by the amount of data in a 3270 emulation window. (We assume you are using a 3270 emulator that supports the clipboard cut and paste functions.) To use the clipboard simply use the edit subcommands **copy** and **paste** between the environments. This is supported by the PCOM 3270 emulator sessions, the OS/2 System and Enhanced Editors and should work for any OS/2 application that supports the standard OS/2 clipboard function. PCOM performs EBCDIC - ASCII conversion automatically.

The PC File Transfer program (IND\$FILE) is distributed with the AD system and fully supported by many 3270 emulator programs. The following may be useful when using this facility:

- ▶ Be certain that your TSO session is either in TSO READY mode before starting a transfer. Any other status will probably cause a failure. In principle, ISPF option 6 should also work, but we have had trouble with this in various emulated I/O environments, especially when copying from a PC to OS/390.
- ▶ You can enter **SEND** and **RECEIVE** commands in an OS/2 window or a window in Microsoft operating systems, if the **SEND** and **RECEIVE** commands are installed. These commands are often distributed as part of a 3270 emulator product.

```
SEND C:SOURCE.TXT B: 'OGDEN.LIB.ASM(PROG1)' CRLF ASCII
RECEIVE C:TARGET.ASM B: 'OGDEN.PROG1.ASM' CRLF ASCII
```

The PC file is listed first. The B: operand indicates that 3270 emulator session B is the target window in this example. You would not need quotes around the OS/390 file name if you depend on your current TSO prefix value. Using the quotes tends to prevent errors.

- ▶ Some emulators (such as PCOM) provide pulldown menus and panels to perform **SEND** and **RECEIVE** functions. (Some users find the line commands easier to use than the pull-down functions.)
- ▶ In the transfer type definitions, the transfer type should be TEXT for normal text/JCL/documentation. This ensures that EBCDIC/ASCII translation is performed and line/record formatting is preserved between the environments (CRLF).
- ▶ The CRLF operand, in the examples above, tells IND\$FILE to convert between PC records (delimited by carriage return and line feed characters) and OS/390 records.
- ▶ The ASCII operand indicates that ASCII/EBCDIC conversion should be performed.

Another method of transferring data is via ftp, assuming OS/390 TCP/IP is operational.

5.95 How to select BLKSIZE for load libraries

Use BLKSIZE=32760 for all new load libraries.

The linkage editor (and binder) take this as the maximum blocksize, but use it in an intelligent way. This blocksize does not result in large amounts of wasted space at the end of a track. The linkage editor will adjust actual block sizes written to fit in the remaining space on a track.

For existing load libraries, continue to use the BLKSIZEs that are already defined. Smaller block sizes³⁷ may cause slightly slower system operation when loading programs. There are several factors involved in this timing: lost revolutions (does not apply to emulated CKD devices), dynamic chaining of channel programs, PCI processing, and so forth. The performance differences are usually not important enough to have users make the effort to reblock their load libraries (or reblock system load libraries).

If you want to reblock a load library, IEBCOPY provides the COPYMOD function for PDS load libraries (but not for PDSE). However, there are a few special cases where this will not work. We suggest you do not attempt to reblock system load libraries unless you have a very specific need to do so.

5.96 What are BPXAS jobs

The MVS console frequently shows BPXAS being started. What is this?

³⁷ Many older load libraries have BLKSIZE=6144. This size was tuned for 2314 disk drives, many years ago, and has remained common ever since--mostly because few users understand how the linkage editor handles block sizes.

These are address spaces used for *fork* processing in UNIX System Services. They are created by WLM (WorkLoad Manager). WLM attempts to keep a pool of created, but idle BPXAS address spaces to speed *fork* processing. BPXAS address spaces that have been idle for 30 minutes are automatically terminated. You can stop the creation of these address spaces with the MVS command:

```
F BPXINIT,SHUTDOWN=FORKINIT
```

although you would normally use this command only as part of a system shutdown sequence.

In general, you can ignore the BPXAS STARTING messages that appear on the MVS console during normal operation.

5.97 How to list system variables

The MVS operator command D SYMBOLS is the most basic way to display system symbols.

The following REXX clist provides an example of another way to list simple system variables. You can enter this code into a sequential data set or PDS member, and then execute it with an EXEC command.

```
/* REXX */
SAY 'LOGON USERID: SYSUID = ' SYSVAR('SYSUID')
SAY 'TSO LEVEL: SYSTSOE = ' SYSVAR('SYSTSOE')
SAY 'LOGON PROCEDURE: SYSPROC = ' SYSVAR('SYSPROC')
SAY 'CPU TIME USED (SECONDS): SYSCPU = ' SYSVAR('SYSCPU')
SAY 'SYSNODE = ' SYSVAR('SYSNODE')
SAY 'JES ID: SYSJES = ' SYSVAR('SYSJES')
SAY 'RACF LEVEL: SYSLRACF = ' SYSVAR('SYSLRACF')
SAY 'VTAM TERMINAL ID: SYSTEMID = ' SYSVAR('SYSTEMID')
X = SYSCPUS('CPUS.')
SAY 'CPU SERIAL = ' CPUS.1
/* SOME OTHER POTENTIAL SYSTEM VARIABLES: */
/* SYSDNAME SYSDMVS SYSDOPSYS SYSDFP SYSDMS SYSHSM SYSPLEX */
/* SYSSMFID SYSAPPCLU SYSSECLAB SYSCLONE SYSSYMDEF */
RETURN
```

For example, the above could be entered into OGDEN.LIB.CNTL(VARS) and executed with the TSO command EXEC 'OGDEN.LIB.CNTL(VARS)'.

These system variable (and others) can be used to construct symbolic parameters that are used in PARMLIB members, PROCLIB members, and used for other purposes. There are more OS/390 system variables than those used here; we do not know of any document that lists all the system-provided variables.

5.98 How to write to stdout from the assembler

UNIX System Services Programming: Assembler Callable Services Reference, SC28-1899, a very large manual, describes how to call many UNIX-type services from assembly programs. In this case:

```
CALL BPX1WRT(1,ADDR(buf),0,len,rv,rc,rsn)
```

where the first operand specifies the file handle. By convention, stdout is file handle 1. All parameters are full words and the caller must be in AMODE31. The third parameter, shown as 0, can specify an ALET for cross-address space use. The rv, rc, and rsn parameters are the return value (number of bytes written), the return code, and the reason code.

If you plan to use UNIX functions from assembler, you will need the manual referenced above.

5.99 How to find your TSO logon ID

The following C program will display your TSO logon id.³⁸ The program can be run from batch, TSO, or OMVS.

```
//OGDEN7A JOB 1,OGDEN,MSGCLASS=X,MSGLEVEL=(0,0)
/* COMPILE AND EXECUTE C PGM
//          EXEC EDCLLG,PARM.COMPILE='SOURCE'
//COMPILE.SYSIN DD *,DLM='<>'
#include <stdio.h>
int main()
{
int flccvt, ascb, asxb, acee, i;
char aceeusri>9@;
i = 0x224;                               /* PSAAOLD          */
memcpy( (void *)&ascb, (void *)i, 4);    /* get a(ASCB)     */
i = ascb + 0x6c;                          /* ASCBASXB        */
memcpy( (void *)&asxb, (void *)i, 4);    /* get a(ASXB)     */
i = asxb + 0xc8;                          /* ASXBSENV        */
memcpy( (void *)&acee, (void *)i, 4);    /* get a(ACEE)     */
i = acee + 0x15;                          /* ACEEUSRI        */
memcpy( (void *)&aceeusri, (void *)i, 8); /* get userid      */
aceeusri>8@ = 0;
printf("My userid is %s \n" ,aceeusri);
}
<>
```

To compile the program under OMVS, remove the JCL (including the delimiter line at the end), copy to an HFS with name chain.c, and use `cc -o chain chain.c`.

This is a simple example of following control block chains in MVS. It starts with a pointer to the current address space control block, which is stored at a constant address in low storage. This technique may not work for complex situations, such as multiuser servers and so forth.

As shown, the program is not very useful. It is intended only as a simple example of control block chaining and usage, using C.

5.100 How to expand a VTOC size

When you initialize a DASD volume for OS/390 use, you must specify a VTOC size. Many examples use 14 tracks. (The first track of the first cylinder of a disk is used for the volume label (and for an IPL program, if present). The remaining 14 tracks of the first cylinder are often used for the VTOC. This may not be ideal for performance, but it is commonly done.) This may not be sufficient in some cases, and you may experience Sx37 ABENDS if the VTOC is full.

It is fairly unusual to take preventative action for VTOC sizes. That is, the system administrator usually initializes volumes with a selected VTOC size and takes no additional actions unless an ABEND occurs.

³⁸ There are other ways to determine your logon id, of course. However, you should not depend on the TSO PREFIX value as one of these.

You can use ICKDSF to expand a VTOC, although there are rare cases where this may not work. The DSF job might be:

```
//OGDENJ3 JOB 1,OGDEN,MSGCLASS=X
// EXEC PGM=ICKDSF
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  REFORMAT UNITADDRESS(A91) VERIFY(WORK02) EXTVTOC(44)
/*
```

Using this form, the volume must be OFFLINE before you submit this job. In this example, volume WORK02 is verified to be at address A91 before any other actions are taken. The VTOC is then expanded to 44 tracks. Expanding from 14 tracks to 44 tracks (which uses an even three cylinders, if it starts on cylinder 0 track 1) is a reasonable expansion factor for many environments.

More information can be found in *Device Support Facilities: User's Guide and Reference*, GC35-0033.

5.101 How to repair system with TSO down

There are various reasons why you might be unable to use TSO. These reasons are usually related to systems programming activities; for example, a newly-installed exit might prevent TSO logons due to a logic error. You need to submit and run a job to fix the error, but are unable to log onto TSO to submit the job.

One way to deal with this problem is to submit the job from a card reader, since this does not require TSO. The P/390 systems, MP3000, and EFS systems can emulate a card reader. Input for the reader comes from OS/2 (or UNIX). You can build a job, with the appropriate JCL, using an OS/2 editor and copy it to an OS/2 directory that is defined in the P/390 or emulated I/O configurator as part of the definition for a 2540 device. This is described elsewhere; you can read the AWS2540.DOC file for more information. The equivalent functions for EFS follow the same pattern.

In order to use this function, you need to have a card reader defined to JES2. The current AD systems do not have a card reader defined in JES2. (In our example, you cannot change the JES2PARAM definitions in PARMLIB, because you cannot log onto TSO to do it.) You should define a card reader to JES2 *before* you have a problem.

If you have not defined a card reader to JES2, you can dynamically define one when JES2 is starting, provided you can reply to the JES2 startup message. At JES2 startup time, you normally reply WARM or NOREQ to the initial JES2 message. Instead, you can reply CONSOLE. If you do this, JES2 will read its normal JES2PARAM definitions and then write this message on the MVS console:

```
nn $HASP469 REPLY PARAMETER STATEMENT, CANCEL, OR END
```

You can define a reader by entering:

```
R nn,'RDR(1) UNIT=00C'
```

JES2 will then reply:

```
$HASP603 RDR1 UNIT=00C,AUTH=(DEVICE=NO,JOB=NO,SYSTEM=NO),
$HASP603 CLASS=A,HOLD=NO,MSGCLASS=A,PRIOINC=0,
$HASP603 PRIOLIM=15,PRTDEST=LOCAL,PUNDEST=LOCAL,
$HASP603 START=YES,TRACE=NO,XEQDEST=LOCAL
nn $HASP469 REPLY PARAMETER STATEMENT, CANCEL, OR END
```

You would reply END. You may need to issue the command \$\$ to start JES2 processing. You may need a \$SRDR1 command to start the reader. After JES2 is finished initializing, you can use OS2 to copy your job to the correct OS/2 directory and JES2 should read and process it.

Job output will return to JES2. Since, in our example, you cannot use TSO you may need to start a JES2 printer to see the output. You could use the same method to define a printer. (The current AD systems do not have a printer defined.) See the AWS2821.DOC file for information about emulated 1403 printers. One setup option will direct printer output to an OS/2 file.

Note that you need to communicate with JES2 when it is initializing in order to make the dynamic device definitions described here. The current AD systems have an *automatic IPL* that does not give you a chance to respond to JES2 startup messages. In this case, you will need to stop JES2 (as cleanly as possible --- see the SHUTDOWN script in PARMLIB in the AD systems for examples) and restart it with a S JES2 command.

If your problem was due to a logic error in an exit, your correction job might look something like this:

```
//KMEISERA JOB 1,KURT,MSGCLASS=A,USER=KMEISER,PASSWORD=XXYYZZ
//S EXEC ASMACL
//C.SYSIN DD *
IKJEFLD1 CSECT
        SLR  15,15
        BR   14
        END
/*
//L.SYSLMOD DD DISP=SHR,DSN=SYS1.LINKLIB(IKJEFLD1)
```

In the case we describe here, you would create this job with an OS/2 editor. You need the USER and PASSWORD options in the JOB statement if a valid RACF userid is needed to properly run the job. In this case, we are altering LINKLIB and this is RACF protected; we need a userid that has ALTER authority for LINKLIB.

In this example, a module is replaced with a new module that does nothing except set a zero return code. Correcting your problem will require something different, of course.

5.102 How to understand condition codes

You can include condition codes on EXEC statements. These check the completion codes (return codes) of previous job steps. Examples are:

```
//STEP3 EXEC PGM=MYPGM3,COND=(4,GE)
//STEPX EXEC PGM=MYPGM4,COND=(4,LT)
//STEPZ EXEC PGM=XYZ,COND=((8,EQ,STEP3),(16,LT,STEP4))
```

These can be difficult to understand because they use a negative logic and use operands in the reverse order that most people would find intuitive. If the condition stated in the COND

The first example can be read as *bypass this step if 4 is greater than or equal to any earlier completion code (in this job)*. That is, bypass this step if all earlier steps had completion codes in the range 0 - 3. (This step might be a cleanup program that is run only if earlier steps failed.)

The second example reads *bypass this step if 4 is less than the completion code of any previous step*. That is, if any earlier step had a completion code of 4 or greater, skip this step. (This step might be a final program that is run only when all earlier programs were successful.)

The third example has two tests. If either (or both) are true, the step is bypassed. If the completion code from STEP3 was 8, or if the completion code from STEP4 was 16 or greater, then this step will be skipped.

In these examples, if an earlier step ABENDED, then all following steps are bypassed. However, you can check for ABENDS:

```
//STEPK EXEC PGM=ABC,COND=ONLY
```

```
//STEPJ EXEC PGM=ABC,COND=EVEN
```

These two tests use positive logic and refer to ABENDs in previous steps. STEPK will be executed only if an earlier step ABENDED and STEPJ will be executed even if an earlier step ABENDED.

See *OS/390 MVS JCL Reference (GC28-1757)* for more complex examples. While not required to use COND functions, there is a general assumption that:

```
completion code 0 means the program was successful
completion code 4 means there were minor errors or warnings
completion code 8 means there were errors
completion code 16 means there were severe errors
```

At the system level, whatever value is in register 15 when a jobstep program ends is taken as the condition code. All high-level languages have functions to set the return code. In this context, return code and completion code are synonyms.

5.103 How to use LLA (Library Lookaside)

VLF (Virtual Lookaside Facility) is an OS/390 facility that stores objects and retrieves copies when requested. In a sense it is a server. LLA is a facility that improves module fetch time by storing the module pointer in VLF. Before MVS fetches a module from disk, it asks LLA (which, in turn asks VLF) if it has a copy of the module.

VLF keeps its objects in separate address spaces; these use expanded storage (if available) or central storage and are pageable. LLA is a client of the VLF service, and deals only with load modules. Other VLF clients deal with RACF profiles, and so forth.

Member COFVLFxx in PARMLIB contains a list of all the users of VLF. LLA is one such user. Member CSVLLAxx in PARMLIB contains a list of all libraries (and/or library members) that are managed by LLA. If this member does not exist, then all LNKLST libraries are managed by default (assuming that COFVLFxx contains a statement for the CSVLLA class, to enable LLA).

The MVS operator command D LLA will display all the libraries currently managed by LLA. The list it produces is rather long, and is best used with the SDSF log display. If you change a library (add or alter a member) that is managed by LLA, you should issue the MVS operator command F LLA,REFRESH to update the LLA information. Until you do this, the system (via LLA) will continue to use the old module.

VLF and LLA are perhaps not as well documented as a new system user would like, but more information can be found in the *OS/390 Initialization and Tuning Guide*. You can use LLA to make your programs load faster by adding your load module library to the LLA list. This makes sense if your program is (a) fetched hundreds of times each day, (b) is not in LPA, and (c) is not in LNKLST.

5.104 How to place SCEERUN and old COBOL libraries

You should not have both LE (meaning the SCEERUN library) and another COBOL library in LNKLST at the same time because they have some duplicate module names. Whichever is listed first will be used and the other is unreachable (via the LNKLST). Likewise, you cannot have both in LPA at the same time, because there will be duplicate module names.

You will need to use STEPLIBs or JOBLIBs for one or the other. The R10 AD CD-ROM has SCEERUN in the LNKLST.

5.105 How to control SuperZAP

At one time, the superzap program (currently named AMASPZAP) was regarded as a severe security exposure. Current versions obey all RACF rules and are not exposures if data sets are properly protected by RACF and if the operator does not automatically **reply U** to requests he does not understand.

An easy way to provide even more control of superzap is to:

- ▶ Copy the AC=1 version (the distributed version) to a private, authorized library that is protected with UACC=NONE, and
- ▶ Then relink the public version with AC=0.

With this arrangement, any user can superzap his own data sets, or anyone else's unprotected data sets, but he cannot work on VTOCs and so forth.

AMASPZAP is currently found in SYS1.MIGLIB, which is automatically included in any LNKLST.

5.106 How to use ISRDDN

ISPF offers a little-known function known as ISRDDN. Fundamentally, it consists of a panel that lists all the data sets currently allocated to your TSO session, along with the DDname used for the allocation. Its method of handling concatenated partitioned data sets is what makes ISRDDN so useful. You can go to ISPF option 6 and enter the command ISRDDN to start the function.

When looking for a specific member in concatenated partitioned data sets (such as the PARMLIB or PROCLIB concatenations), the system will search the first data set in the concatenation, then search the second, and so forth. As soon as it finds the specified member, the search stops. The same member name may exist in several of the concatenated PDSs; the member that is found is the earliest one in the concatenation. This makes the order of concatenation very important.

For example, the AD systems recently have two PARMLIBs. These are SYS1.ADCD10.PARMLIB (or some similar name) and SYS1.PARMLIB and they are concatenated in that order. If member IEASYSCS exists in both PARMLIBs, it is the one in SYS1.ADCD10.PARMLIB that is used, because SYS1.ADCD10.PARMLIB is first in the concatenation.

If you want to update member VATLSTAA in PARMLIB, you must first look in SYS1.ADCD10.PARMLIB to see if it is there; if it is not there, you then look in SYS1.PARMLIB. (If it is in both PARMLIBs, you would update the SYS1.ADCD10.PARMLIB member because that is the member that the system will use. If it is not in any library in the concatenation, you can add it to any of the libraries.) This manual searching of concatenated PDSs, in the order they are used by the system (or by an application program in other cases), to find the *first occurrence* of a specific member name, is cumbersome

The ISRDDN function can make browsing or editing of concatenated PDSs much easier, because it can list all the members of the multiple PDSs *using the first member found if the same member name exists in several of the PDSs*.

ISRDDN lists only data sets that are allocated to your TSO session. Its use is not limited to working with PARMLIBs, but we will use PARMLIBs for illustration. The system PARMLIBs are not normally allocated to your TSO session. You can use the ALLOC command to do the allocation; it is easier to make a small CLIST to do the job. For the AD CD-ROM system, edit SYS1.CLIST and create member PARMEDIT as follows:

```
FREE DD(PARMLIBS)
ALLOC DD(PARMLIBS) DA('SYS1.ADCD06.PARMLIB' 'SYS1.PARMLIB') SHR
ISRDDN
FREE DD(PARMLIBS)
```

Then, from ISPF option 6, execute PARMLIBS (or whatever member name you used for the CLIST). (You could place the CLIST in your own PDS and execute it with something like EXEC 'MY.LIB.CNTL(XX)'; you will need the fully-qualified name in parenthesis unless your LLQ is CLIST.)

The ISRDDN display is something like this:

```
Volume Disposition Act DDname Data Set Name.....
SCPMV5 SHR,KEEP >_ CIDTABL SYS1.CIDTABL
MOD,DEL >_ DSQDEBUG-----Subsystem File-----
*VIO NEW,DEL >_ DSQEDIT SYS98365.T113211.RA000.P390.R0100039
OS39R6 SHR,KEEP >_ PARMLIBS SYS1.ADCD06.PARMLIB
SCPMV5 SHR,KEEP >_ SYS1.PARMLIB
SCPMV5 SHR,KEEP >_ SMPTABL SYS1.SMP.OTABLES
etc
```

In this example note the following:

- ▶ The Subsystem File entry means a SYSOUT data set.
- ▶ VIO means that no volume is involved for this temporary data set.
- ▶ Find the DDname PARMLIBS. Notice that the next line has a blank DDname. This indicates concatenation.
- ▶ On your system, there may be several screens of this listing. You can scroll through the screens using PF8 and PF7.
- ▶ The Act column is where you can enter an action command.

The action commands are single letters, as follows:

```
B or S or / Browse the data set
E Edit
```

V	View
F	Free the allocation
C or Z	Compress a PDS
I	Display information about the data set
Q	List current users of data set

If you browse or edit the *first* data set in a concatenation, then the whole concatenation is used. Assuming a concatenation of PDSs, the member list appears as:

Name	Lib	etc.....
PBXPRM00	1	etc....
PBXPRM01	2	etc....
PBXPRM97	1	etc...

The second column is critical; it appears only if you browse/edit the *first* data set in a concatenation (that is, use the line that contains the DDname). The Lib column indicates which data set in the concatenation contained this member. In this example, BPXPRM00 was in the first PDS in the concatenation, BPXPRM01 was in the second PDS in the concatenation, and so forth. *If the second PDS also contained a BPXPRM00 member, it is not shown in the member list.* If you edit one of the members in this member list, you are editing the effective member *provided the PDSs are always concatenated in the order shown.* This is the case for PARMLIBs in the AD system. It may not be true for other systems or other uses of concatenated PDSs.

The ISRDDN function has a number of other powerful functions, especially for resolving allocation or ENQ contention. We suggest you read the HELP material. (Press PF1 when in the ISRDDN function.)

There is one important limitation to ISRDDN. ISRDDN will consolidate the member list of a limited number of concatenated PDSs. In earlier releases, the number was four. In release 10, the number appears to be 16. If there are more than this number of PDSs in a concatenation, the additional data sets will be listed on the primary ISRDDN panel, but the member lists (for browsing or editing) will contain only members from the first 16 PDSs.

5.107 How to edit an alias

A member name in a PDS can point to another member; it can be an *alias* for another member. That is, the alias name in the PDS directory points to an existing member (which also has a *base name*). You can access the member equally by using the alias name or the base name. There can be more than one alias for a member, although this is fairly rare. Why are aliases used? There are two common reasons. One is to avoid changing hard-coded module names (in other modules) when several modules are combined. (We suspect this is why there are so many aliases in base MVS load libraries.) Another reason is to provide more intuitive names for procedures that have less intuitive base names. (These base names are probably due to a naming convention associated with the product.)

Alias names are frequently used within base OS/390 load libraries. They are also used in IBM-provided procedure libraries. For example, if you look at CBC.SCBCPRC, using ISPF, you will see several members beginning with EDCC that are aliases for other members. There is no information given to indicate which is the base name for an alias.

Unfortunately, the alias status does not survive editing by ISPF (or by any other common editor). Consider the following situation in MY.LIB.CNTL, a common PDS:

Member	MAINNAME	points to	TTR	101	
Alias	OTHERNME	points to	TTR	101	(same place)

If you edit OTHERNME, the editor will fetch the member starting at TTR³⁹ 101. You can change the data and SAVE the member. When it is SAVEd, it will be written at a new TTR in the library and will no longer be an alias. It will be a *base member*. MAINNAME no longer has an alias name. Both names still work, but they point to different data.

Consider the same starting situation, with MAINNAME and OTHERNME (an alias) pointing to the same TTR. Now edit and change MAINNAME. When you SAVE it, it will be written at a new TTR. The MAINNAME directory entry now points to the new TTR, but the OTHERNME alias entry still points to the original TTR. We now have an alias (OTHERNME) without a base name, and a new base (MAINNAME) without an associated base. Both names still work, but they point to different data.⁴⁰

If a maintenance program, such as SMP/E, *knows* about the original ALIAS situation--and assumes it still exists--it might apply future maintenance incorrectly because it may depend on the alias names. There is no straightforward way in ISPF or TSO to create or assign alias names to source/text libraries. You can use the IEBUPDTE utility to work with alias names. The TSO RENAME command also has an option for aliases.

When you copy a PDS, the alias relationships may or may not be maintained, depending on how you perform the copy. If you use IEBCOPY directly, and copy a whole PDS, then the aliases are copied correctly. If you selectively copy members with IEBCOPY, the alias relationship will be lost unless you copy both base and alias names. If you copy a PDS with ISPF 3.3, the alias relationships will be correctly kept ONLY if you copy both base and alias names in the same invocation of 3.3 (that is, while within the member selection panels).

Load module libraries may also contain aliases. Aliases for load modules are assigned by linkage editor control statements. (If an alias name matches a CSECT name or ENTRY name in the module, then calling the module via the alias name causes the module to receive control at an alternate entry point. Thus a single module may be called by multiple names (aliases) and receive control at various entry points in the module.) If you relink a module that has aliases, you MUST include the original alias control statements; if you do not, you may have execution failures. SMP/E keeps track of alias statements for load modules installed through it.

Should you use aliases for your programs or products? We rather strongly suggest the answer is NO, unless you have a specific need for them.

5.108 OS/390 Unix System Services terminology

The terms *OMVS*, *OE*, *OpenEdition*, *USS*, *Unix System Services*, and *OS/390 UNIX System Services* are often used interchangeably. (This redbook does this.) This usage is not quite correct, but it is so widespread that almost everyone involved accepts it.

In more exact terms, *OMVS* is an OS/390 subsystem and also the name of a shell. *OE* and *OpenEdition* were mostly marketing terms. *OS/390 UNIX System Services* is the correct name for the UNIX interfaces and services of OS/390. *USS* is not an approved abbreviation for UNIX System Services.

³⁹ A TTR is a way to point to an offset (track number and record number within the track) inside a PDS. A directory entry in a PDS contains a TTR to the associated member.

⁴⁰ This situation continues to exist after the PDS is compressed.

5.109 How to clear a 3270 screen in a C program

Traditional C programming techniques, especially those used to control line-mode terminal output, do not always fit perfectly in a 3270 environment. Here are two ways to clear a 3270 screen when running in an OMVS environment:

```
    print ("\f");  
or  
    system("tput clear");
```

The second method is obviously not portable.

5.110 How to set and display extended attributes

Extended attributes are set with the `extattr` command. They can be displayed with the `-E` option of the `ls` command. For example,

```
> ls -alE myfile.exe  
  
-rwxr-xr-x -ps 1 STC TSO 1234 Dec 27 myfile.exe  
---
```

In this example, the “p” means that the file is considered program controlled, and the “s” means that it can run in a shared address space. Missing is the “a” flag that would mean the program in the file is APF authorized. (In addition to this flag, the program must have been linked with `AC=1` in order to be APF authorized.) The “s” is set, by default, for all executable files. (This means that execution can be in a shared address space if the `_BPX_SHAREAS` shell variable is set to `YES`.) You would need to use the `extattr` command to reset the “s” bit if you want to prevent an executable from sharing an address space. For example:

```
extattr -s /u/ogden/myexec
```

The `extattr` command syntax is:

```
extattr [+aps] [-aps] filename
```

You can change the “s” bit for files you own, or for any file if you have `UID=0`. You can change the “a” and “p” bits only if you have at least *read* access to the RACF profiles `BPX.FILEATTR.APF` and `PBX.FILEATTR.PROGCTL` profiles (in the `FACILITY` class.) A single profile, `BPX.FILEATTR.*`, could be used to control permission to change both bits.

If you copy or move a file, the extended attributes are set to their default state (`-a -p s`). If you back up a file (with `tar` or another similar program) the extended attributes are preserved. This can create a security exposure. If you restore a file, or otherwise access it and issue a `MOUNT` command to connect it to your OpenEdition file system, the extended attributes (and the `setuid/setgid` bits) are used as they are found in the `MOUNTed` file unless you use the `NOSETUID` option of the `MOUNT` command. The `NOSETUID` option will cause the `setuid`, `setgid`, `apf`, and `progctl` attributes to be ignored. An example of a `MOUNT` command is:

```
MOUNT FILESYSTEM('HFS.MVS.NAME') MOUNTPOINT('/u/xyz') -  
TYPE(HFS) NOSETUID
```

This is a TSO command, but the user issuing it must have an OMVS segment with `UID=0`.

Earlier OS/390 releases (before V2R5) required authorized programs or members of program controlled libraries to reside in MVS data sets. Current releases allow both these attributes (authorized, program controlled) to be specified for HFS files. These attributes are set in an HFS file with the **extattr** command. However, a user must have specific RACF authority to issue the **extattr** command. The necessary RACF authority can be specified as follows. (You must have RACF SPECIAL authority to issue these commands.)

```
RLIST FACILITY BPX.FILEATTR.APF AU           (to verify profile exists)
RLIST FACILITY BPX.FILEATTR.PROGCTL AU      (to verify profile exists)
```

```
if they do not exist:
RDEF FACILITY BPX.FILEATTR.APF UACC(NONE)
RDEF FACILITY BPX.FILEATTR.PROGCTL UACC(NONE)
```

```
give userid=goodguy permission to use them
PERMIT BPX.FILEATTR.APF CLASS(FACILITY) ID(goodguy) ACC(READ)
PERMIT BPX.FILEATTR.PROGCTL CLASS(FACILITY) ID(goodguy) ACC(READ)
SETROPTS RECLIST(FACILITY) REFRESH
```

In recent AD systems both profiles are defined and userids P390 and IBMUSER have been permitted to these two profiles. Being permitted to BPX.FILEATTR.APF allows a user to manipulate the *authorized program* bit with the **extattr** command. Being permitted to BPX.FILEATTR.PROGCTL allows a user to manipulate the *program control* bit with the **extattr** command.

Once RACF authorization has been established, then the user can issue a command such as:

```
extattr +ap /u/ogden/myspecialpgm
```

The ability to establish authorized and/or program controlled modules in HFS largely removes the need to place OE executables in MVS load libraries. However, appropriate programs (highly used, reentrant) may still be placed in LPA. The ability to mark HFS files as authorized can easily compromise the integrity of MVS if such files are not rigorously controlled. In particular, MOUNTED file systems should be (a) completely trusted, or (b) mounted as *nosetuid*.

When a file is *copied* (**cp** command), the authorized and program controlled bits are removed. This is always done, automatically, regardless of the authority of the user doing the moves or copies. If the file is *moved* (**mv** command), the bits are not changed.

Note that if the sticky bit is on, the program attributes (authorized, program controlled) are taken from the MVS module that is found, not from the OE extended attribute bits.

5.111 How to assign a default printer for the lp command

The **lp** command is normally used for printing HFS files that are in a readily printable format. For example,

```
> lp mydata.lst           < will go to default printer & class
> lp -d,x mydata.lst     < will go to JES SYSOUT=X
> lp -drm27,a mydata.lst < will go to JES destination rm27, class A
```

You can set the default printer and class by:

```
export LPDEST=,X,          (where X is a JES2 class)
```

or by adding this command to `/etc/profile` or something similar. This works if you are not using the OS/390 Print Server.

With the z/OS R1.1 AD system, the first example placed output in the JES2 Output queue, class A, from an STC with jobname P3904. (The logon id was P390.) The second example placed output in the JES2 Hold queue, class X, from an STC with jobname P3908.

If you are using the OS/390 Print Server you must create a printer definition that sends data to JES2 class X (or whatever class you want), and then use that definition name in the export LPDEST statement.

The OS/390 Print Server is installed, but not activated on current AD systems. In this case, you can use the method shown first to define a default JES2 class for the `lp` command.

5.112 How to execute an authorized program

To exist as an authorized program in a library, a module must be:

- ▶ Link edited with the `AC=1` parameter.
- ▶ Placed in an authorized library (as defined in `PARMLIB`) or placed in an HFS file with the “a” extended attribute set.

To execute as an authorized program, the program must be loaded into an address space that has never had an unauthorized program loaded in it.

Excluding special situations, this means that the first program executed in an address space (known as the *job step program* in traditional MVS environments) needs to be authorized. It can then transfer (`LINK`, `XCTL`) to another authorized program, and so on. The first non-authorized program that is used breaks the chain and the address space is `ABENDED`.

For OpenEdition, this means that an authorized program must be used via *fork* and cannot be used via a *local spawn*. A local spawn, into the caller's existing address space, would result in the program executing unauthorized, because the caller's shell (and TSO environment, if entered via `OMVS` in a shared address space) is non-authorized. If the extended attribute “s” is not set, the `_BPX_SHAREAS` variable (used to indicate that child processes can be run in the parent's address space) is ignored and a new address space is created. See “How to set and display extended attributes” on page 230 for a discussion of extended attributes.

An OE user can execute an authorized program that is stored in an MVS data set. See “How to use the sticky bit” on page 237 for a discussion of the sticky bit. In this case, the called program is executed in its own address space, via a *fork* from the caller's address space.

Once executing in an authorized mode, the *fork* function does not propagate this `APF` mode to a child process.

An authorized program can call various system functions (APIs) that an unauthorized program cannot use. Being authorized does not automatically provide any additional security (`RACF`) bypasses. An authorized program can, through various techniques, manipulate in-storage `RACF` control blocks and thereby grant itself additional privileges -- but this must be done by logic in the application program.

A somewhat more precise definition of a program executing as an authorized program is one that:

- ▶ Was linked `AC=1` into an authorized library (or HFS equivalent) and is executing as the job step program or as a proper descendant of the job step program through a chain of other `APF` authorized programs. In this case it is executing as an *APF authorized* program. OR,

- ▶ A program that, by any means, is executing in supervisor state or a system protection key (0-7). The *any means* may be through SVC code, PPT controls, and so forth. The restricted APIs will accept APF authorized callers, or callers in supervisor state, or callers in a system protection key. (There are other states that are also accepted, but these are more exotic.)

Note that authorization refers to the program being executed, not the user involved. Common terminology is not precise, and the term *authorized user* is frequently used. In this case, the *user* means the program requesting a restricted interface; this is unrelated to the human *user* that started the program⁴¹

A reference to an *authorized user* that clearly refers to a person can mean a variety of things. It might mean a userid with RACF SPECIAL authority,⁴² it might mean the userid has access to whatever data sets are needed for an application being discussed, it might mean the userid has SAF/RACF access to generic profiles protecting logical paths within the application, and so forth.

Experienced MVS users automatically determine the meaning of *authorized user* in a particular context -- usually.

5.113 How to create an MVS load module from OE

You can create an MVS load module instead of an HFS executable by using the following format:

```
cc -o '//ogden.lib.load(hello)'" hello.c
```

You need to get the single and double quote marks right. In this example, the double quote marks *protect* the // characters from being interpreted by the OE shell. The single quote marks indicate a fully-qualified data set name. (The // characters indicate that an MVS data set name is being specified.)

5.114 How to use the at command

OS/390 UNIX System Services provides the **cron** and **at** commands that are common in other UNIX systems. The **cron** services are defined (via crontab) by a UID=0 user. The **at** service can be used by any userid (and any UID); the normal UNIX controls for the **at** command apply. That is,

- ▶ Your administrator (UID=0) must create `/usr/lib/cron/at.allow` and place your userid in this file, OR
- ▶ Your administrator (UID=0) can create `/usr/lib/cron/at.deny` and *not* place your userid in this file. If the `at.deny` file exists AND if the `at.allow` file does not exist, then all users (except those listed in `at.deny`) can use the **at** command.

In a larger production environment you probably do not want all users to have access to the **at** command. Both **at** and **cron** are explained in the *OpenEdition Command Reference* (SC28-1892) manual.

⁴¹ Of course, the userid involved must have execution access to the library containing the program.

⁴² This only grants the userid the authority to issue any RACF command. Using this, the userid can grant itself access to any RACF-controlled resource.

5.115 How to use NOHUP to start a daemon

Consider an `/etc/rc` script with a single line in it:

```
nohup aaad      (where aaad is the name of an executable)
```

The shell will issue a fork for the **nohup**. As soon as the fork completes (from the parent's viewpoint), the parent process will look for its next command. Finding none, it will exit. The exit sends a `SIGHUP` to the child. In parallel, the child is executing the **nohup** command--the **nohup** command is not effective until it has executed enough instructions to establish the `nohup` condition to block `SIGHUP` signals from the parent. This makes a race condition with erratic results. You can prevent the problem by changing the initial script to:

```
nohup aaad
sleep 5
```

5.116 How to use the TSO command from OE

From the OpenEdition shell, you can issue commands such as:

```
tso listc
tso -t listc
tso -o listc
```

to execute TSO commands from your OMVS shell. There are two possible flags, `-t` and `-o`. (If no flag is specified, it defaults to the `-t`.)

The `-o` flag means that your TSO/OMVS session will be used to execute the indicated command. Almost all normal TSO functions are available this way, including *authorized* TSO commands. You can use the `-o` flag only if your OpenEdition session was started via TSO and OMVS. You cannot use `-o` if your OpenEdition session was started via telnet (or rlogin) directly to OpenEdition.

The `-t` flag indicates that the OpenEdition TSO command should create a mini-TSO environment in order to execute the command. This would be used if you do not have a TSO session; that is, if your OpenEdition session was started by telnet (or rlogin) directly to OpenEdition. The mini-TSO environment cannot handle *authorized* TSO commands such as `SUBMIT`.

5.117 How to source a script (dot command)

If you execute a shell script (by naming the script in the OpenEdition command line), a child process will be forked for the shell script. Any environmental variables set by the script are lost once the script ends. That is, environmental variables set by a child process are not returned to the parent process.

The *dot* command (a period) is used to run a shell script in the current shell. When the shell script ends, any environmental variables it sets are still available. Assume `scriptx` is a file, in the current directory, containing a shell script:

```
> scriptx          (creates a child process to run it)
> . script         (executes in the current process)
```

This is standard UNIX behavior, and there are no unique OS/390 considerations. We mention it because traditional MVS users are often not sensitive to this technique.

5.118 How to restart inetd

When inetd starts, it creates the file `/etc/inetd.pid` with no data in the file. The existence of this file means that inetd is running. When the system is stopped, inetd will delete the file.

You may encounter situations where you cannot start inetd. This could be due to the existence of `/etc/inetd.pid` when it should not exist. That is, for some reason the file was not deleted when the system was shut down previously. If inetd is not running, you can use the `rm` command to delete `/etc/inetd.pid`.

If the program was started from JCL, the file name may be `/etc/INETD.pid`.

5.119 How to reclaim disk space in an HFS

You should not need to take any action to reclaim space in an HFS after you delete a file or directory. (There is no equivalent to a PDS compression or a volume defragmentation.) However, users have noted that space reclaiming is not instantaneous -- some minutes may be required. Apparently there is a timed function that is used to reclaim HFS disk space. (The author has not found any documentation about this and is only reporting comments from other users.) In an extreme situation, UNMOUNT and reMOUNTing the HFS forced the reclaim process to complete.

In normal ITSO environments, including various workshops, we have not noted any problems with space reclaiming. We conclude that this may be a problem only in unusual situations.

5.120 How to determine the size of a loaded program

The SYSPRINT (output listing) from the binder reports the size of the program (or the *length of the text*). This is the amount of storage used by the loaded program text. It does not include the control information that is stored with the program on disk. This control information includes relocation data, symbol tables, and so forth. Thus the reported size of a file does not directly indicate the amount of storage used by the loaded module.

5.121 How to prevent a user from using OE

You can prevent a specific user from using OpenEdition, even if you have defined a default OpenEdition userid/groupid. If you define an OMVS segment for the user with no UID specified, that user cannot access OMVS. For example:

```
ALU badguy OMVS(NOUID)
```

5.122 How to estimate SMF output from OMVS

If you have SMF Type 30 recording enabled, you will see the following records in SMF:

```
Spawn (local spawn)
  No separate SMF record

Spawn (into new address space)
  subtype 1 (job start record)
  subtype 4 (jobstep total information)
  subtype 5 (jobstep termination record)
```

```

Fork followed by EXEC (in child process)
  subtype 1 (job start record)
  subtype 4 (jobstep before EXEC)
  subtype 4 (jobstep for final information)
  subtype 5 (jobstep termination record)

```

If you have interval accounting turned on for OMVS, you will also have subtype 3 and 2 records:

```

Spawn (local spawn)
  No separate SMF record
  subtype 2 in intervals (if interval is long enough)

```

```

Spawn (into new address space)
  subtype 1 (job start record)
  subtype 2 in intervals (if interval is long enough)
  subtype 3 (last interval of step)
  subtype 4 (jobstep total information)
  subtype 5 (jobstep termination record)

```

```

Fork followed by EXEC (in child process)
  subtype 1 (job start record)
  subtype 2 in intervals (if interval is long enough)
  subtype 3 (last interval of step)
  subtype 4 (jobstep before EXEC)
  subtype 2 in intervals (if interval is long enough)
  subtype 3 (last interval of step)
  subtype 4 (jobstep for final information)
  subtype 5 (jobstep termination record)

```

This may be much more SMF data than you expect to see, based on TSO experience. TSO attaches commands into the same address space, while OMVS fork and non-local spawn deal with new address spaces for every command of every OMVS user and this generates a lot more address space-related data.

Using `_BPX_SHAREAS=YES` will obviously reduce the number of records written (as well as improving performance for many other reasons). You can edit `SMFPRMxx` in `PARMLIB` to change your recording parameters. For example, if you want only subtype 4 and 5 records, you could specify:

```
SUBSYS(OMVS,TYPE(30(4,5)),NOINTERVAL)
```

See *Initialization and Tuning Reference* for more information about `SMFPRMxx` controls.

5.123 How to move an executable from HFS to LPA

You can move an executable from HFS to LPA (or to any MVS load library) by relinking it. For example:

```

//OGDENL JOB 1,OGDEN,MSGCLASS=X
//LINK EXEC PGM=IEWL,
// PARM='LIST,XREF,LET,RENT,REUS,AMODE=31,RMODE=ANY,CASE=MIXED'
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(10,10))
//SYSPRINT DD SYSOUT=*
//SYSLMOD DD DISP=SHR,SYS1.LPALIB
//HFSIN DD PATH='/u/ogden/'
//SYSLIN DD *
INCLUDE HFSIN(myexec)

```

```
ENTRY  CEESTART
NAME   MYOWN(R)
/*
```

There are several considerations here. You must tailor the PARM operands to fit your module. You would probably use these as shown here, except for the RENT and REUS operands; these mean reentrant and reusable. RENT and REUS are required for LPA, but not required for common load libraries; we suggest you omit these two unless you are certain they apply.

Note that the PATH is a directory. As shown, file /u/ogden/myexec will be the input to the link edit. All straightforward programs produced by LE compilers (which includes the current C compilers) use CEESTART as the entry point for a program. The target of this linkedit will be SYS1.LPALIB(MYOWN). The (R) operand indicates that an existing MYOWN module should be replaced.

5.124 AMODE24 programs from OE

Executing AMODE24 programs under Open Edition (that is, executing them from an HFS) is not supported. You will need to recompile the program using AMODE31. (This is the default when using the C compiler under OMVS.)

5.125 How to bounce the Web Server

You can perform a quick restart of the Web Server with the following MVS operator command:

```
F HTTPD1,APPL=-RESTART
```

This causes the web server to reread its configuration file and activate new parameters in it. This type of restart is much faster than stopping the web server and then starting it again. Most, but not all, configuration changes (such as adding new DLLs and Servlets) can be activated this way.

5.126 How to use the sticky bit

If the sticky bit is on (in an HFS file) and you execute this file, the system will:

- ▶ Truncate the name to eight characters.
- ▶ Drop any suffix (such as .so or .exe).
- ▶ Change it to upper case.
- ▶ Use the standard MVS search order to find a module with the resulting name. (If a STEPLIB is to be part of this path, it must be established prior to calling the target program).
- ▶ If the module is not found in the MVS search order, the initially specified HFS file is executed.

This process will not work correctly if the HFS file name contains special characters, such as an underscore. If you need to execute such file names via an MVS search order, you can use an external link to point to an MVS load library member.

MVS modules found this way are not eligible for the *local spawn* function, regardless of the value of the `_BPX_SHAREAS` variable.

If an MVS executable module is found, the system will attempt to execute it with the appropriate MVS attributes. It may be an authorized program in an authorized library; this will succeed only if the caller's address space is acceptable for authorized programs. This usually means that all previous programs that have executed in this address space have also been authorized. If this is not true (that is, the program to be executed is authorized, but the address space has previously executed unauthorized programs) the address space will ABEND. (This is to prevent system integrity exposures.)

5.127 Understand CR, LF, NL in ASCII and EBCDIC

You may find some confusion about certain control characters when moving between ASCII and EBCDIC.

ASCII	EBCDIC
LF = x'0A'	LF = x'25'
CR = x'0D'	CR = x'0D'
NL = ??	NL = x'15'

The problem is that typical ASCII does not really use a NL (new line) character. When a UNIX person says NL he really means LF (line feed). EBCDIC has a NL character, different from the LF character. OpenEdition ends text lines with x'15', a defined EBCDIC NL character. The C constant "\n" is x'15' for OpenEdition EBCDIC.

Various translation tables, IBM-provided as well as used coded, may not handle this the way you want. (IBM provides MANY translation tables for OpenEdition, TCP/IP, NFS, and so forth. The challenge is to select the right ones for your environment.)

5.128 MVS data sets that can move to OE via OPUT

The OPUT command copies MVS data sets to OpenEdition files. It is designed to work with sequential files in F, FB, V, or VB format. It will not work with spanned records. It will not work with VSAM (or ISAM) data sets. Members of a PDS or PDSE are considered as sequential data sets. (A complete PDS or PDSE can be copied using the OPUTX command; the same rules apply.)

A data set with RECFM=U is assumed to be a load module library. If it is not, the results are unpredictable.

Except for load modules, the contents of a PDS directory are not copied. (The directory data for load modules is processed rather than copied.) ISPF statistics, for example, are kept in PDS directory entries. Programs using BPAM directly may keep other information in directory entries; this is lost when using OPUT.

Data sets processed with BDAM may or may not copy correctly, depending on how their DCB attributes are defined. Typical BDAM programming does not easily translate to typical OE programming, so there may not be much point in copying data sets used with BDAM.

5.129 The largest file size in HFS

With DFSMS release 1.5, which permits spanning an HFS PDSE over multiple volumes, the largest file system size is 59 volumes. If these are 3390-3 volumes, this would be 59*2.8 GB or 165 GB.

With the same release, the largest file size is approximately 2^{47} (if the file is sparse).

5.130 How to begin using dbx

The current AD CD-ROM systems, as delivered, cannot immediately use **dbx**. We expect this same situation will exist in later releases. The problem is that there is a module conflict between the LE Debug Tool and **dbx**.

In order to use **dbx**, you need to run the job in SYS1.SAMPLIB(CEEEVDBG). Alter this job to place the resulting load module in any private load library. We used OGDEN.LIB.LOAD. Do not place it in a LNKLST or LPA library (or in SYS1.SCEERUN) because it may interfere with the Language Environment Debug Tool.

Once you have assembled and linked the module, use the following shell command in your OMVS session before starting **dbx**:

```
export STEPLIB='OGDEN.LIB.LOAD'
```

using the correct library name, of course. If you do not do this, several **dbx** subcommands will yield unexpected results. In particular, any *step* functions may not step correctly.

5.131 SMP/E

System Modification Program Extended (SMP/E) is a tool designed to manage the installation, service (fixes), and modifications of software products. A generalized, step-by-step formula for using SMP/E is far beyond the scope of this document--and may not be possible because there are so many paths for SMP/E usage. A good reference for a concise overview of SMP/E usage is *MVS Systems Programming*, by Dave Elder-Vass (ISBN 0-07-707767-9, McGraw-Hill Book Company).

As a user of the AD CD-ROM system, you may need SMP/E for any of the following:

- ▶ Applying PTFs (fixes) to the AD system. The PTFs may be sent to you when you call IBM for a software problem.
- ▶ Applying PUT tapes. A PUT tape is a collection of PTFs; a new PUT tape may be released every 4 - 6 weeks. Some installations apply all PUT tapes, on the theory that this may forestall software problems.
- ▶ Installing new software products. In general, almost all IBM OS/390 products require SMP/E for installation. Many third-party products offer an option of SMP/E or non-SMP/E installation.
- ▶ Maintaining your own applications through the SMP/E process. This is your option, but is not required.
- ▶ Applying and maintaining local modifications to OS/390 or other IBM (or third-party) software products.

5.131.1 More about SMP/E

SMP/E frequently uses the term SYSMOD, meaning System Modification. This definition, system modification, sometimes imparts a misleading tone. Practically everything that SMP/E deals with is a SYSMOD, and it is easiest to think of a SYSMOD as simply a module or update for a module. You may use SMP/E to update or replace modules (that is, make a system modification) or to install modules (new product (FMID), or new modules within an existing product.)

The building blocks begin with SOURCE code and CSECTs, which are the lowest level of program (or part of a program, such as a subroutine or function). One or more CSECTs (or high level language equivalents) are assembled or compiled together to produce an OBJECT module. One or more object modules are linked together to produce a LOAD module. Several load modules may produce a software product, which can be given a unique identifier, or FMID, by SMP/E control statements. A PTF provides SOURCE, or OBJECT modules, or LOAD modules to update an existing load module that is part of an FMID. An APAR does the same thing, but it is considered a more temporary fix that will later be replaced by a PTF. USERMODs are local modifications (not generated by IBM) that are handled about the same as PTFs.

In the SMP/E environment, there are two distinct types of libraries. They are referred to as distribution libraries and target libraries. You IPL and run with the target libraries. Distribution libraries contain all the elements, such as modules and macros, that were used to build your target libraries. Today, this building process is usually done by IBM, as part of creating a ServerPac, for example. (The AD CD-ROM systems are created using a standard ServerPac.) IBM provides the distribution libraries because PTF fixes (and, sometimes, new products you may install) require the distribution libraries in order to build a new working module for your target libraries. In a general sense, you have two copies of every module in your OS/390 system -- one in a distribution library and one in a target library. You do not need the distribution libraries to IPL and run your system. You do need the distribution libraries to apply PTF fixes and, possibly, to install new software products.

The Consolidated Software Inventory (CSI) data sets contain all the information SMP/E needs to track the distribution and target libraries. The CSI contains an entry for each element in its libraries. The CSI entries contain the element name, type, history, how the element was introduced into the system, and a pointer to the element in the distribution and target libraries. The CSI does not contain the element itself, but rather a description of the element it represents. You can have multiple CSIs, depending on how the initial systems programmer set up your system. He might have elected, for example, to place all CICS-related software in its own CSI.

Look at the basic processing commands you need to use SMP/E:

- ▶ Setting the zone you want to work on
Before processing SMP/E commands, you must first set the zone on which you want SMP/E to work (global, target, or distribution). You do this by issuing the SET command. The SET command can also be used to request a particular set of predefined processing options. For more information about the SET command, refer to the *SMP/E Reference* manual.
- ▶ Receiving the SYSMOD into SMP/E's Data Sets
For SMP/E to install a SYSMOD, the SYSMOD must be *received* into data sets that can be used by SMP/E. The SMP/E RECEIVE command performs the task of copying the SYSMOD from its distribution medium.
- ▶ Applying the SYSMOD to the Target Libraries

Once a SYSMOD has been received, you want to *apply* the SYSMOD to the appropriate target libraries. The SMP/E APPLY command invokes various system utilities to install the SYSMOD's elements into the target libraries.

- ▶ Restoring the Target Libraries to the previous level

Should you experience problems after applying a SYSMOD, you may want to *restore* its elements in error to a previous and stable level. The SMP/E RESTORE command replaces a failing element with a copy from the distribution libraries.

- ▶ Accepting the SYSMOD and updating the Distribution Libraries

After you have performed a SYSMOD RECEIVE and APPLY, you want to *accept* the elements into the distribution libraries for backup. However, this should be done only after you are satisfied with the performance and stability of the elements of the SYSMOD. Once you ACCEPT a SYSMOD, you cannot RESTORE its element to a previous level. The SMP/E ACCEPT command updates the distribution libraries so they are available for backup of any future SYSMODs.

- ▶ Displaying SMP/E Data

The SMP/E CSI and other primary data sets contain a great deal of information you may find useful when installing new elements or functions, preparing user modifications, or debugging problems. There are several ways SMP/E allows you to display that information, as well as information about modules, macros, and other elements:

- Query dialogs display specific information through interactive dialogs with SMP/E.

We suggest that new SMP/E users work with the ISPF SMP/E panels rather than attempt to use command statements with SMP/E batch jobs.

- The LIST command can generate a variety of information lists.
- REPORT commands check, compare, and generate listings about the contents of zones on your system.

- ▶ OS/390 SMP/E structure

The AD system has a number of CSI data sets. The easiest way to list them is to go to the ISPF 3.4 panel and enter:

```
Dsname Level . . . . **.CSI
Volume serial . . .
```

This will search all catalogs for data sets with a qualifier of CSI. The qualifier name CSI is not reserved, but it is unlikely that non-SMP/E data sets would contain this name.

- ▶ Where to find more information

You can find more information in the MVS Software Management Cookbook, GG24-3481. The cookbook covers a broader scope than simply SMP/E considerations. It explains the tasks necessary to maintain an MVS environment. Topics include maintenance rules, system design for maintenance and software installation, testing, and implementation. You can read the entire document to gain an overall understanding of large system maintenance, or you can read specific sections as you need during a system update.

5.131.2 How to display PTF status

A common question is “Do you have PTF number X installed?” You can use the ISPF SMP/E panels to help answer this question. The first screen of the SMP/E panels is roughly shown in Figure 5-34 on page 242. (The screen images illustrated in this section do not reflect all the contents of the actual panels. A subset is illustrated for brevity.)

```
                                SMP/E PRIMARY OPTION MENU

====> 3

      1 ADMINISTRATION
      2 SYSMOD MANAGEMENT
      3 QUERY
      4 COMMAND GENERATION
      5 RECEIVE

SMPCSI DATA SET ====> 'mvs.global.csi'

Generate DD statement ====> yes
```

Figure 5-34 SMP/E Primary Option menu

To use the panels, you must supply the SMPCSI data set name requested in this panel. The name shown in the example is used by several releases of the AD system. It will *almost always* have a middle qualifier of GLOBAL, and should *always* have a last qualifier of CSI. You can start a query with other CSI data sets, but our example will use the GLOBAL CSI. Select option 3 (QUERY). This should produce something like Figure 5-35.

```
                                QUERY SELECTION MENU

====> 2

      1 CSI QUERY
      2 CROSS-ZONE QUERY
      3 SOURCEID QUERY
```

Figure 5-35 SMP/E Query Selection menu

You would then select option 2 (CROSS-ZONE QUERY). The next screen should be something like Figure 5-36.

```
                                CROSS-ZONE QUERY

====>

      ENTRY TYPE ====> sysmod

      ENTRY NAME ====> hbb6601
```

Figure 5-36 SMP/E Cross-Zone Query

Enter the PTF number on this panel. The ENTRY TYPE will usually be SYSMOD. This should produce results something like shown in Figure 5-37 on page 243


```

                                CSI CROSS-ZONE QUERY ENTRY SELECTION

====>
Entry Type:  SYSMOD
Entry Name:  HBB6601

          * - Entry not found
          ** - Zone could not be allocated

                                STATUS
-----
ZONE      TYPE      FMID      STATUS      DATE      TIME      REWORK
-----
CICSDZN   *
GLOBAL    *
MVSD100   FUNCTION HBB6601   ACC          96.162    20:27:29  1996103
s MVST100   FUNCTION HBB6601   APP          96.163    13:07:00  1996103

```

Figure 5-37 SMP/E CSI Cross-Zone Query Entry Selection

This screen indicates that the name HBB6601 was known in two zones. It was accepted in the DLIB zone (MVSD100) and applied in the MVS target zone (MVST100). You probably want to know about the target zone. Position the cursor before this zone line, and enter the character S (and press Enter). This will display all known information about this sysmod in the target zone. (Your zone names may not be exactly as shown in the illustrations.)

If you are not familiar with SMP/E, or the CSI structure of your system, we suggest using the cross-zone query shown here. It will check all the CSIs known to this global CSI for your module.

There can be multiple GLOBAL CSIs on a system. For example, someone in your organization might install a major program product, using SMP/E for installation, and decide to create a totally separate SMP/E CSI structure. Any name can be used for CSIs, although the low-level data set name qualifier must be CSI. The installer should use a middle qualifier of GLOBAL, but this is not an absolute requirement.

5.132 How to manage spool space

JES2 spool space is used to store JCL, in-stream data sets, and SYSOUT for jobs, started tasks, and TSO sessions. The SYSOUT stored in the JES2 spool are punch and print files. The few punch files produced usually have a special use, for example the TSO transmit/receive facility uses punch files but this is transparent to the users of these facilities.

Print files however are used extensively and most batch jobs, as well as many started tasks and subsystems, will produce print SYSOUT. This printout is stored on the spool. From the spool, the print files can be sent to real printers or viewed by users of the system using TSO facilities such as SDSF, ISPF 3.8 or the OUTPUT command or manipulated by TSO facilities (or specialized program products such as RMDS - Report Management and Distribution System).

As the typical OS/390 workload is processed, the JES2 spool becomes filled with a large number output files. The management of this data becomes an issue. Some areas of concern include:

- ▶ The spool is a finite size and unless positive action is taken will inevitably fill up.

- ▶ Many users, especially developers, tend to leave their output files sitting in spool. That is, a compiler listing or core dump listing is left in spool because they *might* want to look at it again, sometime.
- ▶ Files on the spool have different priorities. Some jobs submitted to the system require immediate execution while others may wait until overnight processing. Similarly, some print output will require immediate printing, for other print output a 24 hour turnaround may be adequate.
- ▶ Typically, print output to be printed to paper will be printed to any one of a number of locations. These could be printers attached to the channel on your system, remotely attached via VTAM or sent to another OS/390, VM or VSE system to be printed on a printer controlled by the remote system. These are referred to as destinations by JES2.

By far the most common and easiest method of control over the JES2 Spool is by using the SDSF program product (distributed with the AD system but must be separately ordered with other S/390 configurations).

Some items of interest when using SDSF:

- ▶ To be effective in managing all the spool, you will require full access to all data on the spool. As with most components in OS/390 over recent times a lot of work has been done on the security interface. SDSF can use its own internal security (implemented by an assembler coded module or a SYS1.PARMLIB member) or with calls to a security product such as RACF. Further details about SDSF security can be found in “Understanding basic SDSF security” on page 181.
- ▶ The following commands will display the types of data on the spool:
 - I (display input files)
 - O (display output files)
 - H (display held output files)
- ▶ If you wish to see ALL the files use the following filter commands:
 - PREFIX * (Display all jobnames)
 - DEST (Display jobs for all destinations)
 - OWNER * (Display jobs for all owners)
 - FILTER OFF (Remove any filter restrictions)
- ▶ When you have displayed a JES2 queue, that is input (I), output (O) or held (H) queue, you can manipulate the jobs on the queue by overtyping the field to the right of the jobname. This is the best way to change a jobs Class, Priority or Destination.
- ▶ Other actions may be taken against a file on one of the queues by using an action character to the left of a jobname. The following are the most common action characters. (Use PF1 then option 3 to get help for all available action characters from that display):
 - P - Purge a job (from any queue; i.e., DELETE the job completely)
 - A - Release a job (from the Input Queue)
 - O - Release a job (from the Held queue to the Output queue)
 - C - Cancel a job (from the DA display)

If SDSF is unavailable, you can still manage the JES2 spool by using JES2 operator commands from a system or sub-system console. A few common and useful commands follow. Note that many JES2 commands require either the alphanumeric name itself or a number for the job, started task, or TSO session (entered as Jnnnnn, Snnnnn or Tnnnnn). These are collectively referred to as the job_id below:

- ▶ **\$D'xxxxxxx'** Display status and job_id of all jobs/STC/TSO with the name xxxxxx. The single quote marks around the job name are required.
- ▶ **\$C job_id** Cancel, used for a running job, STC, or TSO logon.

- ▶ **\$P job_id** Purge the job_id from the system.
- ▶ **\$A job_id** Release a held job.
- ▶ **\$DN,X=XEQ** Display all jobs waiting for execution.
- ▶ **\$DQ** Display number of jobs in each type of JES2 queue and how much of the spool is used.
- ▶ **\$DPRTnn** Display the characteristics of printer nn, mainly details of what attributes are required for output to be eligible to print on the printer.
- ▶ **\$SPRTnn** Start printer nn (JES2 will then print all ready output eligible to print on this printer).
- ▶ **\$pprtnn** Drain (stop) printer number nn.

The commands described above that use a job_id parameter of the form Jnnnn, Snnnn or Tnnnn can also be used to affect a range of jobs. For example, if you want to purge a block of jobs you could use:

```
$PJmmm-nnnn
```

where mmmm is the number of the first job to be cancelled and nnnn is the last job number.

Important: Be careful using this command. \$PJ1-99999 will delete all job output from your system!

If you wish to completely clean your spool (delete all input and output from all queues) you can COLD start JES2. This can be achieved by replying to the \$HASP426 SPECIFY OPTIONS - JES2 startup message with COLD. You will also need to enter \$\$ after the cold start before JES2 will start performing work. Cold starting JES2 occasionally will also remove unwanted rubbish from the spool (if you manually delete all entries from the spool a spool display will not show empty).

If you neglect the spool and it becomes full, cold starting JES2 may be the only way to recover.

Commands such as the following can be used to provide a less drastic cleanup:

- ▶ **\$OQ,Q=x,CANCEL,A=y** Cancel all output in class x older than y days.
- ▶ **\$PQ,Q=x,A=y** Cancel ready output in class x older than y days.

You can use JES2 automatic command processing to execute a given command every day at a set time. The syntax is:

```
$TA,T=hh.mm,'command.....'
```

The following example illustrates how to use these commands. Insert the following commands at the bottom of the JES2 startup initialization deck (JES2PARM member of SYS1.PARMLIB, see "How to work with PARMLIB" on page 110 for details on safely modifying SYS1.PARMLIB).

```
$TA,T=00.10,'$OQ,Q=OTX,CANCEL,A=10'  
$TA,T=00.15,'$PQ,Q=AJ6,A=10'  
$SA,ALL (start automatic command processing)
```

These commands would result in JES2 deleting all held output in classes O, T and X older than 10 days at 10 minutes past midnight each day and deleting all ready output in classes A, J and 6 older than 10 days at 15 minutes past midnight each day.

5.133 How to hot start JES2

In rare cases you may want to stop and restart JES2 while TSO is running. This can be used, for example, to try a change to JES2PARMS. (You should always have an alternate JES2 parameter member that is known to work. This is your fallback recovery if you corrupt the normal JES2PARM member.) The theory is that you can change JES2PARM (using TSO) and stop and restart JES2 (without stopping TSO or VTAM) to use the changed JES2 parameters.

If there is a problem restarting JES2, you can re-edit JES2PARM (since TSO is still running) and try again. During the period when JES2 is down, its services are obviously not available; in particular, you cannot log onto TSO during this period.

The commands (from the MVS console) are:

```
$PJES2,ABEND
  reply END
S JES2
  reply NOREQ (or whatever other JES2 reply is needed)
```

We cannot recommend this technique for normal system customization or usage, but it may be useful in special cases. You should, of course, have minimal other activity (batch jobs, other TSO users, and so forth) in the system when planning to use this hot start function.

5.134 How to identify a data set user

If you are unable to run a job or use a data set from TSO because the data set is in use by another program, you can issue the following commands from the MVS console to determine which program (or user) has the data set allocated:

```
D GRS,RES=(SYSDSN,datasetname*)      or,
D GRS,RES=(*,datasetname*)           for example,
D GRS,RES=(SYSDSN,OGDEN.LIB.CNTL*)
```

You can use a full data set name, or only part of a data set name plus an asterisk. You can also use a full data set name plus the asterisk, although it may have no meaning in this case.

Do not try `D GRS,RES=(*,*)` unless you are on a small system with few users. This will generate a large amount of output on the operator console and there is no way to cancel the command.

If you are using the ISPF 3.4 functions, and you get a Data Set In Use message, you can press PF1 twice to get a list of the enqueued owners.

5.135 How to display basic IPL parameters

After most of the IPL activity completes, the commands:

```
D IPLINFO
D PARMLIB
D IOS,CONFIG
```

will display the most basic parameters for this IPL, including the LOADxx member used and the IODF used.

5.136 How to recover the MVS console

MVS does not like to lose its last console. If a *system console* (part of the Service Element function; accessed through the Operating System Messages icon) is not available, the system will stop if no MVS console is available. If a *system console* is available, MVS will automatically attempt to use it if all other consoles fail. If this happens, you need display the Operating System Messages function on the Support Element (or HMC) and to activate the system console before you can use it. A few messages will appear before you activate it, but you will be unable to enter any commands.

On the system console panel (of the Support Element), click **Response** and enter:

```
V CN(*),ACTIVATE
```

You should begin receiving MVS messages and you should be able to enter commands. You need to click **Response** in order to enter every command. If you are able to restart your normal MVS operator terminal, you can vary it online for use as a console. Assuming your normal MVS console address is 700:

```
V 700,CONSOLE
```

You should see messages begin to appear on the normal console. If it is working correctly, you can deactivate the system console:

```
V CN(*),DEACTIVATE
```

P/390 and S/390 Integrated Server systems did not have a *system console* function until fixpack 2.5.2 (for P/390s) and 3.2 (for Integrated Servers). See “P/390 fixpak” on page 4 for more information about this fixpack.

If a system console is not available, the following recovery procedures may help.

If you accidentally erase the master console (or otherwise confuse it), but the connection is not lost, try using the PA2 key. (Finding the PA2 key can be difficult, because various different 3270 emulators hide it in different places. With PCOM, position the mouse pointer in the emulator window and press the right mouse button. This provides a popup that contains PA2.)

If you lose the master console connection, due to TCP/IP problems for example (when using a P/390 or similar system), you should restore the connection and then use the External Interrupt from the operator panel. (For a P/390-based system, the operator functions are started by the Manual Operators icon.) If this partly restores the console, then use PA2. Wait several seconds between each action.

5.137 Capturing SVC dumps

“I sometimes see these two messages on the MVS console:

```
IEA794I SVC DUMP HAS CAPTURED: DUMPID=..... etc
```

```
IEA911E COMPLETE DUMP ON SYS1.DUMPxx ..... etc
```

and my system seems to slow down during this time. What is happening?”

An error has caused MVS to take a system dump, usually known as an SVC dump. (An application ABEND, which creates a dump using your SYSUDUMP DD statement, is a different kind of dump.) The SVC dump may disable (cease dispatching other jobs) for a short interval while it is preparing the dump. This is a very short interval and is probably not visible as a system slowdown.

The SVC dump sends the dump information to another address space, which acts as a holding area. When this has completed, you will see the IEA794I message. If the SVC dump is capturing several large address spaces (and filling the dump holding address space) this may cause substantial paging. This might cause a apparent slowdown for a period.

After the dump is captured (in the dump address space) it is written to a system dump data set. When this completes, you will see the IEA911E message. The system dump data set might be one of the SYS1.DUMPxx data sets (if any exist and are unused) or might be a dynamically allocated dump data set. You need to check your storage volumes and delete unwanted dump data sets; they can be quite large and, over time, can cause a shortage of scratch space.

An SVC dump is in a binary format. The only practical way to look at one of these is with the IPCS tool.

SVC dumps (that is, dumps taken by *the system* and written to SYS1.DUMPxx data sets (or dynamically allocated equivalents)) can sometimes cause the system to appear slower for several seconds. The following discussion concerns the mechanics of capturing one of these dumps. This is probably not a concern for typical P/390 development installations, but may be a concern for production installations.

The process of capturing data and placing it in a dump data space uses virtual memory and real memory. This may cause paging, and initial paging will be directed to Expanded Storage (if you have any defined).

Like any other address space, DUMPSRV's dump dataspace can be backed by expanded, real, and auxiliary storage. DUMPSRV does not care. RSM and SRM manage stealing of pages from real to expanded or auxiliary. Special processing is done, however, to mark dump dataspace pages as *old* with a UIC of 254 so they will be among the first to be stolen when real becomes constrained. SRM has special logic for migrating dump pages from expanded to auxiliary; DUMPSRV will always be the first address space selected for migration, and, if it satisfies the shortage, then no other address space's pages will be migrated. The end result is that we try to keep DUMPSRV from impacting expanded if at all possible. Installations should not try to fence expanded storage to protect it from DUMPSRV.

The key to the capture process is having a robust paging subsystem, as it effects the performance of both the page movement directly from real to auxiliary and the expanded storage migration processing. Dump processing does not direct captured data to a specific paging volume. Normal ASM selection occurs for each page that is sent to auxiliary and data will be written to all local page data sets at the same time.

Two local paging data sets do not make a robust paging subsystem. (However, it may be for an emulated I/O system, because the parallel channel and CU operations implied by more paging data sets are not relevant for the emulated DASD of current P/390-based systems.) For typical OS/390 systems, we usually recommend a minimum of four local page data sets; six is even better. While their size is important, as far as performance goes it is the number, physical separation, and underlying physical implementation that is key. (Again, note that these comments do not apply well to an emulated I/O system such as a P/390-based machine, since other constraints limit DASD performance and bandwidth.)

For example, if you assume:

- ▶ Expanded is mostly full, so that very little of the captured dump can stay there until written.
- ▶ The size of the dump is 2000 MB (a good sized DB2 dump)
- ▶ The average transfer rate to the page data set is 2.5 MB per second

then performance has these characteristics:

- ▶ With two local page data sets it will take around 6.5 minutes to transfer the dump dataspace data to them, and there will be contention on real (and expanded) storage for this period of time, potentially slowing other work on the system.
- ▶ With four local page data sets this drops to around 3.5 minutes.
- ▶ With six, this drops to around 2 minutes.

Of course, issues such as channel path usage, control unit contention, control unit caching, and others may increase or decrease the average data rate you will obtain. But two local page data sets are not enough to handle large SVC dumps efficiently.

On the other hand, if you have several hundred megabytes of expanded storage that is lightly used at the time of the dump, two page data sets may be enough, as very little data must be migrated to them. But you must plan for and configure for the worst case -- of a dump during prime time with little excess real or expanded to hold the captured data.

MAXSPACE is a system wide threshold on the sum of the space for all dumps that are captured and unwritten. A MAXSPACE of 1000 would allow 5 200M dumps, or 2 500M dumps, or 1 1000M dump to be captured and unwritten at any given time. The required value for MAXSPACE varies greatly from one installation to another. From an OS/390 perspective, 500M is usually enough for a dump from our code. But multi-address space applications, such as DB2, often require a much larger value, due to the fact that they dump multiple address spaces and include their buffer pools. If you run DB2 then 2500M is not unreasonable.

DUMPSRV will only capture a single dump at a time. Any request for another dump while one is currently being captured will be rejected. Once the capture process is complete a new dump request will be accepted, assuming that DUMPSRV is not currently sitting with captured dumps at its MAXSPACE value

MAXSPACE, mentioned several times here, can be displayed with the DISPLAY DUMP,OPTIONS (D D,O) command from the MVS console. It can be changed with the CHANGEDUMP (CD) command.

5.138 How to use STARTED task control

Your OS/390 system uses many *started tasks*. These are tasks (programs in address spaces) that you start from the OS/390 console or that OS/390 starts automatically during IPL. Examples of started tasks are initiators, JES2, VTAM, and so forth. RACF can control started tasks through *STARTED* class profiles. The primary purpose of a profile in RACF's *STARTED* class is to associate a userid with the started task.⁴³ The recent AD CD-ROM systems are provided with the RACF *STARTED* class activated, but with few profiles in it.

⁴³ The userid is used to check and provide access permissions to whatever data sets and resources the started task program requires. A started task does not automatically have permission to all resources.

If the STARTED class is activated, RACF will check (in the RACF database) for a profile for every task that is started. If no profile is found, it will check an assembled table named ICHRIN03. This table normally has only a default entry. If the default is used, RACF issues a console message and assigns a userid to the task that is equal to the started task name (which is usually the member name in SYS1.PROCLIB). This situation does no harm, but you might want to provide better control of started tasks,⁴⁴ perhaps without defining so many extra userids.

There are several ways to go about this, and we shown one approach here.

We define a few userids, with different levels of system access, to be used by started tasks. (The NOPASSWORD option for these userids means they cannot be used to log onto TSO or run batch jobs.)

```
AU start1 DEFLTGRP(SYS1) OWNER(SYS1) NOPASSWORD OPERATIONS OMVS(UID(0) HOME(/))
AU start2 DEFLTGRP(SYS1) OWNER(SYS1) NOPASSWORD OMVS(UID(0) HOME(/))
PERMIT 'SYS1.*' ID(start1) ACCESS(ALTER)
PERMIT 'SYS1.*' ID(start2) ACCESS(UPDATE)
```

These commands define two userids, *start1* and *start2*. The first one has more authority because it has RACF OPERATIONS privileges. Both are permitted to the 'SYS1.*' profile, but with different levels of access. (The PERMIT command for *start1* is probably unnecessary because of the OPERATIONS authority.) We want all our started tasks to use one of these two userids. Using these two userids, we can define three levels of *started task* authority:

- ▶ We can define started task names with the *trusted* attribute, which bypasses all further RACF checking.⁴⁵
- ▶ We can assign started tasks to start1, and these can access almost any data set (due to OPERATIONS authority).
- ▶ We can assign started tasks to start2, whose only unusual authority is to update any SYS1.* data set.

We first define a list of known system started tasks that require high authority:⁴⁶

```
SETR GENERIC(STARTED)
RDEF STARTED jes2      STDATA(USER(start1) GROUP(sys1) TRUSTED(YES))
RDEF STARTED lla      STDATA(USER(start1) GROUP(sys1) TRUSTED(YES))
RDEF STARTED vlf      STDATA(USER(start1) GROUP(sys1) TRUSTED(YES))
RDEF STARTED catalog  STDATA(USER(start1) GROUP(sys1) TRUSTED(YES))
RDEF STARTED dumpsrv  STDATA(USER(start1) GROUP(sys1) TRUSTED(YES))
RDEF STARTED ieevmpr  STDATA(USER(start1) GROUP(sys1) TRUSTED(YES))
RDEF STARTED smf      STDATA(USER(start1) GROUP(sys1) TRUSTED(YES))
RDEF STARTED vtam     STDATA(USER(start1) GROUP(sys1) TRUSTED(YES))
RDEF STARTED ixglogr  STDATA(USER(start1) GROUP(sys1) TRUSTED(YES))
RDEF STARTED xcfas    STDATA(USER(start1) GROUP(sys1) TRUSTED(YES))
RDEF STARTED tso      STDATA(USER(start1) GROUP(sys1) TRUSTED(YES))

RDEF STARTED nfss     STDATA(USER(start1) GROUP(sys1) TRUSTED(NO))

RDEF STARTED **      STDATA(USER(start2) GROUP(sys1) TRUSTED(NO))
SETR RACLIST(STARTED) REFRESH
```

⁴⁴ For best control, you would provide a STARTED class profile for all started tasks and then reassemble the ICHRIN03 table to remove the generic entry. This would prevent anyone from starting a task that does not have an associated STARTED profile.

⁴⁵ We are ignoring the *PRIVILEGED* attribute, which is about the same as the *TRUSTED* attribute.

⁴⁶ In recent AD system releases, there are already STARTED class profiles for FTP, TCPIP, web server, router, and a number of DCE functions.

The second group (only NFSS in this example) uses the start1 userid, but does not have the trusted attribute. The last RDEF, with the asterisks, defines a started class profile that matches any started task name not specifically defined earlier and this uses the lower privilege userid *start2*.

In order to start tasks that use OMVS, the userid must have an OMVS segment. As shown, the OMVS UID is zero for these userids; that is, this is a *superuser* or *root* user, and any started task that uses these profiles will have root authority.

Your installation will probably have slightly different requirements; in particular, you may have more highly-privileged started tasks that should be included. Some installations simply define a single started profile, named **, and make it *trusted*. This works, but lowers the overall security tone of the system.

The OS/390 R10 and z/OS R1.1 AD systems, as delivered, use only a few *started* class profiles and depend on the ICHRIN03 table to assign userids to most started tasks. This should be sufficient for many AD development installations. It would probably not be acceptable in a large production installation.

5.139 Getting started with RMF

RMF is a large, sophisticated, and somewhat complex product. It is widely used for a number of purposes, including:

- ▶ Casual observation (usually by systems programmers) of system performance, with attention given to any bottlenecks or unexpected results.
- ▶ Scheduled use as part of a formal system performance monitoring program.
- ▶ Capacity planning exercises, used in a variety of ways.

A complete description of RMF use is much beyond the scope of this redbook. We provide here a step-by-step guide to using a few basic functions through ISPF panels. In recent AD systems, RMF can be started from the extended primary menu (the “M” option). Other systems may have other ways to start it. The first RMF page is shown in Figure 5-38 on page 252. (The complete ISPF panels are not shown in these RMF examples, in order to keep the text more compact.)

```

RMF - Performance Management                                OS/390 2.10.0 RMF
Selection ==>

Enter selection number or command on selection line.

  1 Postprocessor      Postprocessor reports for Monitor I, II, and III      (PP)
  2 Monitor II         Snapshot reporting with Monitor II          (M2)
  3 Monitor III        Interactive performance analysis with Monitor III (M3)

U USER                User-written applications (add your own ...)      (US)

R RMFPP               Performance analysis with the Spreadsheet Reporter
P PM of OS/390        PM of OS/390 Java Edition
N News                What's new in OS/390 2.10 RMF

                        T TUTORIAL      X EXIT

RMF Home Page:      http://www.ibm.com/s390/rmf/

```

Figure 5-38 RMF - initial panel

You may receive a message about "...creating P390.RMFOS210.ISPTABLE"; this can be ignored. Option 2 (Monitor II) is the only option that can be used without additional setup. Selecting this option should produce the panel shown in Figure 5-1. The first four options in this panel can be used.

```

RMF Monitor II Primary Menu                                OS/390 2.10.0 RMF
Selection ==>

Enter selection number or command on selection line.

  1 Address Spaces     Address space reports
  2 I/O Subsystem      I/O Queuing, Device, Channel, and HFS reports
  3 Resource           Enqueue, Storage, SRM, and other resource reports

L Library Lists       Program library information
U User                User-written reports (add your own...)

                        T TUTORIAL      X EXIT

```

Table 5-1 RMF - Monitor II Primary Menu

Select option 1 (Address Spaces); this should produce the panel shown in Figure 5-39 on page 253. All of the options in this menu may be used. Start by selecting option 1 (Address space resource data).

```

RMF Monitor II Address Space Report Selection Menu
Selection ==>

Enter selection number or command on selection line.

1 ARD          Address space resource data
2 ASD          Address space state data
3 ASRM         Address space SRM data

4 ARDJ         Address space resource data by jobname
5 ASDJ         Address space state data by jobname
6 ASRMJ        Address space SRM data by jobname

```

Figure 5-39 RMF - Monitor II Address Space Report Selection Menu

This should produce the display in Figure 5-40, although the names of your jobs will be different, of course. There is a great deal of information in this report. Some columns (generally on the left side) are cumulative or represent state information. Other columns (on the right side and the header line) represent delta information, obtained since the last *report*. (Every time you press Enter on the terminal, you receive a new *report*.) You may need to scroll forward and backward to see all jobs (address spaces).

```

RMF - ARD Address Space Resource Data
Command ==>
Line 1 of 35
Scroll ==> PAGE

MIG= 119K CPU= 20    UIC= 254 PR= 0    System= SYS1 Total

14:57:24 DEV  FF PRV LSQA LSQA X  SRM TCB  CPU  EXCP SWAP LPA CSA NVI V&H
JOBNAME  CONN 16M FF  CSF  ESF M CR ABS  TIME  TIME RATE  RATE  RT  RT  RT  RT

*MASTER* 0.000 0 235 93 0 0.0 90.28 761.7 0.00 0.00 0.0 0.0 0.0 0.0
PCAUTH 0.000 0 2 28 0 X 0.0 0.17 0.18 0.00 0.00 0.0 0.0 0.0 0.0
RASP 0.000 --- --- ---- --- X 0.0 0.09 1.97 0.00 0.00 0.0 0.0 0.0 0.0
TRACE 0.000 0 3 39 0 X 0.0 0.09 0.11 0.00 0.00 0.0 0.0 0.0 0.0
DUMPSRV 0.000 0 2 63 0 0.0 8.05 10.22 0.00 0.00 0.0 0.0 0.0 0.0
XCFAS 0.000 0 277 1119 0 X 0.0 441.1 458.6 3.00 0.00 0.0 0.0 0.0 0.0
GRS 0.000 0 27 34 0 X 0.0 0.33 0.36 0.00 0.00 0.0 0.0 0.0 0.0
SMXC 0.000 0 2 132 0 0.0 84.72 108.8 0.00 0.00 0.0 0.0 0.0 0.0
SYSBMAS 0.000 0 6 42 0 0.0 3.33 3.69 0.00 0.00 0.0 0.0 0.0 0.0
CONSOLE 0.000 0 2 40 0 X 0.0 14.00 14.20 0.00 0.00 0.0 0.0 0.0 0.0
WLM 0.000 0 51 139 0 X 0.0 269.8 314.7 0.00 0.00 0.0 0.0 0.0 0.0
ANTMAIN 0.000 0 8 139 12 X 0.0 19.30 21.05 0.00 0.00 0.0 0.0 0.0 0.0
ANTAS000 0.000 0 4 70 11 X 0.0 1.16 1.28 0.00 0.00 0.0 0.0 0.0 0.0
OMVS 0.000 0 77 224 0 X 0.0 40.14 53.93 0.00 0.00 0.0 0.0 0.0 0.0
JESXCF 0.000 0 6 54 7 X 0.0 25.99 35.60 0.00 0.00 0.0 0.0 0.0 0.0
ALLOCAS 0.000 0 2 24 0 X 0.0 0.05 0.05 0.00 0.00 0.0 0.0 0.0 0.0
IOSAS 0.000 0 29 58 0 X 0.0 2.99 3.26 0.00 0.00 0.0 0.0 0.0 0.0
MMS 0.000 0 2 48 0 X 0.0 1.02 1.04 0.00 0.00 0.0 0.0 0.0 0.0
IXGLOGR 0.000 0 4 132 0 X 0.0 4.62 4.76 0.00 0.00 0.0 0.0 0.0 0.0
SMF 0.000 0 2 43 0 X 0.0 0.02 0.02 0.00 0.00 0.0 0.0 0.0 0.0
LLA 0.000 0 28 64 8 X 0.0 19.67 20.72 0.00 0.00 0.0 0.0 0.0 0.0
JES2AUX 0.000 0 12 22 0 0.0 0.02 0.03 0.00 0.00 0.0 0.0 0.0 0.0
PF 1=HELP 2=SPLIT 3=END 4=RETURN 5=RFIND 6=SORT
PF 7=UP 8=DOWN 9=SWAP 10=LEFT 11=RIGHT 12=RETRIEVE

```

Figure 5-40 RMF - ARD report

The following is a brief overview of the types of data in this display. Some of the fields represent complex topics and you should research these before seriously using the information reported by RMF.

- ▶ Header Line:
 - MIG is the average time (in seconds) that pages are held in expanded storage before they are migrated to disk paging data sets. The number shown in the example (119K) is a large number of seconds and indicates that there is, effectively, no movement of pages from expanded storage to paging data sets.
 - CPU is the CPU utilization since the last report.
 - UIC is the highest UIC (Unreferenced Interval Count) in the system at the time of the report. The highest possible value is 255. A large number indicates that there are central storage pages that (1) are still in memory (that is, have not been stolen), and (2) that these pages have not been referenced for a considerable number of seconds.
 - PR is the overall system page-in rate since the last report.
- ▶ Left side of report (cumulative or state information): (Some titles are read vertically.)
 - JOBNAME is the name of the address space.
 - DEV CONN is the device connection time (in seconds) since the beginning of the job. Larger numbers indicate (1) lots of I/O operations, and/or (2) contention for I/O devices.
 - FF BEL is the number of fixed pages (below the 16-megabyte *line*) for this job.
 - FF PRIV is the number of fixed pages (excluding LSQA) for the job.
 - LSQA CSF is the number of fixed LSQA pages in central storage for the job.
 - LSQA ESF is the number of fixed LSQA pages in expanded storage for the job.
 - X M indicates that this is primarily a cross-memory address space that provides services for other address spaces.
- ▶ The following can be cumulative or rate information, depending on how the RMF is configured. For the AD system, these provide cumulative information:
 - SRM ABS indicates the SRM service absorption rate.
 - TCB TIME is the number of seconds of CPU time used under TCBs for this job.
 - CPU TIME is the number of seconds of CPU time used under TCBs plus SRBs for this job.
- ▶ Rate Information:
 - EXCP RATE is expressed in EXCPs per second since the last report.
 - SWAP RATE is the paging rate expressed in pages per second since the last report.
 - LPA RT is the page-in rate from LPA, expressed in pages per second, since the last report.
 - CSA RT is the page-in rate from CSA, expressed in pages per second, since the last report.
 - NVI RT is the private page-in plus page-out rate (excluding VIO), expressed in pages per second, since the last report.
 - V&H RT is the VIO and hiperspace page-in plus page-out rate, expressed in pages per second, since the last report.

If no information is available for a report, a field is shown as dashes. The rate information is available only if an address space is in central storage.

You can exit from this RMF display by using PF3. RMF will return to the panel shown in Figure 5-39 on page 253. Select option 2 (Address space state data, or ASD report) to obtain a report similar to the one shown in Figure 5-41 on page 256.

In the ASD report, the header line contains the same information as in the ARD report. The lines in the body of the report show the value of internal system indicators and fields at the time of the report. Pressing Enter on your terminal produces a new report. The fields are:

- ▶ DMN is the SRM domain number.
- ▶ PG is the SRM performance group number.
- ▶ PP is the SRM performance period number.
- ▶ SRVCLASS is the service class name.
- ▶ SP is the service class period.
- ▶ CL is the current location of the job:
 - IN means in central storage
 - OT means swapped out but ready to run
 - LO means logically swapped out
 - NS means non-swappable
 - WM means waiting for a resource. (swapped in, not nondispatchable, but accumulating no CPU time)
 - WL means a long wait (other than a terminal wait)
 - WT means terminal wait
 - WO means some other wait (not long wait or terminal wait)
 - DL means delayed by SRM
 - PR means privileged
 - << or >> means the job is being swapped into or out of central storage
- ▶ R LS is the reason for the last swap out. A blank means the job in IN, NS, or PR.
 - TI means terminal input wait
 - TO means terminal output wait
 - LW means long wait
 - XS means auxiliary storage shortage
 - RS means central storage shortage
 - DW means detected wait
 - RQ means the job requested the swap
 - NQ means enqueue exchange
 - EX means exchange based on SRM/WLM recommendation value
 - US means unilateral
 - TS means transition swap
 - AW means APPC wait
 - IC means improve central storage use/performance
 - IP means improve system paging rate
 - MR means to make room for a job that has been swapped out too long
- ▶ DP PR is the dispatching priority of the job.
- ▶ CS F is the number of central storage frames (pages) assigned to the job.
- ▶ ESF is the number of expanded storage frames (pages) assigned to the job.
- ▶ CS TAR is the SRM central storage target value.
- ▶ TAR WSS is the target working set size (in number of pages).
- ▶ X M indicates that this is a cross-memory address space, used primarily to provide services for other address spaces.
- ▶ PIN RT is the page-in rate since the last report.
- ▶ ES RT is the rate of movement (in pages per second) to/from expanded storage due to demand paging. It excludes VIO, hiperspace, and shared storage pages.
- ▶ TX SC is the swap count for the job or current transaction.
- ▶ SWAP RV is a WLM-recommended value.
- ▶ WSM RV is a recommended value for working set management, and is not normally used.

You may never need the data in the ASD report, especially the SRM material. However, this report can be useful in determining why a particular job (or transaction) is unusually slow. It can also be useful if you want to tune the SRM parameters in your system. (This is not recommend for new OS/390 owners; also much of this become irrelevant when WLM is used in goal mode.)

RMF - ASD Address Space State Data														Line 1 of 35	
Command ===>														Scroll ===> PAGE	
MIG= 119K CPU= 7 UIC= 254 PR= 0 System= SYS1 Total															
15:01:25	P	P	C	R	DP	CS	ESF	CS	TAR	X	PIN	ES	TX	SWAP	WSM
JOBNAME	DMN	G	P	L	LS	PR	F	TAR	WSS	M	RT	RT	SC	RV	RV
MASTER	0	0	1	NS	FF	760	0		0		----	----	0	0	
PCAUTH	5	1	1	NS	21	54	1		0	X	----	----	0	0	
RASP	5	1	1	NS	FF	235	1		0	X	----	----	0	0	
TRACE	5	1	1	NS	21	60	1		0	X	----	----	0	0	
DUMPSRV	0	0	1	NS	FF	315	1		0		----	----	0	0	
XCFAS	5	1	1	NS	FF	1905	1		0	X	----	----	0	0	
GRS	0	0	1	NS	FF	376	1	32.8K	X		----	----	0	0	
SMXC	5	1	1	NS	FF	5559	1		0		----	----	0	0	
SYSBMAS	5	1	1	NS	28	74	1		0		----	----	0	0	
CONSOLE	5	1	1	NS	FF	332	1		0	X	----	----	0	0	
WLM	5	1	1	NS	FF	930	1		0	X	----	----	0	0	
ANTMAIN	5	1	1	NS	FF	1904	106		0	X	----	----	1	98	
ANTAS000	5	1	1	NS	22	189	479		0	X	----	----	1	88	
OMVS	5	1	1	NS	FF	10.6K	1		0	X	----	----	0	0	
JESXCF	0	0	1	NS	FF	463	36		0	X	----	----	0	998	
ALLOCAS	5	1	1	NS	FF	88	0		0	X	----	----	2	383	
IOSAS	5	1	1	NS	FF	215	1		0	X	----	----	0	0	
MMS	5	1	1	NS	FF	254	1		0	X	----	----	0	0	
IXGLOGR	5	1	1	NS	FF	891	0		0	X	----	----	0	0	
SMF	0	0	1	NS	FF	153	1		0	X	----	----	0	998	
LLA	5	1	1	NS	26	1517	35		0	X	----	----	0	998	
JES2AUX	5	1	1	NS	25	58	1		0		----	----	1	0	
PF 1=HELP	2=SPLIT			3=END			4=RETURN			5=RFIND			6=SORT		
PF 7=UP	8=DOWN			9=SWAP			10=LEFT			11=RIGHT			12=RETRIEVE		

Figure 5-41 RMF - ASD report

A typical display for Address Space SRM Data is shown in Figure 5-42 on page 257. (This display is selected by option 3 in the menu shown in Table 5-1 on page 252.) This report concentrates on *transactions*. A batch job step is a transaction, a TSO command is a transaction, and CICS/IMS transactions are considered transactions here. New fields, not previously described, include:

- ▶ TRANS ACTIVE is the elapsed time since the current transaction became active.
- ▶ TRANS CUR RES is the elapsed time since the current transaction became resident in central storage.
- ▶ TX CT is the transaction count for the job. (As a TSO user, press Enter several times and watch the number increase.)
- ▶ TX SC is the swap count for the current transaction.
- ▶ TX involves several SRM measurements consumed by the transaction;
 - CPU
 - MSO (main storage use)

- IOC (I/O service)
- SRB (SRB time)
- ▶ SESS TOTAL is the total SRM services consumed by the job (not the transaction).

RMF - ASRM Address Space SRM Data											Line 23 of 36	
Command ==>											Scroll ==> PAGE	
MIG= 120K CPU= 5 UIC= 254 PR= 0 System= SYS1 Total												
15:06:27	P	P	TRANS	TRANS	TX	TX	TX	TX	TX	TX	SESS	
JOBNAME	DMN	G	P ACTIVE	CUR RES	CT	SC	CPU	MSO	IOC	SRB	TOTAL	
JES2	5	1	1	21:45 21:45:55	1	0	1790K 20259K	415510	130384	22594K		
VLF	5	1	1	21:45 21:45:57	1	0	16930 149294	600	67	166891		
VTAM	5	1	1	21:44 21:44:06	1	0	182235 2543K	15650	31744	2772K		
DLF	5	1	1	21:45 21:45:58	1	0	1608 7946	690	83	10327		
BPXOINIT	5	1	1	21:43 06:50:05	1	7.8K	23902 9746	40	7	33695		
SMS	5	1	1	21:45 21:45:47	1	0	814003 1181K	239041	6000	2240K		
RACF	0	0	1	21:45 21:45:57	1	0	24791 48607	1975	213	75586		
CATALOG	0	0	1	21:46 21:46:33	1	0	1266K 6163K	745	780	7431K		
TNF	0	0	1	21:44 21:44:03	1	0	107 85	10	0	202		
VMCF	0	0	1	21:44 21:44:00	1	0	5912 4625	35	10	10582		
P390	1	2	1	00:00 00:00:00	106	0	0 0	0	0	612703		
SDSF	5	1	1	21:42 21:42:34	1	0	13698 135611	1010	152	150471		
TCPIP	0	0	1	21:42 21:42:26	1	0	2045K 41049K	4925	1027K	44125K		
NFSC	0	0	1	21:44 21:44:08	1	0	3346 73788	1030	119	78283		

Figure 5-42 RMF - ASRM report

The last three options shown in Table 5-1 on page 252 show the same data as the ARD, ASD, and ASRM reports, but they show it for only a single job. This can be very helpful when monitoring a job that has performance problems. Select option 4 (Address space resource data by jobname) and you should obtain the panel shown in Figure 5-43. Enter the job name of whatever job you want to monitor and press PF3.

RMF Monitor II - Address Space Option - Single Job	
Command ==>	
The jobname entered here applies to ARDJ, ASDJ, and ASRMJ. To exit press END.	
Jobname ==>	httpd1__ Specify a 1 to 8 character jobname.

Figure 5-43 RMF - jobname entry

This should produce the display shown in Figure 5-44 on page 258, but with no data. Press Enter; this should produce the first line of data. Press Enter again, this should produce another line of data, and so forth. The column meanings are the same as in Figure 5-40 on page 253. You can change the jobname P390A, for example, by entering ARDJ P390A in the command line.

Options 5 and 6 (address space state data and address space SRM data) work the same way. Once in these reports, the command to change jobnames is ASDJ jobname or ASRMJ jobname.

RMF - ARDJ Address Space Resource Data														Line 1 of 3			
Command ==>														Scroll ==> PAGE			
MIG= 121K CPU= 7 UIC= 254 PR= 0 System= SYS1 Total																	
P390	DEV	FF	PRV	LSQA	LSQA	X	SRM	TCB	CPU	EXCP	SWAP	LPA	CSA	NVI	V&H		
TIME	CONN	16M	FF	CSF	ESF	M	CR	ABS	TIME	TIME	RATE	RATE	RT	RT	RT		
15:16:17	0.000	0	3	79	0		7K	43.16	44.44	----	----	---	---	---	---		
15:16:38	0.000	0	3	79	0		7K	43.23	44.52	0.24	0.00	0.0	0.0	0.0	0.0		
15:16:39	0.000	0	3	79	0		7K	43.29	44.58	4.00	0.00	0.0	0.0	0.0	0.0		

Figure 5-44 RMF - ARDJ report

5.140 Operating the system

MVS has a strong concept of a system operator (or operators). An operator works through an MVS console. On a typical P/390 or MP3000 installation, this is one 3270 window on the primary PC display. As in any MVS system, it is possible to define and use multiple MVS consoles.

With proper authorization (typically the OPER authority under the TSOAUTH profile in the FACILITY class of RACF), users can issue MVS operator commands from TSO. In practice, this is usually done when viewing the SDSF LOG facility, and the LOG display is used to view the results of the command.

The following text describes several common operator commands and, in some cases, shows typical system output from the command. To make the text clearer, we usually show commands in upper case; in practice, they may be entered in lower case on an MVS console.

5.140.1 What the system is doing

The D A,L command

Probably the most common single command on a smaller OS/390 system is D A,L. It provides a summary of the current jobs in the system. Not all address spaces are listed, although all *started* address spaces and TSO users are listed.

```

00- 15.55.05          d a,l
      15.55.05          IEE114I 15.55.05 2001.170 ACTIVITY 641          C
      JOBS      M/S      TS USERS      SYSAS      INITS      ACTIVE/MAX VTAM      OAS
      00002     00012     00001         00027     00008         00001/00040         00011
      LLA      LLA      LLA      NSW S      JES2      JES2      IEFPROC      NSW S
      VLF      VLF      VLF      NSW S      VTAM      VTAM      VTAM      NSW S
      DLF      DLF      DLF      NSW S      RACF      RACF      RACF      NSW S
      INETD4    STEP1    BPXOINIT OWT AO TSO      TSO      STEP1      OWT S
      SDSF      SDSF      SDSF      NSW S      TCPIP      TCPIP      TCPIP      NSW SO
      PORTMAP   PORTMAP   PMAP      OWT SO      FTPD1     STEP1     FTPD      OWT AO
      HTTPD1    HTTPD1    WEBSRV1   IN  SO      NFSS      NFSS      GFSAMAIN  NSW SO
P390      OWT  0

```

In this example, the statistics indicate:

- ▶ JOBS - the number of address spaces running under initiators
- ▶ M/S - the number of address spaces created by MOUNT or START commands
- ▶ TS USERS - the number of TSO address spaces (users)

- ▶ SYSAS - the number of system address spaces
- ▶ INITS - the number of initiators (both JES and APPC initiators)
- ▶ ACTIVE/MAX VTAM - the current number of TSO users via VTAM and the maximum possible number

Batch jobs and started tasks are then listed, two per line. Consider the INETD4 entry in the example above:

- ▶ The job (or started task) name is INETD4
- ▶ The current step name is STEP1 (or this could be a task identifier for a started task)
- ▶ The step name within a JCL procedure is BPXOINIT (other meanings are possible)
- ▶ The job status is OWT = swapped out, waiting, not ready to run. Other possibilities are:
 - IN - swapped in
 - OUT - swapped out, ready to run
 - OU* - being swapped out
 - IN* - being swapped in
 - NSW - non-swappable

The last two flags indicate the nature of the address space:

- ▶ A - created by another address space (APPC, WLM, OMVS)
- ▶ J - job
- ▶ M - mount command
- ▶ S - started task
- ▶ (blank) - system address space
- ▶ O - dubbed for OMVS functions

In this example, only one TSO user (userid P390) is logged onto the system.

The D A,A command

This command D A,A is not often used because it creates considerable output, but contains information of interest to systems programmers investigating problems. It is most useful if issued from the LOG function of SDSF, because the output can be scrolled on your terminal. An extract of typical output is shown here:

```

      FTPD1   STEP1   FTPD     OWT AO  A=002D  PER=NO  SMC=000
                                PGN=001  DMN=005  AFF=NONE
                                CT=000.293S  ET=00188.23
                                WUID=STC00201  USERID=FTPD
      NFSS    NFSS    GFSAMAIN NSW SO  A=0037  PER=NO  SMC=000
                                PGN=001  DMN=005  AFF=NONE
                                CT=040.589S  ET=00188.23
                                WUID=STC00213  USERID=NFSS

```

The left side has the same meanings as for the D A,L command. The remaining portion shows the address space number (A=), is a PER trap active (PER=), outstanding *step must complete* requests (SMC=), SRM performance group number (PGM=), SRM domain number (DMN=), processor affinity (AFF=), TCB time used (CT=), elapsed time active (ET=), work unit identifier (JES ID)(WUID=), and the associated userid (USERID=).

Several more output fields are possible are documented in the *MVS System Messages* manual. The time data (for CT and ET) can be in four formats:

- ▶ sss.tttS (when the time is less than 1000 seconds)
- ▶ hh.mm.ss (when the time is more than 1000 seconds and less than 100 hours)
- ▶ hhhhh.mm (when the time is over 100 hours)
- ▶ ***** (when the time is over 100000 hours)

The \$DA command

The \$DA command lists the names of jobs running in JES2, including jobs being printed. This command is often used in installations that run considerable numbers of batch jobs and/or do much printing through printers controlled by JES2.

The D R,L command

This command lists any outstanding replies waiting for operator action, together with the associated reply number. MVS normally leaves, on the operator display, any messages requiring replies. However, the operator can remove these messages--usually to make more display space available on the console. Removing the message from the operator display does not remove the requirement to reply to the message. The D R,L command lets the operator see all the messages waiting for replies.

```
00- 15.56.46          d r,l
15.56.46          IEE112I 15.56.46 PENDING REQUESTS 644          C
RM=1   IM=0   CEM=0   EM=0   RU=0   IR=0   AMRF
ID:R/K   T TIME   JOB ID  MESSAGE TEXT
          02 R 08.44.26 STC00085 *02 ISTEXC200 - DYN COMMANDS MAY BE
                               ENTERED
```

In this example, only one message is waiting for a reply. The reply number is 02 and the text of the original message is ISTEXC200 - DYN COMMANDS MAY BE ENTERED. It is normal to have several replies waiting indefinitely, and this example is one of these. The output format of the D R,L command, when multiple replies are outstanding, can be a little confusing. Again, it may be better to use the command under the SDSF LOG function so the output can be scrolled.

5.140.2 Cancel a batch job, started task, or TSO user

Typical cancel commands are:

```
C QSAM13          (Cancel a batch job or started task named QSAM13)
C U=OGDEN         (Cancel TSO session for userid=ogden)
C MYJOB,DUMP      (If job has SYSUDUMP or SYSABEND DD statements and the job owner
                  requested that you "cancel with a dump")
```

5.140.3 Screen control

The K commands

The MVS operator screens can be formatted and controlled in a number of ways. We mention only a few basic topics here.

Consider the screen shown in Figure 5-45 on page 261. This is an AD system after IPL has completed. The operator has just entered a D A,L command, which produced the output at the bottom of the screen. You discover that the output from the D A,L command does not scroll up and vanish off the top; it appears to permanently occupy the bottom of the screen. This is because, by default, the system has a *display area* of 10 lines. Some display commands, such as D A,L, use a display area if it exists. (There is a separate set of commands to manage the display area (K D,F to scroll forward, for one) that we do not discuss.)

You can delete the contents of the display area with the command K E,D. This will restore the "normal" use of the complete screen. The command K A,NONE eliminates the display area; after this, the output from D A,L will scroll up and vanish like other messages. The author always issues a K A,NONE command after IPLing a system. The command K A,10 will redefine a 10-line display area.

```

ACTIVE  USE    NUMBER  VOLUME  DATASET
-----  ---  -----  -
YES     PRIM    1      OS39M1  SYS1.RACFDS
YES     BACK    1      OS39RA  SYS1.RACFDS.BACKUP
ICH15020I RVARV  COMMAND HAS FINISHED PROCESSING.
- 14.59.36      d t
14.59.36      IEE136I LOCAL: TIME=14.59.36 DATE=2001.171 GMT:
00 TIME=19.59.36 DATE=2001.171
- 14.59.44      $da
14.59.44      $HASP612 NO ACTIVE JOBS
- 14.59.49      $dspool
14.59.49      $HASP893 VOLUME(OS39M1) STATUS=ACTIVE,PERCENT=21
14.59.49      $HASP646 21.9230 PERCENT SPOOL UTILIZATION
- 15.00.00      v a81,online
15.00.00      IEE302I OA81    ONLINE
- 15.00.43      v e22,offline
15.00.43      IEE303I OE22    OFFLINE
- 15.00.50      d a,1
IEE114I 15.00.50 2001.171 ACTIVITY FRAME 1 F E SYS=P390
JOBS M/S TS USERS SYSAS INITS ACTIVE/MAX VTAM OAS
0002 00011 00001 00027 00012 00001/00040 00007
LLA LLA LLA NSW S JES2 JES2 IEFPROC NSW S
VLF VLF VLF NSW S VTAM VTAM VTAM NSW S
DLF DLF DLF NSW S RACF RACF RACF NSW S
INETD4 STEP1 BPXOINIT OWT AO TSO TSO STEP1 OWT S
SDSF SDSF SDSF NSW S TCPIP TCPIP TCPIP NSW SO
PORTMAP PORTMAP PMAP OWT SO FTPD1 STEP1 FTPD OWT AO
NFSS NFSS GFSAMAIN NSW SO
IEE612I CN=01 DEVNUM=0700 SYS=P390

IEE163I MODE= RD

```

Figure 5-45 Operator screen

The command K E,1 erases the top line on the screen. This is useful for removing “permanent” messages that waste space on the screen; the DYN COMMANDS MAY BE ENTERED message of the AD system is a good example of a message that can be deleted.

You can use PFKs on the operator console. The command D,PFK displays the current definitions of the PFKs. A command such as K N,PFK=(5,CMD='D U,DASD,ONLINE,A80'),CON=N changes the assignment of a PFK.

5.140.4 Query or control I/O devices

The D U,,,nnn,m command

This command is very common, and displays the high-level operating system state of one or more devices. Using devices starting at address (device number) E20 as an example, the command might be D U,,,E20,3. This will display the state of three devices, starting at address E20. If device E20 does not exist, it will find the device with the closest (higher) address and start there. If no device count is given, the command will attempt to display 16 devices. Undefined addresses (device numbers) are skipped over. Be certain to use the three commas shown.

```

00- 15.57.44      d u,,,e20,3
15.57.44      IEE457I 15.57.44 UNIT STATUS 647      C
UNIT TYPE STATUS      VOLSER      VOLSTATE
0E20 CTC A-BSY

```

```
OE21 CTC A
OE22 CTC OFFLINE
```

In this example:

- ▶ UNIT is the address or device number
- ▶ TYPE is the generic type of the unit, such as CTC (channel-to-channel adapter), 3390, and so forth.
- ▶ STATUS is usually one or more of the following:

A Allocated	BOX Hardware error	BSY Busy	F Offline
M Special mgmt	MTP Mount pending	NRD Not ready	O Online
PND Pending offline		R Reserved (shared DASD)	
S IPL device	SYS System-owned		

The key STATUS indicators are A (allocated, implying the device is being used by a task), F or OFFLINE (meaning the device is not available to the system), O (online, implying a task can allocate and use the device). If the device is a disk drive, and it is online, the volsers and use attributes of the volume are shown. See the D U,DASD,ONLINE command for an example.

The D M=DEV(nnn) command

This is another command showing the status of a device, in a different format. This command displays more information about the channel connections and ESCON switch connection of the device. This command is typically used only by a systems programmer when investigating a device problem.

```
00- 15.58.32          d m=dev(e20)
    15.58.32          IEE174I 15.58.32 DISPLAY M 650          C
    DEVICE OE20      STATUS=ONLINE
    CHP              FC
    DEST LINK ADDRESS  OD
    DEST LOGICAL ADDRESS 10
    PATH ONLINE       Y
    CHP PHYSICALLY ONLINE Y
    PATH OPERATIONAL   Y
    MANAGED           N
```

The D U,DASD,ONLINE command

This is a very common command, primarily because it displays the volsers of disk volumes. The output format is the same as for the D U,,,nnn,m command described earlier.

```
00- 15.59.09          d u,dasd,online
    15.59.09          IEE457I 15.59.09 UNIT STATUS 653
    UNIT TYPE STATUS  VOLSER   VOLSTATE
    OAA0 3390 S       OS39RA  PRIV/RSDNT
    OAA1 3390 A       OS3RAA  PRIV/RSDNT
    OAA2 3390 A       OS39M1  STRG/RSDNT
    OAA3 3390 O       OS39PA  PRIV/RSDNT
```

The VOLSTATE has two parts. The first part is the use attribute and is explained in “How to use OS/390 volume use attributes” on page 129. The second part, RSDNT, is irrelevant on most systems.

By default, the listing will start with the lowest address number online DASD unit and list 16 devices. You can specify a starting address, D U,DASD,ONLINE,AA8, and/or a device count, D U,DASD,ONLINE,300,32. If no devices are available, you will receive the following:

```
00- 16.00.15          d u,dasd,online,AB0
    16.00.15          IEE455I UNIT STATUS NO DEVICES WITH REQUESTED
    ATTRIBUTES
```

This indicates that no online DASD devices at address A80 (or higher) were available.

The D U,,ALLOC,nnn command

This command lists the job names (or TSO userids) that have currently allocated a device. For a single-user device, such as a tape drive, this will be a single job. For shared devices, such as disk drives, this can be a long list. This command is most useful when the system will not let you vary a device offline because some job has allocated it. The command can display multiple devices (and defaults to 16), but it is most often used with a device count of one.

```
00- 16.01.32          d u,,alloc,a80,1
    16.01.32          IEE106I 16.01.32 UNITS ALLOCATED 665          C
    UNIT  JOBNAME  ASID  JOBNAME  ASID  JOBNAME  ASID  JOBNAME  ASID
    OAA0  *SYSTEM  0000  *SYSTEM  0001  XCFAS    0006  OMVS    000E
    OAA0  LLA      0016  JES2     0018  VTAM     001A  CATALOG 0020
    OAA0  P390    0029  TCPIP    002E  PORTMAP  0030  NFSS     0031
    OAA0  NFSC     00FB
```

This example indicates that device A80 (which was a 3390 disk drive) is allocated to the system (the asterisk in *SYSTEM indicates this is not a job named SYSTEM) in two address spaces (0 and 1), allocated to XCFAS in address space 6, and so forth. The address space numbers are in hex

The DS P,nnn command

This is yet another way to display device information. It provides more hardware detail than the previous commands. Its primary advantage is that it displays the state of all the channel paths to the device.

```
00- 16.02.31          ds p,aa0
    16.02.31          IEE459I 16.02.31 DEVSERV PATHS 674          C
    UNIT DTYPE  M CNT VOLSER  CHPID=PATH STATUS
    RTYPE  SSID CFW TC  DFW  PIN  DC-STATE CCA  DDC  ALT  CU-TYPE
    OAA0,33903 ,A,113,0S39RA,FD=+
    IDSK02 FD00                                00 00          3990-2
    ***** SYMBOL DEFINITIONS *****
    A = ALLOCATED          + = PATH AVAILABLE
```

This is sometimes known as the *devserv* command. The output can become quite complex and you should consult the *MVS System Messages* manual for a full description.

VARY nnn,ONLINE and VARY nnn,OFFLINE commands

If a device is offline, it cannot be used by the system.⁴⁷ If a device is online, it is available for use. The initial state of a device, after IPL, can be controlled by IODF parameters (and changed with the HCD utility). You cannot vary a device offline if the device is *allocated* to a user.

Typical VARY commands are:

```
V A86,OFFLINE          (V is an abbreviation for VARY)
V A86,ONLINE
V A84-A85,OFFLINE
V A84,ONLINE
```

⁴⁷ There are a few exceptions. For example, the ICKDSF program used to initialize a disk is normally used with offline drives.

5.140.5 Displaying IPL information

It is very common to want to know how a system was IPLed. Three commands provide basic information: D IPLINFO; D IOS,CONFIG; and D PARMLIB. Typical examples are shown here.

```
00- 16.04.14          d iplinfo
16.04.15          IEE254I 16.04.14 IPLINFO DISPLAY 677      C
SYSTEM IPLED AT 08.43.37 ON 06/19/2001
RELEASE OS/390 02.10.00
USED LOAD00 IN SYS1.IPLPARM ON OAA2
ARCHLVL = 1
IEASYM LIST = 00
IEASYS LIST = 00 (OP)
IODF DEVICE OAA2
IPL DEVICE OAA0 VOLUME OS39RA

00- 16.04.46          d ios,config
16.04.46          IOS506I 16.04.46 I/O CONFIG DATA 680      C
ACTIVE IODF DATA SET = SYS1.IODF01
CONFIGURATION ID = CBIPO          EDT ID = 00

00- 16.05.21          d parmlib
16.05.21          IEE251I 16.05.21 PARMLIB DISPLAY 683      C
PARMLIB DATA SETS SPECIFIED
AT IPL
ENTRY  FLAGS  VOLUME  DATA SET
   1     S    OS39RA  SYS1.ADCD10.PARMLIB
   2     S    OS39M1  SYS1.PARMLIB
```

The core information from these displays is:

- The IPL device was AA0 and the IPLPARM was OAA200 (taken from *IODF device = OAA2* and *LOAD00* was used).
- The LOADxx member used SYS1.IODF01
- The PARMLIBs are SYS1.ADCD10.PARMLIB and SYS1.PARMLIB, in that order.

5.140.6 ENQ/DEQ contention

The D GRS,CONTENTION command

The MVS internal locking mechanism is based on the ENQ and DEQ macros and is often known by this name. The most obvious use of ENQ/DEQ is through JCL parameters such as DISP=OLD (for exclusive use of a data set) or DISP=SHR (for shared use of a data set). However, it is used for many more internal functions within MVS. In general, JCL and batch jobs will take care of themselves, as far as ENQ/DEQ is concerned, but problems can occur.

Consider the following situation:

```
- 13.16.05 JOB00226 $HASP373 P390ZT   STARTED - INIT 2   - CLASS A - SYS
- SYS1
*13.16.05 JOB00226 *IEF099I JOB P390ZT   WAITING FOR DATA SETS
- 13.18.03          d grs,contention
00 13.18.05          ISG343I 13.18.03 GRS STATUS 786      C
S=SYSTEM  SYSDSN  SYS1.SAMPLIB
SYSNAME   JOBNAME      ASID      TCBADDR  EXC/SHR  STATUS
P390     OGDENW        0017     008F6920 EXCLUSIVE OWN
P390     P390ZT        0022     008F6920 SHARE   WAIT
NO LATCH CONTENTION EXISTS
```

Here we see job P390ZT starting, followed by a message that it is waiting for data sets. This normally means that some other job is using a data set that this job needs. It is fairly normal to see such messages. The normal operator response is to simply wait a while to see if the situation clears itself, which should happen when the earlier job ends. If the situation does not clear, the operator can issue the D GRS,CONTENTION command to see what ENQ/DEQ conflicts are present in the system. In this example, we see that job OGDENW owns an *exclusive* ENQ on SYS1.SAMPLIB and job P390ZT has requested a *shared* ENQ on the same data set and is waiting for it.

At this point the operator can wait longer, or call someone about job OGDENW and explain that it has been running for a long time and is blocking another job. In this example, OGDENW was hung (absorbing no CPU time and doing no I/O, as shown by the SDSF DA function) so it was cancelled. Job P390ZT then ran normally.

The D GRS,RES command

A variation of the D GRS command can be used to display any system-wide ENQ/DEQ lock. To use it, you need to understand a bit about ENQ/DEQ. These functions use a two-part name to make a queue or locking entity. The first part, known as the major name, is a maximum of eight characters. The second part, the minor name, is up to 44 characters. At the system-wide level⁴⁸ certain major names are reserved. One of these is SYSDSN. All data set enqueue functions (whether from JCL or dynamic allocation) are translated into an ENQ with major name SYSDSN and the minor name equal to the data set name. We can ask, for example, who has SYS1.LINKLIB allocated.⁴⁹

```
00- 16.06.08          d grs,res=(SYSDSN,SYS1.LINKLIB)
    16.06.08          ISG343I 16.06.08 GRS STATUS 686          C
    S=SYSTEM  SYSDSN  SYS1.LINKLIB
    SYSNAME   JOBNAME   ASID   TCBADDR   EXC/SHR   STATUS
    P390      XCFAS     0006   008FFBF8  SHARE     OWN
    P390      LLA       0016   008FFBF8  SHARE     OWN
```

This says that jobs XCFAS and LLA (which are both system-related started tasks) have SYS1.LINKLIB allocated (locked) at the SHARE level. Both these started tasks probably had DD statements for SYS1.LINKLIB with DISP=SHR.

5.140.7 Started tasks

The S, P, and F commands

For MVS operator commands, S means start, P means stop, and F means modify. (Early OS/360 designers felt compelled to use single-letter abbreviations for common operator commands.) The START command typically names a procedure in SYS1.PROCLIB. An address space is created by the start command and the named procedure is run in the address space.⁵⁰

If programmed to do so, a started task can listen for a STOP and/or MODIFY command. Some of the system-supplied started tasks do this, and some do not. You need specific documentation or instructions about each task, although it does not hurt to try a STOP command (assuming you want to stop the task). If the program is not listening for the command, the system will simply discard it. The format of parameters for the MODIFY command is generally undefined; you need the specific documentation for a program in order to use a MODIFY command (assuming the program listens for it).

Some examples are:

⁴⁸ There are other levels of ENQ/DEQ, not described here.

⁴⁹ Among other meanings, to have a data set *allocated* means to hold an ENQ lock for it/

⁵⁰ There are other variations for started procedures (*started tasks*) and started jobs, but this is the most basic and most common.

S TCPIP	(start a <i>started task</i> named TCPIP)
P TCPIP	(issue a stop command for TCPIP)
S HTTPD1	(start the web server)
F HTTPD1,APPL=-restart	(bounce the web server)
P HTTPD1	(stop the web server)

These are real examples from the R10 and R1.1 AD systems. The TCPIP started task, which starts the system TCP/IP function, listens for a stop command. The HTTPD1 started task listens for modify and stop commands. The specific modify functions include the one shown in the example, with the syntax shown; it causes the Web server to reread its configuration file.

5.140.8 Sending messages to users

The SE command

The operator can send messages to everyone in the system, or to a specific TSO user. For example:

```
SE 'I will IPL in 5 minutes. Please logoff'
SE 'Lunch in 5 minutes?',user=(p390Z)
```

Remember to use single quotes if you use these functions. In general, ISPF users do not like to receive messages because it disrupts their screen session. A TSO user can disable reception of messages, so you cannot assume that everyone sees messages you send.

5.140.9 JES2 commands

There is a large number of JES2 commands. It is possible to almost completely reconfigure JES2 by using operator commands. There are commands to exercise considerable control over printers controlled by JES2, and so forth. We will mention only a few of the JES2 commands to demonstrate their general flavor.

The \$DI command

This command displays the state (ACTIVE, INACTIVE, DRAINED) of all the defined initiators, the job classes handled by the initiator, and the name of the initiator. The DRAINED state means the initiator is not started. The INACTIVE state means it is waiting for a job.

```
00- 16.10.19      $di
    16.10.19      $HASP892 INIT(1)  STATUS=INACTIVE,CLASS=A,NAME=1,
    ASID=0022
    16.10.19      $HASP892 INIT(2)  STATUS=INACTIVE,CLASS=AB,NAME=2,
    ASID=0023
    16.10.19      $HASP892 INIT(3)  STATUS=INACTIVE,CLASS=ABC,NAME=3,
    ASID=0024
    16.10.19      $HASP892 INIT(4)  STATUS=INACTIVE,CLASS=ABCDE,NAME=4,
    ASID=0025
    16.10.19      $HASP892 INIT(5)  STATUS=DRAINED,CLASS=ABCD,NAME=5
    16.10.19      $HASP892 INIT(6)  STATUS=DRAINED,CLASS=ABCD,NAME=6
    16.10.19      $HASP892 INIT(7)  STATUS=DRAINED,CLASS=ABCD,NAME=7
    16.10.19      $HASP892 INIT(8)  STATUS=DRAINED,CLASS=ABCD,NAME=8
    16.10.19      $HASP892 INIT(9)  STATUS=DRAINED,CLASS=ABCD,NAME=9
    16.10.19      $HASP892 INIT(10) STATUS=DRAINED,CLASS=ABCD,NAME=10
    16.10.19      $HASP892 INIT(11) STATUS=DRAINED,CLASS=ABCD,NAME=11
    16.10.19      $HASP892 INIT(12) STATUS=DRAINED,CLASS=ABCD,NAME=12
```


The \$TI command

This command changes the job classes handled by an initiator. In the following example, the initiator named "3" is changed to run job classes X, Y, and Z.

```
00- 16.11.10      $ti3,C=XYZ
    16.11.10      $HASP006 $TI3,C=XYZ   TRANSLATED BY EXIT HASX05C TO
    $TI(3),C=XYZ
    16.11.10      $HASP892 INIT(3)   STATUS=INACTIVE,CLASS=XYZ,NAME=3,
    ASID=0024
```

Initiator names and initial job classes are set in the JES2PARM member of PARMLIB. In the current AD systems initiator names are the same as their internal number in JES2PARM.

The \$SI and \$PI commands

Commands to start and stop (drain) initiators are straightforward. Stopping an initiator will not take effect until the current job using the initiator (if any) completes.

```
$PI3
$SI6
```

The \$DSPOOL command

This command displays the percentage of the JES2 spool space that has been used. JES2 will issue many other messages if it starts to exhaust spool space. However, many operators issue this command now and then to informally monitor spool space. Completely running out of spool space is a bad situation that requires some skill to overcome. (The most common response of new OS/390 owners to simply cold start the system; this erases everything in spool. In a large production system, an unplanned cold start is considered a major disaster.)

```
00- 16.11.59      $dspool
    16.11.59      $HASP893 VOLUME(OS39M1) STATUS=ACTIVE,PERCENT=10
    16.11.59      $HASP646 10.2564 PERCENT SPOOL UTILIZATION
```

The \$A and \$H commands

These commands hold (\$H) and release (\$A) jobs in the JES2 job queue. They are commonly used in large batch-oriented installations and there are many variations of the commands. Simple examples are:

```
$H A              (hold all jobs)
$A               (release all held jobs)
$H 'OGDENW'      (hold the job named OGDENW)
$A 'OGDENW'      (release OGDENW)
```

In these examples, *hold* means to not feed the job to an initiator. If the job is already started (being processed by an initiator), the commands have no effect.

Queue management

The \$DN,ALL command will display all the jobs in JES2 queues. This can be a long list.

The \$D 'jobname' command will display the JES2 status of a job with the indicated name.

A \$P 'jobname' command will purge the indicated job from all JES2 queues. A \$PJnnn command will purge the indicated job (using the JES2 identifier for the job) from all JES2 queues. A JES2 identifier might be J000179; you can omit the leading zeros and enter the command as \$PJ179

5.140.10 SVC dumps

The DUMP command

This command should not be confused with the C jobname,DUMP command, which cancels a job with an *application* dump. The DUMP command produces a *system* or *SVC* dump, and is typically used to debug system-related problems. It is most commonly used at the request of IBM technical assistance groups. In the following example, TSO userid *p390* has done something really unusual and is harming the system. IBM service suggested obtaining an SVC dump with region, trace table, and summary data the next time the problem happened.

```
- 16.15.16          dump comm=(dump user p390 for loop)
| 16.15.16          *03 IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND
- 16.15.57          3,tsoname=p390,sdata=(rgn,trt,sum)
16.15.57           IEE600I REPLY TO 03 IS;TSONAME=P390,SDATA=(RGN,TRT,SUM)
16.15.58 TSU00106 IEA794I SVC DUMP HAS CAPTURED:
DUMPID=001 REQUESTED BY JOB (*MASTER*)
DUMP TITLE=DUMP USER P390 FOR LOOP
00- 16.16.00        IEA611I COMPLETE DUMP ON SYS1.P390.DMP00001
- DUMPID=001 REQUESTED BY JOB (*MASTER*)
- FOR ASID (0029)
- INCIDENT TOKEN: ADCDPL P390 06/19/2001 21:15:57
```

The operator entered an initial DUMP command with an identifying comment. The system asked for more parameters. The reply listed the *tsoname* to dump (this could have been a *jobname*), and the nature of the data to be dumped. Typically, IBM service will help you define this parameter. In this example, the dump was written to a dynamically allocated dump data set named SYS1.P390.DMP00001. See “How to manage dump data sets” on page 192 and “Capturing SVC dumps” on page 247 for more about SVC dumps.

5.140.11 RACF

The #RVARY command

The #RVARY command can list the RACF data base names and can be used to activate or deactivate RACF data bases. This is seldom done by new OS/390 owners and is not illustrated here.

```
00- 16.24.14          #rvary list
16.24.14 STC00092 IRR011I (#) OUTPUT FROM RVARY:
ICH15013I RACF DATABASE STATUS:
ACTIVE  USE    NUMBER  VOLUME    DATASET
----- ---    -
YES     PRIM    1     OS39M1    SYS1.RACFDS
YES     BACK    1     OS39RA    SYS1.RACFDS.BACKUP
ICH15020I RVARY COMMAND HAS FINISHED PROCESSING.
```

All the other RACF commands used throughout this redbook are TSO commands, not MVS commands.

5.140.12 How to reply to console messages

Console messages that require a reply contain a reply number that you must enter as an identifier with your reply. When OS/390 is starting, it wants replies in the format:

```
reply 02,'xxxx'
```

where 02 is the reply number and xxxx is your reply to the message. After the early phases of IPL, you can use shorter forms of reply, such as:

```
2xxxx          (cannot use if JES3 modules are installed)
```

2,xxxx (use with both JES2 and JES3)

The first one, 2xxxx, assumes that xxxx begins with a non-numeric character. If it begins with a numeric character, you need to use a comma to separate the reply number from the text of the reply.

If the reply text contains blanks, you will need single quotes around it.

5.140.13 Common VTAM commands

There are many special VTAM commands that can be issued from the OS/390 console, and an experienced VTAM administrator can often isolate and correct problems using only these commands. A few of the common commands are:

```
D NET,APPLS      /* Display VTAM applications status      */
D NET,BFRUSE     /* Display VTAM buffers in use                          */
D NET,CDRMS      /* Display VTAM cross domains status                    */
D NET,CDRSCS     /* Display VTAM cross domain resources                  */
D NET,COS        /* Display VTAM cos table in use                        */
D NET,GROUPS     /* Display VTAM groups status                           */
D NET,ID=name    /* Display VTAM resource status                         */
D NET,ID=*.name  /* Display VTAM resource status in any net             */
D NET,ID=LINES   /* Display VTAM lines status                            */
D NET,ID=MAJNODES /* Display VTAM major nodes status                     */
D NET,U,ID=userID /* Display VTAM user ID status                          */
Z NET,CANCEL     /* VTAM abnormal termination command                   */
Z NET,QUICK      /* VTAM normal termination command                     */
V NET,ACQ,ID=ncpname /* Acquire an NCP                                     */
V NET,ACQ,ID=pu_name /* Acquire a nonswitched PU                            */
V NET,ACT,ID=name /* Activate VTAM resource                               */
V NET,DRDS,ID=name /* Activate VTAM dynamic reconfiguration               */
V NET,INACT,ID=name /* Inactivate VTAM resource                            */
V NET,LOGON=applname,ID=name /* Forces logon to an application                       */
V NET,REL,ID=ncpname /* Release an acquired NCP                             */
V NET,REL,ID=pu_name /* Release an acquired PU                              */
```




More administration and use

This chapter continues the “how to” sections, in no particular order.

6.1 How to display all PTFs in a system

Using the M option in ISPF, go to the SMP/E primary menu. (The M option is unique to the AD CD-ROM systems; other OS/390 packages usually have something similar.) On the first panel of SMP/E select option 3 (query) and enter 'MVS.GLOBAL.CSI' in the SMPCSI DATA SET field.¹ Press the Enter key. In the next panel (Query Selection Menu), select option 1 (CSI Query). In the next panel (CSI Query) enter SYSMOD in the ENTRY TYPE field. In the next panel (Query Zone Selection) you will see several zones listed. You are interested in *target* zones. Select the most likely name in the list by placing an S before the name. This will list the APARs (names begin with A-K or V-Z) and PTFs (names begin with U) in the selected target zone.

6.2 Prelinking C programs

The C Programmer's Guide discusses the *prelink* step that may be required with C programs. When do I need this? The following tips may help:

- ▶ C code compiled NORENT, NOLONGNAME does not require the prelinker, regardless of the target (PDS, PDSE, HFS). Some of the simple procedures distributed with the system (such as EDCCLG) use NORENT and NOLONGNAME and then direct the load module to a PDS; no prelink is needed for this.
- ▶ For C or C++ code with RENT and/or LONGNAME (the most common situation):
 - If you use the prelinker + any link editor (including the binder), the output can be placed in a PDS.
 - If you use the binder (without a prelink step), the output must be placed in a PDSE or HFS.

6.3 How to change the maximum OE core dump size

The maximum size of a core dump (in Open Edition) is controlled by the MAXCORESIZE() parameter in the BPXPRMxx member in PARMLIB. The default size is 4 MB. Many BPXPRM values may be changed dynamically by the SETOMVS or the SET OMVS operator commands. A permanent change can be made by editing the BPXPRMxx members you use.

6.4 Oedit and su

I log onto TSO and go to the omvs shell. While there, I **su** to a userid with UID=0. I then try to use **oedit** to edit a file. Since I am now UID=0, I should be able to edit any UNIX file. It fails with the following messages in syslog:

```
ICH408I USER(bill) GROUP(sys1) NAME(Bill          )
/u/bill/hello.c
CL(FSOBJ) FID(01C8C6E2F0F0F1000F06000000000003)
INSUFFICIENT AUTHORITY TO OPEN
ACCESS INTENT(-W-) ACCESS ALLOWED(R-X)
```

However, if I log onto TSO with the same UID=0 identity that was the target of my **su**, I can edit the file with no problems. Why?

¹ This is the global CSI name in the AD CD-ROM system. Your name may differ. You can probably find the name used in your system by using the ISPF 3.4 function with Dsnlevel = **.CSI.

Understanding this involves a mixture of UNIX concepts and TSO concepts. In the omvs shell, you can **su** to a different effective userid. However, this affects only a subshell created by **su**. In particular, your TSO identity is not changed. The **oedit** editor runs in your TSO session, even though you start **oedit** from your omvs shell. Thus, **oedit** obeys file permissions that apply to your TSO identity (which is unaffected by **su** commands). If you log onto TSO with another userid (which might have a UID=0, although this is not relevant in this discussion), that userid responds to its own file permissions. In your example, this second userid has *write* permission for the file.

6.5 How to work with oedit and permission bits

The *umask* variable works in Open Edition in the same way as other UNIX systems. If you create a file with the **touch** command, it will have permission bits of 666 minus the bits turned off by *umask*. By default, *umask* is 022. With this, the **touch** command (or almost any other UNIX command that creates a text file) will have permission bits 644.

The **oedit** command runs in the TSO environment and has different rules. If **oedit** is used with an existing file, it will leave the permission bits unchanged.² If you create a new file using the **oedit** command, it will have permission bits 700 and these are not affected by *umask*. If you want different permissions you should (a) use **chmod** after exiting from **oedit**, or (b) use **touch** to create the file before using **oedit**.

6.6 How to create an MVS data set from Open Edition

The following code creates an MVS data set (with a hard-coded name) from a C program. The program could be used in both MVS and OE environments. This example creates a member in a new PDS. It could be used to create a sequential data set by removing the member name and changing the space parameters.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
int main()
{ FILE *stream;
  stream = fopen("///'p390a.lib.new(mem1)''", "w,recfm=FB,lrecl=80,blksize=8000,
    space=(trk,(10,2,10))");
  if (stream == NULL) exit(4);
  fprintf(stream, "This is a record\n");
  fprintf(stream, "This is another record\n");
  fclose(stream)
}
```

6.7 How to specify tape device types for HCD

The following translation table may help:

Common term	HCD term	
3480	3480	(without IDRC)
3480+IDRC	348X	(IDRC = COMPACT feature in HCD)
3490	348X	(same as 3480 with IDRC)
3490E	3490	

² This statement assumes you have permission to edit the file, of course.

6.8 Maximum line length for ISPF edit

By default the maximum line length for the ISPF editor is 255 characters. However, if the data set being edited already contains a line longer than 255, then the maximum is set to the longest line + 25%. Once an edit session has started, the maximum length will not be changed.

This seldom creates problems in traditional TSO use, but it might create a problem in Open Edition. (The ISPF editor is called under the ish shell and by the oedit command under the omvs shell.) Very long lines are more common in a UNIX environment. One suggestion is to write a trivial C program that adds a long line to a target file. When you then edit the target file with ISPF, it will obtain edit buffers that are 25% longer than the long line you created.

6.9 How to change SQA and CSA sizes

These sizes are set in the IEASYSxx members of PARMLIB. In general, the sizes set in the AD system are appropriate and seldom need to be changed. However, we have been told that use of both WebSphere and the Web server can stress several aspects of the distributed AD system. In particular, you may need to add more paging data sets (with perhaps 500 MB total space) (see “How to provide more paging space” on page 187). In addition to paging space, you may need more SQA and CSA space. The following values have been suggested:

```
CSA=(3000,250M),  
SQA=(15,449),
```

Both these parameters are set in IEASYSxx members in PARMLIB. Remember to change all the IEASYSxx members you may use (IEASYSCS, IEASYS00, IEASYSDC, and so forth). Also, be certain to change the correct PARMLIB. The syntax is odd. The suggest CSA value is 3000 1K blocks in low storage (below the 16 MB line) and 250 MB in high storage (above the 16 MB line). The SQA values are in units of 64 KB.

Always edit IEASYSxx members with care and remember the commas at the end of each parameter (except the last).

6.10 How to adjust telnet sessions to USS

There are many telnet client packages in use. In general, you should be able to connect to Unix System Services with any of these. (With the AD systems, OS/390 TCP/IP port 1023 is used for an ASCII telnet connection to USS.) Some of the telnet clients are unable to use **vi** under USS. If this occurs, try the command **export \$TERM=vt100** from the telnet session and then try **vi** again.

6.11 Binder versus linkage editor

The binder, for all practical purposes, has replaced the traditional MVS linkage editor. Perhaps the greatest advantage of the binder is that it removes the need to prelink C and C++ programs.

Also, if the binder detects that the SYSLMOD DD statement points to a PDSE, it will create a program object instead of a load module. A program object is an executable, like a load module, but it has additional advantages. If SYSLMOD points to a PDS, a traditional load module is produced.

The commonly used JCL procedures for various compilers have been updated to use the binder. If you are using the IBM-provided procedures, you will use the binder automatically.

6.12 How to test for z/Architecture

A byte at address 163 (x'A3') contains architectural level information. For z/Architecture bit 7 (the last bit) is one and all the other bits are zero. If this bit is one, you can use 64-bit instructions and other z/Architecture functions.

6.13 How to run a shell script at every IPL

Update `/etc/rc` to call your script. This will cause it to be run during every IPL. See the Unix System Services manual for details about customizing `/etc/rc`. (You may note that the distributed version of `/etc/rc` has many lines commented out. These lines were used in earlier releases of USS and are no longer needed.)

6.14 How to use an ASCII tar file

The `pax` command can process tar files, and provides additional services. If someone sends you a tar file that contains ASCII text material, you might try this command to use the file in EBCDIC-based USS:

```
pax -r -o from=ISO8859-1,to=IBM-1047 -f my.file.tar
```

Use the name of your tar file, of course. This should work for ASCII text files within the tarball, but it will corrupt binary files, of course. However, another `pax` or `tar` (without translation parameters) should recover binary files.

6.15 Exposures of UACC(READ)

What harm is there in making my source, JCL, and test data available (read-only) to other users? If my userid is JOE, I can use the following RACF command to accomplish this:

```
AD 'joe.*' UACC(READ)          (if EGN is turned off. Default in AD systems)
AD 'joe.**' UACC(READ)         (if I have EGN turned on)
```

Read authority allows another user to submit your jobs. The job is submitted under the other person's userid, of course, but you might not want anyone running (or trying to run) your jobs.

Also, other users can browse your data sets. This means that you have a `DISP=SHR` enqueue on the data set(s). In turn, this means you cannot edit the data sets or do any other function that requires a `DISP=OLD` enqueue. In an extreme case, this could prevent you from doing your work.

6.16 How to work with VTAMLST

At IPL time, MVS reads COMMNDxx from PARMLIB. In this member there is a statement to start the VTAM subsystem. This command is processed and, after JES2 starts, VTAM starts its own procedure which reads all parameters and statements it needs to provide its functions. The line in COMMNDxx might look like this:

```
COM='S VTAM,,,(LIST=00)          Automatic START of VTAM'
```

Typically, there is a procedure in SYS1.PROCLIB(VTAM) to start VTAM. It usually looks something like this:

```
//VTAM PROC
//VTAM EXEC PGM=ISTINM01,REGION=0M,
//          DPRTY=(15,15),TIME=1440,PERFORM=8
//VTAMLST DD DSN=SYS1.LOCAL.VTAMLST,DISP=SHR
//          DD DSN=SYS1.VTAMLST,DISP=SHR
//VTAMLIB DD DSN=SYS1.LOCAL.VTAMLIB,DISP=SHR
//          DD DSN=SYS1.VTAMLIB,DISP=SHR
//SISTCLIB DD DSN=SYS1.SISTCLIB,DISP=SHR
//SYSABEND DD SYSOUT=*,HOLD=YES
```

When VTAM is started (as a result of executing the VTAM procedure in SYS1.PROCLIB), VTAM reads member ATCSTR00³ from the libraries associated with the VTAMLST DDname. Older systems typically use only SYS1.VTAMLST. More recent systems, including the AD CD-ROM systems, concatenate SYS1.LOCAL.VTAMLST first. The ATCSTR00 member provides important data to VTAM, such as NETID which indicates the Network ID for this Network, CONFIG which indicates the VTAM configuration member to use, and so forth. Here is the content of ATCSTR00 of SYS1.LOCAL.VTAMLST for the AD system:

```
SSCPID=06,NOPROMPT, X
CONFIG=00,MAXSUBA=31,SUPP=NOSUP, X
SSCPNAME=P390SSCP, X
NETID=P390, X
HOSTPU=P390$PU, X
NODETYPE=NN, X
DYNLU=YES, X
HOSTSA=6, X
CRPLBUF=(208,,15,,1,16), X
IOBUF=(400,508,19,,1,20), X
LFBUF=(104,,0,,1,1), X
LPBUF=(64,,0,,1,1), X
SFBUF=(163,,0,,1,1)
```

Several critical parameters are set by this member; in particular, the subarea and SSCPNAME are set here. If you are integrating your system with an existing SNA network, you will probably need to customize this member.

When editing members in VTAMLST, *be certain you are editing the correct data set*. For example, SYS1.VTAMLST and SYS1.LOCAL.VTAMLST (if it exists) often have duplicate member names, and normal concatenation rules apply. We use the term VTAMLST to refer to all the libraries concatenated to this DDname.

VTAM also reads VTAMLST(ATCCONxx) (usually ATCCON00) and automatically starts all the *major nodes* (members in VTAMLST) that are listed there. A typical ATCCON00 member is:

```
A0600,NSNA70X,NSNA90X,DYNMODEL,A0TCP,P390APP,etc
```

³ Or ATCSTRxx if another LIST member is specified in the **start** command.

During the VTAM start process at IPL time you will receive some messages on the system console, these messages will inform you about the processes being done for VTAM. You will notice that VTAM initialization parameters were executed in order to activate VTAM initial configuration. Here is an example segment of the system log when VTAM is starting:

```

S VTAM,,,(LIST=00)
IEF695I START VTAM WITH JOBNAME VTAM IS ASSIGNED TO USER VTAM
, GROUP SYS1
$HASP373 VTAM STARTED
IEF403I VTAM - STARTED - TIME=09.54.00
IST315I VTAM INTERNAL TRACE ACTIVE - MODE = INT, SIZE = 100 838
IST199I OPTIONS = NONE
IST314I END
IST093I ISTRTPMN ACTIVE
IST984I USER EXIT ISTEUCUV IS ACTIVE
IST093I ISTDWMM ACTIVE
IST984I USER EXIT ISTEUCSD IS ACTIVE
IST093I A0600 ACTIVE
IST984I USER EXIT ISTEUCGR IS ACTIVE
IST093I NSNA70X ACTIVE
IST093I NSNA90X ACTIVE
IST020I VTAM INITIALIZATION COMPLETE FOR V4R3
IST1349I COMPONENT ID IS 5695-11701-301
IST1348I VTAM STARTED AS SUBAREA NODE
IST1214I FFST SUBSYSTEM IS NOT INSTALLED

```

Most IBM-provided VTAM configurations contain member A0600. (There is nothing special about this name. It was selected years ago as part of an example.) This member contains the definitions of many of the local applications, such as TSO and CICS, that use VTAM. This member is usually rather long, and only excerpts are shown here:

```

A0600 VBUILD TYPE=APPL APPLICATION MAJOR NODE
*
*
BNJDSERV APPL AUTH=CNM, REQUIRED FOR NPDA VERSION 2 *
ACBNAME=BNJDSERV APPLID FOR ACB
*
DSILOG APPL AUTH=CNM, REQUIRED FOR NCCF REL 2 *
ACBNAME=DSILOG APPLID FOR ACB
*
* APPL DEFINITION STATEMENTS FOR NETWORK LOGICAL DATA MANAGER
*
AAUTSKLP APPL AUTH=CNM,EAS=5 NLDM REL 1
*
DSIAMLUT APPL AUTH=(ACQ),VPACING=15 NLDM REL 1
*
* APPL DEFINITION STATEMENTS FOR CICS
*
A06CICS APPL EAS=160, EST CONCURRENT SESSIONS *
ACBNAME=CICS, APPLID FOR ACB *
AUTH=(ACQ,BLOCK,PASS) CICS ACQUIRE & PASS TMLS
* CICS REQUEST BLOCKED INPUT
*
A06CICS1 APPL EAS=160, EST CONCURRENT SESSIONS *
ACBNAME=CICS1, APPLID FOR ACB *
AUTH=(ACQ,BLOCK,PASS) CICS ACQUIRE & PASS TMLS
* CICS REQUEST BLOCKED INPUT
*
A06CICS2 APPL EAS=160, EST CONCURRENT SESSIONS *
ACBNAME=CICS2, APPLID FOR ACB *
AUTH=(ACQ,BLOCK,PASS) CICS ACQUIRE & PASS TMLS

```


In the AD system, members NSNA70X and NSNA90X contain VTAM local terminals definitions. *Local* means they appear to OS/390 and VTAM as locally-attached 3270 terminals, such as would be provided by a bus & tag 3174 control unit. Excerpts from NSNA70X are:

```
LCL70X  LBUILD
***** LOCAL NON-SNA TERMINALS 701-71F ****
*
LCL701  LOCAL TERM=3277,CUADDR=701,FEATUR2=(EDATS,MODEL2),      X
        USSTAB=USSN,DLOGMOD=DYNAMIC
*
LCL702  LOCAL TERM=3277,CUADDR=702,FEATUR2=(EDATS,MODEL2),      X
        USSTAB=USSN,DLOGMOD=DYNAMIC
etc
```

Emulated 3270 sessions (using P/390 or MP3000 emulated I/O) typically use the 70x and 90x address ranges.

6.17 How to stop and start VTAM

You normally stop VTAM by first stopping all the applications (TSO, CICS, and so forth) using VTAM, and then issuing the OS/390 command:

```
Z NET,QUICK
```

This stops VTAM in a controlled way and will produce messages such as:

```
Z NET,QUICK
IST097I HALT ACCEPTED
IST133I VTAM TERMINATION IN PROGRESS
IST105I ISTCDRDY NODE NOW INACTIVE
IST105I ISTDILU NODE NOW INACTIVE
IST105I ISTRTPMN NODE NOW INACTIVE
IST105I ISTDWMMN NODE NOW INACTIVE
IST105I LOCAL900 NODE NOW INACTIVE
IST105I LOCAL700 NODE NOW INACTIVE
IST105I AO600 NODE NOW INACTIVE
IST105I ISTGROUP NODE NOW INACTIVE
IST412I VTAM COMMAND PROCESSING TERMINATED
IST984I USER EXIT ISTEUCUV IS INACTIVE
IST984I USER EXIT ISTEUCSD IS INACTIVE
IST984I USER EXIT ISTEUCGR IS INACTIVE
IEF403I VTAM - ENDED - TIME=11.40.00
```

Another way to stop VTAM is to cause it to *crash* all the VTAM applications and stop almost instantly. The command is:

```
Z NET,CANCEL
```

This command will end VTAM without performing normal termination processes for those resources or sessions active at that time. You will receive messages such as:

```
IEF102I VTAM TERMINATION COMPLETE
IEF402I VTAM FAILED IN ADDRESS SPACE xxxx ENDED AT END OF MEMORY XXX
```

and, possibly, various error messages from applications using VTAM.

You can also use MVS commands to terminate VTAM but it is not recommended. Commands from MVS should be used only when VTAM is not responding to VTAM commands. The MVS commands below could be helpful to terminate VTAM when not responding to VTAM termination commands:

```
CANCEL VTAM
      or
FORCE VTAM,ARM
```

6.18 How to define a new VTAM application or terminal

VTAM parameters are complex and you should not try to create new definitions unless you understand VTAM fairly well or have good instructions to follow.

Adding an application to VTAM means changing VTAMLST. For a new application, you could add stanzas to VTAMLST(A0600) or create a new member in VTAMLST. We strongly suggest you add new members and do not change A0600 (except for, perhaps, increasing the number of TSO connections defined). If you create a new member to define a new application, you can stop and start this member (*VTAM major node* or *application node*) independently of other VTAM applications.

You might, for example, create member VTAMLST(TCICS) containing⁴:

```
TCICS  VBUILD TYPE=APPL
*          STATOPT='APPLS CICS  '
HNCICS02 APPL ACBNAME=HNCICS02,          C
          AUTH=(ACQ,PASS)
HNCICSST APPL ACBNAME=HNCICSST,          C
          AUTH=(ACQ,PASS)
```

You can then activate the new member without restarting VTAM:

```
VARY NET,ACT,ID=TCICS      start new major node
VARY NET,INACT,ID=TCICS   stop new major node
```

After it is working correctly, you can update VTAMLST(ATCCON00) to include the new member name; this will cause it to be automatically started when VTAM starts.

New terminals are defined in about the same way as new VTAM applications. We strongly suggest you create a new VTAMLST member for a group of new terminals, rather than add definitions to an existing member. You might add a new member named HNNH0209:

```
HNNH0209 VBUILD TYPE=LOCAL      ** 3820 definition      **
*          STATOPT='SNA 919-01  '
HNNH0919 PU  CUADDR=919,          C
          ISTATUS=ACTIVE,          C
          MODETAB=MTABNEW,          C
          MAXBFRU=15,              C
          PUTYPE=2
*          STATOPT='HNNH0209 3820 PRT5'
HNNP0572 LU  LOCADDR=1,          C
          DLOGMOD=M382010S,          C
          LOGAPPL=HNPSF03
*          STATOPT='HNNH0209 3820 PRT5'
```

You can start and stop it manually by:

⁴ This could be in SYS1.VTAMLST or another library concatenated to it. Recent AD systems concatenate SYS1.LOCAL.VTAMLST before SYS1.VTAMLST.

```
VARY NET,ACT,ID=HNNH0209
VARY NET,INACT,ID=HNNH0209
```

There is another way to define VTAM resources, such as lines, domains, physical units, and logical units by using VTAM Dynamic Reconfiguration. This reduces the number of changes you would need to make to VTAMLST. However, describing the setup for this is beyond the scope of this document.

6.19 How to create a VTAM logo

The VTAM logon is what you see on a 3270 screen before logging into an application. It is created by a module named USSTAB. (There are two modules involved, one for non-SNA terminals and one for SNA terminals. This section describes the non-SNA terminal module.) The module is written in assembly language, and the source code is provided with the AD system. The name of the USSTAB actually used can be specified in VTAMLST in the terminal definitions. For the AD system, look in SYS1.LOCAL.VTAMLST(NSNA70X), for example.

The module should be placed in SYS1.VTAMLIB (or a library concatenated with it), and not VTAMLST. In the AD system, it is placed in SYS1.LOCAL.VTAMLIB, which is concatenated to SYS1.VTAMLIB. The source code is in SYS1.VTAM.SOURCE in members USSN. The JCL to assemble and link it is in member USSNJCL. (The library name has tended to change in various AD releases, so you may need to look for it.) The following is an example that differs somewhat from that distributed in R10:

```
//USSTAB JOB 1,P390,MSGCLASS=X
//*
//* THIS JOB CHANGES THE SCREEN THAT VTAM DISPLAYS AT TERMINALS
//* THAT HAVE NO ACTIVE SESSIONS (THOSE ON WHICH NOBODY IS LOGGED
//* ON TO A VTAM APPLICATION).
//*
//SO EXEC PGM=ASMA90,REGION=1024K,PARM='NODECK,OBJECT'
//SYSPRINT DD SYSOUT=*
//SYSLIB DD DSN=SYS1.MACLIB,DISP=SHR,UNIT=3380
// DD DSN=SYS1.SISTMAC1,DISP=SHR,UNIT=3380
//SYSUT1 DD UNIT=SYSALLDA,SPACE=(CYL,(20,5))
//SYSUT2 DD UNIT=SYSALLDA,SPACE=(CYL,(10,1))
//SYSUT3 DD UNIT=SYSALLDA,SPACE=(CYL,(2,1))
//SYSUT4 DD UNIT=SYSALLDA,SPACE=(CYL,(2,1))
//SYSUT5 DD UNIT=SYSALLDA,SPACE=(CYL,(2,1)),DCB=SYS1.MACLIB,
// DISP=(,PASS)
//SYSLIN DD DSN=&&OBJ(USSTAB),DISP=(,PASS),
// SPACE=(CYL,(1,1,1)),UNIT=SYSALLDA
//SYSIN DD *
*****
*** SAMPLE USSTABLE FOR AD . ***
*****
USSLOCAL USSTAB FORMAT=V3R2
LOGON USSCMD CMD=LOGON,FORMAT=BAL
USSPARM PARM=APPLID,DEFAULT='TSO'
USSPARM PARM=LOGMODE
USSPARM PARM=P1,REP=DATA,DEFAULT=' '
*
L USSCMD CMD=L,REP=LOGON,FORMAT=BAL
USSPARM PARM=APPLID,DEFAULT='TSO'
USSPARM PARM=LOGMODE
USSPARM PARM=P1,REP=DATA
*
TSO USSCMD CMD=TSO,REP=LOGON,FORMAT=BAL
```

```

USSPARM PARM=APPLID,DEFAULT='TSO'
USSPARM PARM=LOGMODE
USSPARM PARM=P1,REP=DATA
*
MSG10 USSMSG MSG=10,BUFFER=MSG10BFR
MSGBEG EQU *
MSG10BFR DC AL2(MSGEND-MSGBEG) *** BUFFER LENGTH
          DC X'F5C7114040' *** ERASE WRITE AND SBA TO ROW 1 COL 1
DC CL80'
DC CL80' W E L C O M E T O
DC CL80'
DC CL80'
DC CL80'
DC CL80' P P P P P / 3 3 3 3 3 3 9 9 9 9 9 9 9 0 0 0 0 0 0 0
DC CL80' P P / 3 9 9 0 0
DC CL80' P P P P P / 3 3 3 3 3 9 9 9 9 9 9 9 0 0
DC CL80' P / 3 9 0 0
DC CL80' P / 3 3 3 3 3 9 0 0 0 0 0 0
DC CL80'
DC CL80' WHITE P/390 -- OS/390 R10
DC CL80'
DC CL80'
DC CL80' THIS SYSTEM IS INTENDED FOR THE OS/390 INTRODUCTION
DC CL80' TO OS/390 ADMINISTRATION WORKSHOP, SPONSORED BY
DC CL80' THE S/390 PID ORGANIZATION AND PRESENTED BY BILL OGDEN
DC CL80'
DC CL80' YOUR FAVORITE MESSAGE CAN GO HERE
DC CL80'
DC CL80'
DC CL80' ENTER L, LOGON, OR TSO TO LOG ON TO TSO/E.
          DC X'115CF0' *** SET BUFFER ADDR TO ROW 24 COL 1
          DC X'1DC8115D611DC8' *** DEFINE INPUT FIELD
* *** START FIELD; SBA TO R24,C50; SF
          DC X'115CF113' *** INSERT CURSOR AT ROW 24 COL 2
MSGEND EQU *
        USSEND
        END
//S1 EXEC PGM=HEWLH096,COND=(0,LT),
// PARM=('SIZE=(1000K,100K),NCAL,XREF,LET,LIST')
//SYSPRINT DD SYSOUT=*
//SYSLMOD DD DSN=SYS1.LOCAL.VTAMLIB,DISP=SHR
//SYSUT1 DD UNIT=SYSALLDA,SPACE=(6160,(230,760))
//OBJECT DD DSN=*.SO.SYSLIN,VOL=REF=*.SO.SYSLIN,DISP=SHR
//*****
//SYSLIN DD *
        INCLUDE OBJECT(USSTAB)
        NAME USSTAB(R)
/*

```

We suggest you do not experiment too much with the USSTAB unless you can quickly restore a backup copy of the volume containing SYS1.LOCAL.VTAMLIB (or the equivalent, if you are not using the AD system). A serious error may mean that you cannot log onto TSO.

6.20 How to connect a simple SDLC 3174 to a P/390

Configuring an SNA SDLC connection is somewhat complex, and the background information provided here may make the process more meaningful. There are three major P/390 device managers associated with SNA connectivity:

- ▶ WAN3172 provides SNA/SLDC communication in a context that can be used by OS/390 VTAM. This device manager is intended for use only by OS/390 (MVS) systems.
- ▶ LAN3172 provides SNA communication over a LAN, and replaces AWS3172. If your configuration uses AWS3172, this will automatically be mapped to LAN3172. LAN3172 provides a number of functional upgrades and fixes relative to AWS3172, and can be used with any S/390 operating system.
- ▶ MGR3172 provides an interface to NetView for monitoring LAN3172 and WAN3172 operation. It is used only with NetView.

These three device managers use a large amount of common code, and much of this code is also used in “real” IBM 3172 communication controllers. (This accounts for many messages and panels that say *3172* instead of *WAN3172* or *LAN3172*.) This common code also provides a performance and operation monitor that runs under OS/2, and is completely independent from NetView. The common code is generally known as SNACOM, and this name is seen in some of the messages, documentation, and file names. An older device manager, AWS3172, is no longer used.

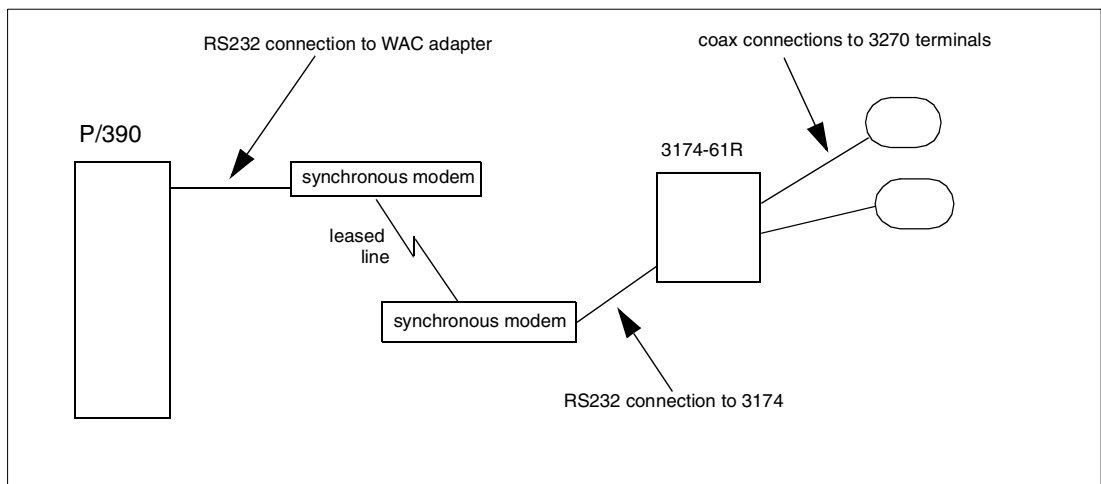


Figure 6-1 Simple SDLC Configuration. Note that synchronous modems are needed; typical PC modems are asynchronous and cannot be used for a basic SDLC connection. The 3174 is a widely-used control unit for 3270-family terminals. The 3174-61R is a remote version, used with up to eight terminals. Our unit was at microcode level C2.0.

WAN3172 uses an interesting technique. From the VTAM side, it appears as a 3172 (and is generated as a 3088 CTC, just like a real 3172). One of the functions of a 3172 is to provide SNA communications through LANs. Generally speaking, WAN3172 appears to MVS VTAM as SNA LAN connections. The WAN3172 code converts the SNA LAN functions to SNA SDLC communications. VTAM is not aware that it is using SDLC connections. In practice, this makes no difference to OS/390 applications, although it does call for somewhat different VTAMLST parameters than would be used with real SDLC connections.

WAN3172 can use two different hardware adapters:

- ▶ Wide Area Connector (WAC) adapters. These adapters provide two lines with a variety of connection options and speeds. For our evaluation, we used RS-232 connections with 19,200 bps modems. Multiple WAC adapters can be used, limited by the availability of adapter slots in the PC Server System/390.
- ▶ Portmaster adapters, providing up to eight lines per adapter, with a limit of two adapters. Two adapters provides 16 SDLC lines, and this is the effective maximum capacity for WAN3172. (For more lines, a 3745 would be used.)

Installing the P/390 device managers

All the files and setup required for WAN3172 are not automatically installed when the P/390 support programs are installed. Stop the P/390 subsystem, go to your P390 directory, and issue the command `inst3172`. After this completes, edit `CONFIG.SYS` and remove the "rem" from the `DEVICE` statement associated with WAN3172 options. You then need to reboot OS/2 to make these changes effective.

The following description uses a P/390 for its examples. Usage on an MP3000 is almost the same. *S/390 Multiprise 3000 Enterprise Server: Emulated Input/Output Subsystem User's Guide and Reference*, GC38-0410, provides an exact description for the MP3000.

Configuring the device managers

IBM PC Server 500 System/390 Installation, Configuration, and User's Guide for MVS/ESA, SA22-7210, provides information about device manager setup, and the `DOC` files of the P/390 (after WAN3172 installation is complete) provide additional details. Neither source can address all the variations possible in a complex communications environment. In this case, a communications specialist is needed, just as with a larger VTAM system. As a starting point, we will describe the parameters needed to use a single IBM 3174 with an SDLC connection to a single port on a WAC adapter. Our 3174 has several 3270 displays attached, and will be used for TSO terminals. Many of the values in the following example are arbitrary, and your installation may be different.

When the WAN3172 function is installed, a new device manager appears in the *Update System Devices* panel of the P/390 configurator. On our system, WAN3174 was device manager C. (This may be different in your system.) You must assign this device manager to a suitable OS390 address. We used address E20, which was already generated as a 3088 in our OS/390 system.

```
>E40 >3088 > > > >C >
```

When WAN3172 is installed, a new function is added to the P/390 configuration panels. This uses F11 to *Configure 3172 SDLC Gateway*. Press F11 to obtain a title panel. Clicking *Configure SNA* on this panel produces the panel shown in Figure 6-2. In this panel, you must select a port in the lower window. With our single WAC adapter with a single line, there was only one option in this window. After selecting it, click **Allocate**.

The screenshot shows a window titled "SNA Configurator". It is divided into two main sections. The top section is titled "Allocated SDLC Ports" and contains an empty rectangular box. To the right of this box are two buttons: "Edit" and "Delete". The bottom section is titled "Available SDLC Ports" and contains a rectangular box with the text "08 WAC_RS232" on the left and "0" on the right. To the right of this box is a button labeled "Allocate". At the bottom of the window, there are four buttons: "Exit", "Exit+Save", "Cancel", and "Help".

Figure 6-2 Basic SDLC configuration panel

This should move the selected port to the upper window and produce the display shown in Figure 6-3.

SDLC Port Edit			
XCP Port	3172 Slot	Port Type	Port number
8	04	Multiport RS232	0
Role:	PRI	SEC	NEG
Dial:	IN	NO	
Datamode:	HALF	FULL	
NRZI:	YES	NO	
Maximum Link Stations:	5		
Maximum Data Size:	521		
<input type="button" value="Set EIA Parameters"/>		<input type="button" value="Set Poll Parameters"/>	
<input type="button" value="OK"/>		<input type="button" value="Cancel"/>	<input type="button" value="Help"/>

Figure 6-3 SDLC Port Edit window

For our purposes, we selected the following options:

- Role: PRIMARY
- Dial: NO
- DATMODE: HALF
- NRZI: YES
- Maximum Link Stations: 1
- Maximum Data Size: 521

After setting these parameters, click **Set EIA Parms**. This should produce the panel shown in Figure 6-4.

EIA Parameters - WAC	
Clocking:	DTE
Line Speed:	9600
RTS:	CONSTANT
Frame Separation:	1
<input type="button" value="OK"/> <input type="button" value="Cancel"/> <input type="button" value="Help"/>	

Figure 6-4 EIA Parameters window

In this panel, set:

- Clocking: DTE
- Line Speed: 9600 (or whatever is appropriate for your modems)
- RTS: CONSTANT
- Frame Separation: 1

Click **OK** to return to the previous panel, and then click **Set Poll Parms** to obtain the display shown in Figure 6-5 on page 286.

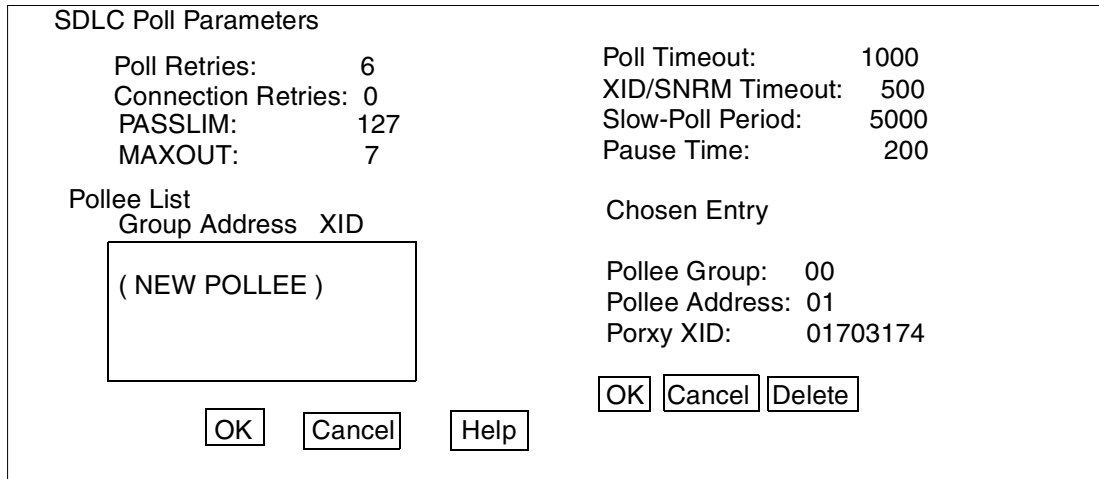


Figure 6-5 SDLC Poll Parameters window

. In this panel, set:

- Poll Retries: 6
- PASSLIN: 127
- MAXOUT: 7
- Poll Timeout: 750
- XID/SNRM Timeout: 500
- Slow-Poll Period: 5000
- Pause Time: 200

Then double-click **NEW POLLEE**. This activates the Chosen Entry section. In this section, set:

- Pollee Group: 00
- Pollee Address: C4
- Proxy XID: 01703174

Click **OK** and then click **OK** again to exit from this panel.

Click **OK** to exit from the previous panel, and click **Save and Exit** to exit from the first panel, and finally **Exit** to exit from the SNA configurator. It will probably tell you that you need to reboot to activate changes.

The Missing Interrupt Handler must be disabled for the address used for the emulated 3172. We created member SYS1.PARMLIB(IOSIEC00) with one statement in it:

```
MIH TIME=00:00,DEV=E40
```

and modified SYS1.PARMLIB(IEASYS00) to contain the line

```
IOS=00,
```

that is needed to use the IOSIEC00 member.

We added the following members to SYS1.VTAMLST:

```
:                MEMBER P390A

                VBUILD TYPE=XCA
WAC11  PORT  ADAPNO=0,CUADDR=E40,MEDIUM=RING,                X
                SAPADDR=04,TIMER=60
WAC11G  GROUP DIAL=YES,DYNPU=YES
```

```

WAC11L  LINE  ANSWER=ON,CALL=INOUT,ISTATUS=ACTIVE
WAC11P  PU    ISTATUS=ACTIVE

                MEMBER P390B

SWNMPT  VBUILD TYPE=SWNET
SLP16   PU    ADDR=C4,MAXDATA=265,PUTYPE=2,ISTATUS=ACTIVE,      X
                MODETAB=ISTINCLM,DLOGMOD=SNX32703,              X
                USSTAB=USSSNA,SSCPFM=USSSCS,                    X
                IDBLK=017,IDNUM=03174
SLP1     LU    LOCADDR=2,ISTATUS=ACTIVE
SLP2     LU    LOCADDR=3,ISTATUS=ACTIVE
SLP3     LU    LOCADDR=4,ISTATUS=ACTIVE
SLP4     LU    LOCADDR=5,ISTATUS=ACTIVE

```

At this point, we stopped OS/390 and OS/2, and rebooted everything to ensure we activated all changes.

You should test your entries by manually activating these VTAM members; later you can add their names to VTAMLST(ATCCON00) so that they will be started automatically. Start the SWNET member first, with the command:

```
V NET,ACT,ID=P390B
```

If this starts cleanly, then:

```
V NET,ACT,ID=P390A
```

At this point, you should observe activity in the modem lights (if you have any) and the logon screen should appear on displays attached to the 3174--*if your OS/390 system has a USSTAB for SNA connections*. Our OS/390 did not have this USSTAB, and usage was more complex:

- ▶ No logo screen is displayed. The keyboard may or may not unlock when the line is started.
- ▶ You must use the SYS REQ function to prepare the terminal for VTAM communications. This usually involves resetting the terminal (to unlock the keyboard) and pressing the SYS REQ key (which is often the ALT-Attn combination). On many 3270-type terminals (and emulators) this should change one of the indicators at the bottom of the display to "a little man in a box." (This is not a very technical description, but it is the most effective description the author has heard.)
- ▶ At this point, enter LOGON APPLID(TSO) and press ENTER. Use uppercase letters. The system should respond with ENTER USERID -, and you would enter a TSO userid and continue normal operation.

6.20.1 How to use the P/390 SNA monitor

This monitor is started with the OS/2 command:

```
C:\SNACOM\LSAMON
```

assuming you used the standard directory names. The panels are too complex to clearly illustrate here. You can display various data rates (frames/second, bytes/second, and so forth), and monitor the activity of any SDLC line or SNA LAN adapter. The monitor is effective for both LAN3174 and WAN3174 device managers. It will not monitor functions not associated with these device managers.

When started, the primary monitor screen displays sub windows for LAN, SDLC (two windows, for the maximum of 16 SDLC lines), and a subchannel window (for the channel between the emulated 3172 and MVS). To see more detail, double-click on the dark bar in a sub window. To remove the detail display, click in the open area at the bottom of the monitor window.

This activity monitor is not related to NetView or any MVS functions. It exists solely as an OS/2 program that inspects samples data being processed by LAN3172 and WAN3172 device managers.

6.21 How to set up NJE

Note: The setup described in this section was tested under MVS 5.2.2. We have not tested it since, but expect that it will work on later releases of OS/390.

This setup will enable two P/390 OS/390 systems, attached to a common token ring, to communicate via NJE. The system names used in this example are RAID and P390. The P390 system uses the default VTAM values from the AD CD-ROM while the RAID system is modified as required. The RAID and P390 names used here are completely arbitrary, and are used only to distinguish between the two systems.

- ▶ Work with LAPS or MPTS in an OS/2 window. The goal is to ensure that the LAN adapter is configured for IEEE 802.2 protocols. Perform the same setup on both P/390 systems. A typical session would be:

```
mpts
CONFIGURE (i.e., select the "Configure" option)
select LAN adapters and protocols
CONFIGURE
inspect the current configuration in the "bottom" window
on the page shown, It should look something like this:
```

```
-----
| IBM Streamer Family Adapter (IBM MPC.OS2)
|   0 - IBM IEEE 802.2
|   0 - IBM OS/2 NETBIOS
-----
```

or

```
-----
| IBM Token-Ring Network Adapter
|   0 - IBM IEEE 802.2
|   0 - IBM OS/2 NETBIOS
-----
```

Note: Click on any of the lines in this window to set low-level parameters.
NETBIOS is not required for NJE, but may be used by LAN3174 or AWS3088.

- ▶ Determine the MAC address of the adapters on both systems. You can use either hardware or locally-administered MAC addresses. One way to find the MAC addresses is to issue (in an OS/2 window): TYPE \IBMCOM\LANTRAN.LOG

if you have booted with the LAN operational. In our example,

MAC address of token ring adapter on RAID = 08005A0D2CFC

MAC address of token ring adapter on P390 = 08005A81ED08

You will need the MAC addresses of both systems later, when defining VTAM parameters.

- ▶ Re-boot OS/2. OS/2 console messages should indicate that LAN connection is effective.
- ▶ On both systems, go to the P/390 Configuration window. We need to define a 3088 address to be used by the emulated SNA 3172. You should select an address that is already known to MVS as a 3088. (If this is not possible, you may need to use HCD to add an MVS address.) In our case, we used address E42 on both systems. Assign this address to P/390 device manager LAN3172. (No CONFIG.SYS changes are involved.)
- ▶ IPL MVS. If there are error messages associated with LAN3172 or the LAN setup, these must be resolved before continuing. Use the MVS console command `D U,,,E42,1` to ensure the address is recognized, and `VARY E42,ONLINE` to verify that it is functional.

You must disable MIH functions for the address chosen for 3172 emulation. Update `SYS1.PARMLIB(IECIOS00)` with something like this:

```
MIH TIME=00:00,DEV=(E40-E47)
```

Verify that `SYS1.PARMLIB(IEASYS00)` has an `IOS=00`, operand. (Remember the comma!) The parmlib changes will not be effective until the next IPL. You can make them effective immediately with the MVS command:

```
SETIOS MIH,DEV(/E42),TIME=00:00
```

- ▶ Edit a number of VTAMLST members, as shown in the next several steps. The RAID system will be subarea 5 and the P390 system will be subarea 6.

Note that VTAM is very particular about formats and columns. Continuation marks must be in column 72 and continued lines must begin in column 16.

The PATH member defines the route to the other system.

```
RAID: VTAMLST(PATH):
```

```
PATH DESTSA=6,ERO=(6,1),VRO=0
```

```
P390: VTAMLST(PATH):
```

```
PATH DESTSA=5,ERO=(5,1),VRO=0
```

- ▶ The CDRM member defines a VTAM to VTAM connection.

```
P390 and RAID: VTAMLST(CDRM): (both systems exactly alike)
```

```
          VBUILD TYPE=CDRM
USS3270  CDRM  SUBAREA=6, ISTATUS=ACTIVE, CDRSC=OPT, CDRDYN=YES,   X
          RECOVERY=YES
RAID390  CDRM  SUBAREA=5, ISTATUS=ACTIVE, CDRSC=OPT, CDRDYN=YES,   X
          RECOVERY=YES
```

- ▶ The AD CD-ROM system has a VTAMLST member (A0600) that defines all the applications that use direct VTAM access. The distributed member contains an entry for JES2, but this entry is not complete. This step will provide different VTAM resource names for the two systems and complete the JES2 definition.

You cannot start TSO if there is an error in the VTAM member containing the TSO application definitions. Before modifying A0600, we suggest you copy it to a new member named A0600BU. If A0600 fails to start, you can manually activate A0600BU with the command `V NET,ACT,ID=A0600BU`.

```
RAID: VTAMLST(A0500)
```

copy member A0600 to new member A0500. Edit A0500, and change

every name field containing "06" to "05". Note that many lines in the member are commented out, and there is no need to change these comment lines. In general, the "06" refers to subarea 06, as used by default in the CD-ROM systems. Our RAID system will use subarea 05, and there could be confusion of VTAM resource names if both systems have the same names containing "06". You have not changed the original A0600 member, so there is no need to provide a backup copy. The use of "06" and "05" is arbitrary, but serves to clearly indicate which subarea owns a given application.

RAID: VTAMLST(A0500)

find the stanza labeled A05JES2 (which was A06JES2 before you changed all the names). Change it as follows:

```
A05JES2  APPL  EAS=1,ACBNAME=JES2,AUTH=(ACQ,VPACE),      X
          VPACING=0,DLOGMOD=RJE379A,MODETAB=MODETAB1
```

P390: VTAMLST(A0600) (remember to backup to A0600BU first)

find the stanza labeled A06JES2.

```
A06JES2  APPL  EAS=1,ACBNAME=JES2,AUTH=(ACQ,VPACE),      X
          VPACING=0,DLOGMOD=RJE379A,MODETAB=MODETAB1
```

- The TYPE=XCA member defines the LAN station (the emulated 3172) to VTAM, and defines connections (LINEs and PUs) using the 3172.

RAID: VTAMLST(A05LAN)

```
A053172  VBUILD TYPE=XCA
SLPORT   PORT  ADAPNO=0,CUADDR=E42,MEDIUM=RING,SAPADDR=4,TIMER=15
SLGRP1   GROUP DIAL=NO
SLL1     LINE  USER=SNA,ISTATUS=ACTIVE
SLP1     PU    ISTATUS=ACTIVE,MACADDR=0800581ED08,          x
          PUTYPE=5,SAPADDR=4,SUBAREA=6,TGN=1
```

(note that the MAC address and subarea are that of the other system)
(for Ethernet, use MEDIUM=CSMACD. Continue to use SAPADDRs)

P390: VTAMLST(A06LAN)

```
A063172  VBUILD TYPE=XCA
SLPORT   PORT  ADAPNO=0,CUADDR=E42,MEDIUM=RING,SAPADDR=4,TIMER=15
SLGRP1   GROUP DIAL=NO
SLL1     LINE  USER=SNA,ISTATUS=ACTIVE
SLP1     PU    ISTATUS=ACTIVE,MACADDR=08005A0D2CFC,          x
          PUTYPE=5,SAPADDR=4,SUBAREA=5,TGN=1
```

(note that the MAC address and subarea are that of the other system)
(for Ethernet, use MEDIUM=CSMACD. Continue to use SAPADDRs)

- The ATCSTRxx member contains global VTAM parameters for a system. It is where the local subarea and SSCPNAME is defined.

RAID: VTAMLST(ATCSTR00)

(edit existing member to make indicated changes)


```
SSCPID=05,...
SSCPNAME=RAID390,
HOSTSA=5,
```

The whole member was:

```
SSCPID=05,NOPROMPT,CONFIG=00,MAXSUBA=31,SUPP=NOSUP,
SSCPNAME=RAID390,NETID=P390,HOSTSA=5,NODETYPE=NN,
DYNLU=YES,CRPLBUFF=(208,,15,,1,16),IOBUF=(100,508,19,,1,20),
LFBUF=(104,,0,,1,1),LPBUF=(64,,0,,1,1),SFBUF=(163,,0,,1,1)
```

P390: VTAMLST(ATCSTRO0)

(no changes required)

The whole member was:

```
SSCPID=06,NOPROMPT,CONFIG=00,MAXSUBA=31,SUPP=NOSUP,
SSCPNAME=USS3270,NETID=P390,HOSTSA=6,
CRPLBUFF=(208,,15,,1,16),IOBUF=(100,508,19,,1,20),
LFBUF=(104,,0,,1,1),LPBUF=(64,,0,,1,1),SFBUF=(163,,0,,1,1)
```

- ▶ The ATCCONxx member defines which VTAMLST members should be activated automatically when VTAM starts.

RAID: VTAMLST(ATCCON00)

add A05LAN,PATH,CDRM to the list

P390: VTAMLST(ATCCON00)

add A06LAN,PATH,CDRM to the list

- ▶ Verify that all these VTAM changes work. You must restart the RAID VTAM, since you changed the subarea from 6 (the default for the CD-ROM systems) to 5. You can activate and deactivate various VTAMLST members individually, using appropriate MVS console commands. Useful commands are:

```
v net,act,id=PATH
v net,act,id=CDRM
v net,act,id=A05LAN or A06LAN
v net,inact,id=xxxx
z net,cancel (use with care! Will kill TSO too)
d net,majnodes
d net,stations
d net,e,id=xxxx for more detail
```

- ▶ On both, copy PARMLIB(JES2PARAM) to PARMLIB(JES2BACK) if appropriate. Verify that SYS1.PROCLIB(JES2) has an ALTPARM DD statement that uses the JES2BACK member. Be certain you know how to start JES2 with alternate parameter input. (See the JES2 Commands Reference manual.) The objective is to ensure that you can fallback to a functional JES2.
- ▶ Edit SYS1.PARMLIB(JES2PARAM) to include the following lines. We placed these lines near the beginning, where there is a comment about NJE.

RAID: PARMLIB(JES2PARAM)

```
NJEDEF LINENUM=1,NODENUM=2,OWNNODE=2
LOGON1 APPLID=JES2
APPL(A06JES2) NODE=1
APPL(A05JES2) NODE=2
NODE(1) NAME=A06JES2
NODE(2) NAME=A05JES2
```

```
CONNECT NODEA=1,NODEB=2
LINE(1) UNIT=SNA
```

```
P390: PARMLIB(JES2PARM)
```

```
NJEDEF LINENUM=1,NODENUM=2,OWNNODE=1
LOGON1 APPLID=JES2
APPL(A06JES2) NODE=1
APPL(A05JES2) NODE=2
NODE(1) NAME=A06JES2
NODE(2) NAME=A05JES2
CONNECT NODEA=2,NODEB=1
LINE(1) UNIT=SNA
```

- ▶ Restart JES2 on both systems. It may be easier to re-IPL MVS to do this. You must cold start JES2 because NJE functions were added. Instead of the normal "1,NOREQ" reply for JES2, use "1,COLD" and reply to the messages asking if you really want to do this. After JES2 formats the spool area, it will ask for more commands; reply "\$S" at this point.
- ▶ Try the commands:

```
$SLOGON1
$SLINE1
$SN,A=A06JES2    or  A05JES2
```
- ▶ On the RAID system, try XMIT A06JES2.P390 to send a message to userid P390 on the other system.

6.22 TCP/IP

This section addresses TCP/IP setup and customization within the context of the OS/390 AD CD-ROM V2R10 systems. This chapter is only intended to assist with initial TCP/IP setup and customization in the context of the AD CD-ROMs and the named platforms. There is a much more complete discussion of OS/390 TCP/IP in the IBM Redbooks:

- ▶ *OS/390 eNetwork Communications Server V2R5 TCP/IP Implementation Guide Volume 1 - Configuration and Routing, SG24-5227*
- ▶ *OS/390 eNetwork Communications Server V2R5 TCP/IP Volume 2 - OpenEdition Applications, SG24-5228*
- ▶ *OS/390 eNetwork Communications Server V2R5 TCP/IP Volume 3 - MVS Applications, SG24-5229*

Both P/390-based systems and MP3000 systems have a PC Server that performs emulated I/O. Adding OS/390 TCP/IP connections involves adding LAN adapters to the underlying server, adding and customizing whatever device drivers are needed by the server, defining OS/390 connections that use these server facilities, and then completing the necessary OS/390 TCP/IP customization. None of these steps is overly complex, but the total combination can be confusing.

The underlying servers also can have TCP/IP functions. A P/390 or an MP3000, for example, can have OS/2 TCP/IP and OS/390 TCP/IP both operational. These are separate TCP/IP implementations that may be connected to the same LAN. We will discuss setup for both the server TCP/IP and OS/390 TCP/IP. We have found that describing only the portions relevant to OS/390 TCP/IP, while ignoring the server TCP/IP, leads to more confusion.

We separate TCP/IP setup into several parts:

- ▶ Understanding and selecting IP addresses.

- ▶ Installing hardware, software, and customization of the underlying server. The details are different for each of the hardware platforms involved.
- ▶ Setup for TCP/IP on the server (OS/2).
- ▶ Basic OS/390 TCP/IP setup. This is common for all platforms, and is limited to making TN3270 and ftp operational. (Basic utility functions, such as **ping** and **netstat** are also included.)
- ▶ A discussion of additional customization for OS/390 TCP/IP, presented at an overview level.

6.22.1 How to assign IP addresses

Explaining the general scheme of IP addressing is beyond the scope of this document. Fortunately, there are many TCP/IP publications (at almost any large bookstore) that describe IP addressing. Fundamental points include:

- ▶ Using systems based on OS/2 (P/390 series, MP3000 emulated I/O), you cannot share a LAN adapter between OS/2 TCP/IP and any S/390 TCP/IP. Furthermore, on an MP3000, you cannot share a LAN adapter among multiple TCP/IP stacks in multiple LPARs.
- ▶ You need to assign an IP address to every adapter/LAN interface that is used for TCP/IP. For example, the P/390 installation described below has two TCP/IP interfaces: one for OS/2 TCP/IP and one for OS/390 TCP/IP. It will require two IP addresses. An Integrated Server or MP3000 installation will be similar.
- ▶ Where there is more than one interface (typical P/390, Integrated Server, MP3000), we will assume all interfaces are connected to the same LAN, with different IP addresses of course.
- ▶ If your LAN is totally isolated -- not connected to other LANs or to the Internet -- you can select any IP addresses you wish. You still need to use consistent addresses. The "net" portion of the address (including the effects of a subnet mask) should be the same for each interface connected to the network, and the "host" portion of the address must be different for each interface.
- ▶ If your LAN is connected to other LANs (or already has a defined addressing scheme), you must coordinate with the person managing the LAN to obtain IP addresses. For our basic P/390 examples, you would ask for two addresses on the same network.
- ▶ Do not use the IP addresses from examples in this document unless you have a totally isolated network. (The 9 address range used in the examples is assigned to IBM. Even within the IBM address range, the numbers used in our examples are *uncoordinated*.)

Note that the following ranges of IP addresses are reserved for private networks:

10.0.0.0	-	10.255.255.255	(The whole class A address)
172.16.0.0	-	172.31.255.255	
192.168.0.0	-	192.168.255.255	

These addresses are useful for several reasons:

- ▶ Documentation examples might use these addresses to avoid using *real* IP addresses in examples.
- ▶ You can use these addresses for your local, private network. If your network is somehow connected to the Internet (by a dialup to an internet provider, for example), these addresses cannot be used through Internet routers -- thus protecting your private network.
- ▶ Most routers are configured to ignore (not forward) these addresses.

6.22.2 How to provide basic OS/390 TCP/IP customization

The setup for an underlying server (P/390, IBM Integrated Server, MP3000) that will run OS/390 and OS/390 TCP/IP and customization, as discussed earlier, is the same, regardless of the specific services you want from OS/390 TCP/IP. OS/390 TCP/IP customization depends on your requirements. For the following descriptions, we will assume our goal is:

- ▶ Configure a single interface, with static routing.
- ▶ TN3270 connections to traditional MVS are needed.
- ▶ telnet (ASCII mode) connections to Open Edition (Unix System Services) are needed.
- ▶ ftp connections for both traditional MVS and Open Edition files are needed.
- ▶ The IBM Web Server will be used later.
- ▶ nfs is not used; neither client nor server.
- ▶ A Domain Name Server is not used. (That is, examples, tests, and initial usage will use absolute IP addresses.)

This is a fairly restricted goal, with static routes, no DNS, and no nfs services. It is, however, a useful goal for initial OS/390 TCP/IP customization. Functions beyond these tend to be highly customized for individual situations. If you can produce a stable system, running the functions listed here, then you have a solid base for your own implementation of additional functions.

Missing Interrupt Handler (MIH)

The OS/390 Missing Interrupt Handler (MIH) for the addresses associated with the emulated 3172 used by TCP/IP must be disabled. This is done automatically by recent releases of OS/390, and you do not need to take any actions in this area.

If you are using an older release of OS/390 (earlier than V2R6) or MVS, you need to disable the MIH. This is done by adding member IECIOSxx to PARMLIB. (The member name used is normally IECIOS00, but can have any suffix; the name follows normal PARMLIB rules.) If the addresses are E20 and E21, then the PARMLIB member might contain one line:

```
MIH TIME=00:00,DEV=(E20-E21)
```

You should verify that this member is active in your PARMLIB. This member (and the associated IEASYSxx pointer) are usually already installed on AD CD-ROM systems, but you should verify that it exists and is correct. If you need to add the member, you will also need to add the operand:

```
IOS=xx,          USE MIH MEMBER (REMEMBER THE COMMA!)
```

to your IEASYSxx member(s). (This can be a nontrivial task. You must find all the IEASYSxx members that might be invoked by all the IPLPARM members you might use. These IEASYSxx members might be in SYS1.PARMLIB or in concatenated PARMLIBs. These details tend to change with every release of the AD CD-ROM.)

If you need to add or change the IECIOSxx member, it will not be effective until the next IPL. If you added or changed the member you can avoid an immediate IPL by issuing (from the MVS master console) the command:

```
SETIOS MIH,TIME=00:00,DEV=(E20-E21)
```

The SETIOS command is effective until the next IPL, at which time your PARMLIB member will be used.

You can verify that the MIH is set correctly by using the operator command:

```
D IOS,MIH,DEV=(E20-E21)
```

This should display output such as:

```
0E20=00:00 0E21=00:00
```

We strongly suggest you use this command to verify that your MIH is set up correctly. Adding or changing IECIOS members can be confusing when multiple PARMLIBs and IPLPARMs are involved. The D IOS,MIH command will verify that you have updated the correct IEASYSxx member in the correct PARMLIB (after you reIPL).

You should next display the unit status of the addresses for your emulated 3172 connections. Issue the command:

```
D U,,,E20,2 (Use your correct address, of course)
```

This should display:

```
IEE457I 11:04:42 UNIT STATUS 700
UNIT TYPE STATUS VOLSER VOLSTATE
0E20 CTC 0
0E21 CTC 0
```

This is a critical test. If the status is not O (meaning Online) there is a problem with the server setup or the P/390 configurator settings. Further efforts at OS/390 TCP/IP customization are pointless until the underlying server problem is corrected. If the status is:

```
IEE457I 11:04:42 UNIT STATUS 700
UNIT TYPE STATUS VOLSER VOLSTATE
0E20 CTC A-BSY
0E21 CTC A
```

then OS/390 TCP/IP is probably already active and using these addresses. (We say probably because it is possible to have some other job/task using these addresses; this is unlikely if you are working with a new AD CD-ROM installation.) If this is your case, you might want to stop OS/390 TCP/IP until you complete your customization. You can do this with the OS/390 command P TCPIP, assuming the started task name is TCPIP.

6.22.3 RACF and TCP/IP

The AD CD-ROM should already have the necessary RACF customization defined. If you are not using the AD system, you will need to spend some time with TCP/IP installation documentation in order to complete the RACF requirements.

6.22.4 Procedures and data sets

OS/390 TCP/IP references many data sets and files to obtain its customization parameters. These data sets and files do not have standard names, and the methods OS/390 TCP/IP uses to find them can be confusing. The methods are described, at length, in *OS/390 IBM Communications Server: IP Configuration Reference*, SC31-8726, or later document. For our purposes, we assume you are using the AD CD-ROM system and the data sets/files it provides.

The starting point for TCP/IP customization is the JCL for its started procedure. In the OS/390 AD CD-ROM, this is in SYS1.PROCLIB(TCPIP):

```
//TCPIP PROC PARM='CTRACE(CTIEZB00) '
//TCPIP EXEC PGM=EZBTCPIP,PARM='&PARMS',REGION=7500K,TIME=1440
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FB,LRECL=137,BLKSIZE=137)
//ALGPRINT DD SYSOUT=*,DCB=(RECFM=FB,LRECL=137,BLKSIZE=137)
//SYSOUT DD SYSOUT=*,DCB=(RECFM=FB,LRECL=137,BLKSIZE=137)
//CEEDUMP DD SYSOUT=*,DCB=(RECFM=FB,LRECL=137,BLKSIZE=137)
```

```
//SYSMDUMP DD DUMMY
//PROFILE DD DISP=SHR,DSN=TCPIP.PROFILE.TCPIP
//SYSTCPD DD DISP=SHR,DSN=TCPIP.TCPIP.DATA
```

and is started by issuing S TCPIP from the MVS master console. (It can also be started automatically from PARMLIB(COMMNDxx) or equivalent.) Note that the started task name is *TCPIP*. You cannot readily change this name; it is coded in the BPXPRMxx member of PARMLIB.

The key element of this procedure is that it contains DD statements for PROFILE and SYSTCPD. If these DD statements are not present, TCP/IP will follow a number of paths attempting to find data sets containing PROFILE and DATA parameters. The names shown above have been used in all recent OS/390 CD-ROM releases, although they might change in future releases. In the remainder of this discussion, we will refer to these as the PROFILE and DATA data sets. To find the actual data set names, you should check the JCL for your TCPIP started task.

These statements assume that the started task name is TCPIP. This should remain standard for AD CD-ROM systems. Other OS/390 systems may have different started task names. Furthermore, the started task procedure might not be in SYS1.PROCLIB. If the JCL does not contain PROFILE and SYSTCPD DD statements, you will need to examine the whole process by which OS/390 TCP/IP finds its parameters. This is beyond the scope of this discussion.

Assuming the TCPIP started task described here, your initial OS/390 TCP/IP customization involves a few lines in the PROFILE data set.

6.22.5 TCP/IP profile

The complete PROFILE data set, as used in the author's V2R10 system is listed here. Comment lines have been removed for brevity:

```
ARPAGE 5
TELNETPARMS
  PORT 23
  INACTIVE 28800
  TIMEMARK 28800
  SCANINTERVAL 600
  SMFINIT STD
  SMFTERM STD
  WLMCLUSTERNAME TN3270E ENDWLMCLUSTERNAME
ENDTELNETPARMS
IPCONFIG
  NODATAGRAMFWD
  IGNOREREDIRECT
TCPCONFIG
  RESTRICTLOWPORTS
UDPCONFIG
  RESTRICTLOWPORTS

DATASETPREFIX TCPIP

AUTOLOG 5
  FTPD JOBNAME FTPD1 ; FTP Server
  PORTMAP ; Portmap Server
ENDAUTOLOG

PORT
  7 UDP MISCSERV ; Miscellaneous Server
```

```

7 TCP MISCERV
9 UDP MISCERV
9 TCP MISCERV
19 UDP MISCERV
19 TCP MISCERV
20 TCP OMVS      NOAUTOLOG ; FTP Server
21 TCP OMVS      ; FTP Server
23 TCP INTCLIEN ; Telnet Server
25 TCP SMTP      ; SMTP Server
53 TCP NAMESRV   ; Domain Name Server
53 UDP NAMESRV   ; Domain Name Server
69 UDP OMVS      ; OE TFTP SERVER
80 TCP OMVS      ; OE WEB SERVER
111 TCP PORTMAP  ; Portmap Server
111 UDP PORTMAP  ; Portmap Server
135 UDP LLBD     ; NCS Location Broker
161 UDP OSNMPD   ; SNMP Agent
162 UDP SNMPQE   ; SNMP Query Engine
433 TCP OMVS     ; OE WEB Server
443 TCP OMVS     ; Secure Server
512 TCP RXSERVE  ; Remote Execution Server
513 UDP OMVS     ; OE RLOGIN SERVER
514 UDP OMVS     ; OE syslog server
514 TCP RXSERVE  ; Remote Execution Server
515 TCP LPSERVE  ; LPD Server
520 UDP OROUTED ; RouteD Server
580 UDP NCPROUT ; NCPROUTE Server
750 TCP MVSKERB  ; Kerberos
750 UDP MVSKERB  ; Kerberos
751 TCP ADM@SRV  ; Kerberos Admin Server
751 UDP ADM@SRV  ; Kerberos Admin Server
; 1021 TCP OMVS  ; OE FTP SERVER
1023 TCP OMVS    ; OE TELNET SERVER
1023 UDP OMVS    ; OE TELNET SERVER
1024 TCP OMVS    ; OE SERVICES
1415 TCP CSQ1CHIN ; CSQ1 MQ TCP Listener
3000 TCP CICSTCP ; CICS Socket

DEVICE LCS1 LCS E20
LINK ETH1 ETHERNET 1 LCS1

HOME
9.12.17.151 ETH1
GATEWAY
; Network First Hop Link Name Packet Size Subnet Mask Subnet Value
9 = ETH1 1492 0.255.255.000 0.12.17.00

ITRACE OFF

ASSORTEDPARMS
NOFWD
RESTRICTLOWPORTS
ENDASSORTEDPARMS

BEGINVTAM
TELNETDEVICE 3278-3-E NSX32703 ; 32 line screen - default of NSX327024is 2
TELNETDEVICE 3279-3-E NSX32703 ; 32 line screen - default of NSX327024is 2
TELNETDEVICE 3278-4-E NSX32704 ; 48 line screen - default of NSX327024is 2
TELNETDEVICE 3279-4-E NSX32704 ; 48 line screen - default of NSX327024is 2
TELNETDEVICE 3278-5-E NSX32705 ; 132 column screen - default of NSX32702 is 80

```

```

TELNETDEVICE 3279-5-E NSX32705 ; 132 column screen - default of NSX32702 is 80
DEFAULTLUS
    SCOTCP01 SCOTCP02 SCOTCP03 SCOTCP04 SCOTCP05
    SCOTCP06 SCOTCP07 SCOTCP08 SCOTCP09 SCOTCP10
    SCOTCP11 SCOTCP12 SCOTCP13 SCOTCP14 SCOTCP15
    SCOTCP16 SCOTCP17 SCOTCP18 SCOTCP19 SCOTCP20
    SCOTCP21 SCOTCP22 SCOTCP23 SCOTCP24 SCOTCP25
    SCOTCP26 SCOTCP27 SCOTCP28 SCOTCP29 SCOTCP30
ENDEFAULTLUS
LUSESSIONPEND ; On termination of a Telnet server connection,
                ; the user will revert to the DEFAULTAPPL
; DEFAULTAPPL TSO ; Set the default application for all Telnet
USSTCP USSN ; USS Table name
                ; sessions to allow CLSDST Pass
LINEMODEAPPL TSO ; Send all line-mode terminals directly to TSO.
ALLOWAPPL SAMON QSESSION ; SAMON appl does CLSDST Pass to next appl
ALLOWAPPL TSO* DISCONNECTABLE ; Allow all users access to TSO
                ; applications.
                ; TSO is multiple applications all beginning with TSO,
                ; so use the * to get them all. If a session is closed,
                ; disconnect the user rather than log off the user.
RESTRICTAPPL IMS ; Only 3 users can use IMS.
    USER USER1 ; Allow user1 access.
    LU TCPIMS01 ; Assign USER1 LU TCPIMS01.
    USER USER2 ; Allow user2 access from the default LU pool.
    USER USER3 ; Allow user3 access from 3 Telnet sessions,
                ; each with a different reserved LU.
    LU TCPIMS31 LU TCPIMS32 LU TCPIMS33
ALLOWAPPL * ; Allow all applications that have not been
                ; previously specified to be accessed.
ENDVTAM

START LCS1

```

Basic adapter customization

Your AD CD-ROM system should have a PROFILE data set similar to this. We noticed, starting with the V2R5 AD CD-ROM system, that an executing TCPIP has exclusive control on the PROFILE data set. We needed to stop TCP/IP before we could edit the PROFILE data set. Your initial customization, to get basic TCP/IP functions working with your IP network involves the following few lines:

```

        DEVICE LCS1 LCS E20
        LINK ETH1 ETHERNET 1 LCS1

        HOME
        9.12.17.151 ETH1

        GATEWAY
        9 = ETH1 1492 0.255.255.0 0.12.17.0
...
START LCS1

```

The illustrated parameters conform to our example in which we have a P/390 or MP3000 with an Ethernet adapter dedicated to OS/390 TCP/IP and that will use OS/390 addresses E20 and E21. The adapter is connected via the LCS3172 device manager.

The DEVICE statement should appear approximately as shown. DEVICE and LCS are keywords and must not be changed. LCS1 (as shown) is an arbitrary *device name*; any name can be used provided it is also used in the LINK and START statements. E20 (as shown) is the OS/390 address; it is the first (even) address of an even/odd pair. As shown here, it specifies the addresses E20 and E21.

The LINK statement has four parameters. The first parameter is an arbitrary *link name*; any name can be used provided it is later used in HOME, GATEWAY, and other statements. The second parameter is the type of protocol involved; common keyword values are ETHERNET, ETHERNETor802.3, or IBMTR. (If you are uncertain whether your Ethernet network is DIX Ethernet or IEEE802.3 Ethernet, you can use the ETHERNETor802.3 option. (You can use all upper-case letters to specify it.)

The third parameter is the LAN adapter number; this is the adapter number from the Current Configuration window in OS/2 MPTS.

The fourth parameter is a device name—the same name you used in a DEVICE statement earlier.

If you have multiple interfaces (adapters) for OS/390 TCP/IP, there will be one DEVICE statement and one LINK statement for each interface.

The HOME statement is followed by your IP address and the associated link name. If you have multiple interfaces, you would have one IP-address/link-name line for each interface. This is the single location where you set your IP address. (If you have multiple interfaces, you will have multiple IP addresses, of course. You can have multiple IP addresses with a single interface, but this is beyond the scope of this document.)

Static routing customization

The GATEWAY statements provide static routing. You might later remove these statements if you implement. For your initial customization, we suggest you use static routing. The format of the gateway statement is unusual and is best explained with examples.⁵ We have added a comment line that helps identify the operands:

```
GATEWAY
;Network FirstHop Link PSize SubnetMask SubnetValue
  9      =      ETH1 1492  0.255.255.0 0.12.17.0
```

The first parameter is the network number; ours is 9. The network number should correspond to the class of network address. A class A network has an address between 1 and 126. (127.0.0.1 is automatically used as the loopback address; do not define it here.) A class B network has an address xxx.xxx, where the first xxx is between 128 and 191. A class C network has an address xxx.xxx.xxx, where the first xxx is 192 to 223. Do not include any subnet portion of the address in the Network column.

In the FirstHop column, the symbol = means to use the directly-attached LINK that is named in the third column. The PSize column specifies the maximum packet size for this link. (We suggest 1492 as a starting size for Ethernet. Do not use 1500 for a P/390 or MP3000.)

The SubnetMask column specifies the additional mask to be added to the network class in order to produce your subnet. To subnet a class A address to a class C subnet, the mask would be 0.255.255.0. To subnet a class B address to a class C subnet, the mask would be 0.0.255.0.

⁵ The rather rigid format described here should work with all releases of OS/390. Later releases have more adaptable formats, and an alternate format is briefly described in the next section.

The SubnetValue column specifies the address of your subnet. Do not duplicate the network address; specify only the subnet portion of your address.

More examples might help:

```
Your desired subnet = 151.123.1.0
      Network = 151.123      (a class B address)
      SubnetMask = 0.0.255.0
      (Effective netmask = 255.255.255.0)
      SubnetValue = 0.0.1.0      (your subnet)
```

```
Your desired subnet = 192.168.25.0
      Network = 192.168.25   (a class C address)
      SubnetMask = 0
      (Effective netmask = 255.255.255.0)
      SubnetValue = 0
```

```
Your desired subnet = 10.0.0.0
      Network = 10           (a class A address)
      SubnetMask = 0.255.255.0
      (Effective netmask = 255.255.255.0)
      SubnetValue = 0
```

All three examples assume you want a single byte for your host addresses. A slightly more complex GATEWAY set might be:

```
GATEWAY
;Network FirstHop Link PSize SubnetMask SubnetValue
9         =       ETH1 1492  0.255.255.0 0.12.17.0
193.1.2   9.12.17.1 ETH1 1492  0
```

This set defines the system as being attached to network 9.12.17. (Network 9 with subnet 12.17.) Furthermore, packets to network 193.1.2 should be sent to address 9.12.17.1, whose network (9.12.17) has already been defined in a statement in the GATEWAY section. In this case, 193.1.2 is a class C address and no subnet information is needed. You must specify the 0 for the SubnetMask.

You generally describe all your directly connected networks first; these have the FirstHop value =. You can then define any number of static routes. The FirstHop value in any line must be capable of being resolved to a directly-attached link, using previously defined routes.

One more example may be useful:

```
GATEWAY
;Network FirstHop Link PSize SubnetMask SubnetValue
9         =       ETH1 1492  0.255.255.0 0.12.17.0
193.1.2   9.12.17.1 ETH1 1492  0
DEFAULTNET 9.12.17.50 ETH1 DEFAULTSIZE 0
```

The DEFAULTNET statement indicates that all addresses not routed by earlier GATEWAY statements are to be sent to the system at 9.12.17.50. In this example, routes for networks 9.12.17 and 193.1.2 are specified in previous statements. The routing for this network is known from the first statement in the GATEWAY section. The DEFAULTSIZE operand (for the maximum packet size) is equivalent to a maximum packet size of 576. (This is the default Internet packet size. Discussions about packet sizes can be found in many Internet and TCP/IP texts. 576 is a safe size for packet groups that may go through many intermediate nodes.) You normally specify a SubnetMask value of 0 for a DEFAULTNET statement. You may have only one DEFAULTNET statement, and it is normally the last statement in the GATEWAY section.

At the end of your PROFILE, you need a START statement for the device name specified in your DEVICE statement. If you have multiple DEVICE statements, you would normally have a START statement for each device.

Alternate format

The GATEWAY statement syntax is confusing to many users. Recent releases of OS/390 TCP/IP added an alternate parameter, BEGINROUTES, that can be used instead of the GATEWAY statement. You can use either GATEWAY or BEGINROUTES, but not both.

Examples of the BEGINROUTES parameters are:

```

DEVICE LCS1  LCS      E20
LINK ETH1  ETHERNET 1  LCS1

HOME
  9.12.17.151 ETH1
BEGINROUTES
;      Destination      Mask          First-Hop  Link  MTU Options
ROUTE 9.12.17.0        255.255.255.0  =          ETH1  mtu 1492
ROUTE 193.1.2          255.255.255.0  9.12.17.150 ETH1  mtu 1492
ROUTE default          9.12.17.150 ETH1  mtu 1492
ENDROUTES

```

The syntax of the BEGINROUTES parameters is BSD-like and may be more intuitive than the syntax of the GATEWAY parameters. See *OS/390 IBM Communications Server: IP Configuration Reference*, SG31-8726 for more information about these parameters.

Other PROFILE customizations

Your PROFILE data set (assuming you are using an AD CD-ROM system) should be similar to the one in the earlier listing—although it will contain many comment lines not shown here. We suggest you do not change any statements unless you are reasonably certain you know what you are doing. A brief discussion of some of the statements in this PROFILE data set may be helpful:

- ▶ ARP information will be retained 5 minutes before it is discarded.
- ▶ The standard MVS telnet server (primarily designed for TN3270 connections that interface with VTAM) is present on port 23. See the TELNETPARMS statements and the port 23 line in the PORT statements. INTCLIEN is the name of the standard MVS telnet server.
- ▶ Unauthorized applications cannot obtain port addresses lower than 1023; this is the effect of the RESTRICTLOWPORTS operands.
- ▶ A started procedure named FTPD will be automatically started, and restarted if it fails. The AUTOLOG statement causes this. The jobname will be FTPD1. A started task named PORTMAP is also automatically started and restarted, if necessary.
- ▶ The PORT section simply reserves port numbers for specific application names. It does not start the associated applications. The application name *OMVS* means that the application claiming the indicated port must be started from OMVS. (OMVS, OE, OpenEdition, and OS/390 UNIX System Services are all synonyms, as used here.)
- ▶ Note that port 1023 (both UDP and TCP) is reserved for an OMVS application. Later customization will use this port for *otelnet* (the application that accepts ASCII telnet connections for the Open Edition shell).
- ▶ The ASSORTEDPARMS parameter NOFWD or FWD determines whether OS/390 TCP/IP will perform IP forwarding. This may be a security issue for your installation. For a small development system, such as a P/390 using an AD CD-ROM OS/390 base, you probably want the FWD parameter. With the FWD parameter, your system will act like a gateway or router. (The terms *gateway* and *router* were well defined in the earlier days of TCP/IP.

Their meanings have become somewhat fuzzy now, and they are used here in a rather general sense.)

- ▶ Do not even think about changing the TELNETDEVICE lines unless you understand VTAM mode tables or unless you are following carefully documented, tested advice.
- ▶ You should ensure there are enough DEFAULTLUS defined to handle all the concurrent TN3270 connections to INTCLIEN (normally your port 23). If necessary, simply define more using the pattern shown. The base part of the LU name (SC0TCPnn in the examples) must correspond with VTAM definitions, so do not change this unless you make corresponding VTAM changes.
- ▶ The distributed PROFILE may have some VTAM IMS definitions not shown here. These are examples of ways to limit the number of VTAM connections and can be used or deleted.

There are many PROFILE options and alternatives not discussed here. These are documented in *OS/390 IBM Communications Server: IO Configuration Reference*, SC31-8726. We strongly suggest you obtain a copy of this manual.

6.22.6 Other basic customization

The data sets, files, and customization described in this section should be already complete in your AD CD-ROM system, except for your host names in `/etc/host` and the corresponding MVS files. The details are very briefly described here to assist you in locating the relevant data sets and files and to provide a starting point for making changes (*after* your basic OS/390 TCP/IP is operational).

The DATA data set and `/etc/resolv.conf`

Our TCP/IP started task JCL, shown in “Procedures and data sets” on page 295, contains a DD statement with DDname SYSTCPD. In the JCL listing, this points to TCPIP.TCPIP.DATA. (This is not a standard name and may change in future releases of the AD CD-ROM. You can find the name by examining the TCP/IP started task procedure.) We will refer to this as TCP/IP's DATA data set.

The DATA data set contains several global controls for OS/390 TCP/IP. Our working version, with many comment lines removed, is:

```
TCPIPJOBNAME TCPIP
HOSTNAME MVSEA
DOMAINORIGIN ITSO.IBM.COM
; NSINTERADDR specifies the internet address of the name server.
; LOOPBACK (14.0.0.0) is the default value (your local name server).
; If a name server will not be used, then do not code an NSINTERADDR
; statement (Comment out the NSINTERADDR line below). This will cause
; all names to be resolved via site table lookup.
; NSINTERADDR 9.14.1.30
; NSPORTADDR specifies the foreign port of the Name Server.
; 53 is the default value.
; NSPORTADDR 53
; RESOLVEVIA UDP
; RESOLVERTIMEOUT 30
; RESOLVERUDPRETRIES 1
;
; These datasets may be allocated dynamically:
; datasetprefix.STANDARD.TCPXLBIN (translation tables)
; datasetprefix.ETC.PROTO
; datasetprefix.ETC.SERVICES
; datasetprefix.HOSTS.xxxxxxxx
```

```

;
DATASETPREFIX TCPIP
ALWAYSUTO YES

```

Note that the Domain Name Server address (NSINTERADDR) is commented out. The distributed AD systems usually have a name server address included. This address (an IBM internal name server) may not be useful to you and we suggest you comment out the distributed name server address or change it to one that is useful to you.

The DATA data set provides key parameters for *some* TCP/IP applications and functions. Other applications and functions obtain similar information from /etc/resolv.conf (if it exists). If the /etc/resolv.conf or /etc/services or /etc/hosts files do not exist, TCP/IP (when using the OE sockets) should go to the equivalent MVS files. However, you may need some of the HFS versions of the files --- /etc/services in particular --- and most owners appear to define both the MVS files and the HFS files. We assume this is the case in the following discussions.

Our /etc/resolv.conf looked like this:

```

TCPIPJobname      TCPIP      ;
Datasetprefix     TCPIP      ;
Messagecase       mixed      ;
HostName          mvsea      ;
DomainOrigin      itso.ibm.com;
; NSinterAddr     9.14.1.30   ;
; NSportAddr      53         ;
ResolveVia        UDP        ;
ResolveTimeout    10         ;
ResolveUdpRetries 1          ;

```

This is approximately the same information that appears in the DATA data set. (Since the NSinterAddr address is commented out in the /etc/resolv.conf file, there is no need to comment out the other Resolve values.) Unless you have a good reason to do otherwise, we suggest you should keep your DATA data set and your /etc/resolv.conf file synchronized. This dual structure is a result of earlier implementations of several different TCP/IP *stacks* and several sockets programming interfaces.

The TCPIP job name is the name of the started task. The data set prefix is the qualifier used to dynamically locate required data sets (described later). The HOSTNAME and DOMAIN ORIGIN are used as implied by their names. (Many TCP/IP utilities and library functions append the domain name to any host name they receive that does not contain at least one period in the supplied name.

NSINTERADDR is the address of a name server. If none is specified (which is the case in our files), the appropriate library routines will use the HOSTS.LOCAL and /etc/hosts files described below. If you provide an address for NSINTERADDR there are several additional NS controls you can specify; their purpose is apparent from their parameter names.

6.22.7 FTP customization

Earlier versions of OS/390 TCP/IP had two ftp servers; one for MVS and one for OpenEdition. Current OS/390 (V2R5 and later) has a single ftp server. It can handle both traditional MVS data sets and HFS files. The ftp server is usually started by the AUTOLOG function in the PROFILE data set.

The AUTOLOG function starts a started task, usually named FTPD. The procedure can be found in SYS1.PROCLIB(FTPD):

```

//FTPD  PROC  MODULE='FTPD',PARMS=' '
//FTPD  EXEC  PGM=&MODULE,REGION=4096K,TIME=NOLIMIT,

```

```
//          PARM='POSIX(ON) ALL31(ON)&PARMS'
//CEEDUMP DD SYSOUT=*
//SYSFTPD DD DISP=SHR,DSN=TCPIP.FTP.DATA
//SYSTCPD DD DISP=SHR,DSN=TCPIP.TCPIP.DATA
```

The SYSFTPD DD statement points to an ftp control file. This appears as follows (with many comments removed):

```
;ANONYMOUS                ; anonymous login accepted
;ASATRANS      FALSE      ; do NOT translate control characters
AUTOMOUNT      TRUE       ; automatic mount of unmounted volume
AUTORECALL     TRUE       ; automatic recall of migrated data sets
;AUTOTAPEMOUNT FALSE     ; do NOT automatically mount tape volumes
BLOCKSIZE      6233       ; new data set allocation blocksize
CONDDISP       CATLG      ; data sets catalogued if transfer fails
;CTRLCONN      IBM-850     ; ascii code set for control connection
;DATACLASS     SMSDATA    ; sms data class name
;DB2           D31        ; db2 subsystem name
;DB2PLAN       PLANNAME   ; db2 plan name for OE-FTP
;DCBDSN        MODEL.DCB  ; new data set allocation model dcb name
;DEST          USER14@MVSL ; files destination for store
DIRECTORY      27         ; new data set allocation directory blocks
DIRECTORYMODE  FALSE      ; directorymode vs. data set mode
FILETYPE       SEQ        ; file transfer mode
INACTIVE       300        ; inactive time out (seconds)
JESLRECL       80         ; lrecl of jes jobs
JESPUTGETTO    600        ; timeout for remote job submission put/ge
JESRECFM       F          ; recfm of jes jobs
LRECL          256        ; new data set allocation lrecl
;MGMTCLASS     SMSMGMT    ; sms mgmtclass name
;MIGRATEVOL    MIGRAT     ; migration volume volser
PRIMARY        1          ; new data set allocation primary space
;QUOTESOVERRIDE FALSE    ; single quote(s) are treated as part of
;              ; hfs filename, i.e. single quotes do
;              ; NOT indicate working directory override
RECFM          VB         ; new data set allocation record format
;RETPD         30         ; new data set retention period: 30 days
;SBDATACONN    (IBM-1047,IBM-850) ; ebcdic/ascii code sets for data conn.
SECONDARY      1          ; new data set allocation secondary space
;SMF           STD        ; SMF records use standard subtypes
;SMFEXIT       ;          ; load SMF user exit FTPSMFEX
;SMFJES        ;          ; SMF recording when filetype=jes
;SMFSQL        ;          ; SMF recording when filetype=sql
SPACETYPE      TRACK      ; new data set allocation space type
SPREAD         FALSE      ; sql output format
SQLCOL         NAMES      ; sql output uses column names as headings
;STARTDIR      MVS        ; use MVS directory at connect time
;STORCLASS     SMSSTOR    ; sms storclass name
;TRACE         ;          ; trace active
;TRAILINGBLANKS TRUE      ; include trailing blanks when fixed
;              ; format data sets are retrieved
;UMASK         027        ; octal UMASK to restrict setting
;              ; of permission bits when creating
;              ; new hfs files
;UNITNAME      3380       ; new data set allocation unit
;VOLUME        WRKLB2     ; new data set allocation volume serial
WRAPRECORD     FALSE      ; data is NOT wrapped to next record
```

Read this listing carefully since many lines are commented out. The meaning of most of the controls is apparent, although some of the specified values might not be appropriate for you. In particular, you might want to consider and change the AUTOMOUNT (change to FALSE), BLOCKSIZE (change to half-track blocking), PRIMARY (change to something larger than one track) and INACTIVE (change to a larger timeout number, such as 1200 seconds) parameters. The possible parameters are explained in the IP Configuration Reference manual (SC31-8726).

In addition to simple file transfers, the ftp server can submit jobs to JES2 and run DB2 SQL. Setup information is in the IP Configuration Reference manual. The TCP/IP setup we are describing here does not have these features enabled.

Use of this ftp server is straightforward. From an ftp client, you request to GET or PUT a host file. If the host file name is a traditional data set name, a traditional MVS data set is used. If the host file name is an HFS name, then an HFS file is used. The current host directory can be an HFS directory, a catalog listing based on an HLQ, or a member listing of a PDS. Which is used depends on the format of your **cd** commands from the client.

6.22.8 inetd customization

A started task or process is needed for INET. This is normally started from the /etc/rc file, with the following line:

```
_BPX_JOBNAME='INETD' /usr/sbin/inetd /etc/inetd.conf &
```

It could also be started from a procedure (placed in SYS1.PROCLIB(INETD) if there is a problem starting it from the rc file.

```
//INETD PROC
//INETD EXEC PGM=BPXBATCH,REGION=64M,TIME=NOLIMIT,
//          PARM='PGM /usr/sbin/inetd /etc/inetd.conf'
//* Note the required lower case name above
//SYSPRINT DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD DUMMY
```

This procedure (listed from the R10 AD system) appears to contain two errors. The PARM should probably be 'PGM /usr/sbin/inetd /etc/inetd.conf'.

However inetd is started, it references /etc/inetd.conf for parameters. The basic content of this file, for the AD CD-ROM system, should be:

```
otelnet stream tcp nowait OMVSKERN /usr/sbin/otelnetsd otelnetsd
logon stream tcp nowait OMVSKERN /usr/sbin/rshd rshd -LV
```

For the basic TCP/IP startup we are describing, the otelnet entry in /etc/inetd.conf is important. Remember that:

- ▶ A TN3270 connection to port 23 will produce a 3270 logon connection via VTAM. This uses the INTCLIEN function defined in the PROFILE data set.
- ▶ A telnet connection to port 23 will produce a line mode TSO connection, via VTAM, and is fairly useless.
- ▶ A telnet connection to the port associated with otelnet will produce an ASCII-type connection to the OMVS shell. You can run the vi editor, for example, with this connection. If you use the customization outlined in this paper, otelnet will appear at port 1023.

6.22.9 Local host name customization

If you do not use a domain name server (the DATA and /etc/resolv.conf files listed above specify no name server), TCP/IP will attempt to resolve names using local files. There are two of these files, one for traditional MVS service interfaces and one for OE interfaces. The OE file is /etc/hosts and is in the same format used on many other platforms. Our /etc/hosts file contained:

```
9.12.17.150  os2a
9.12.17.151  mvsea.itso.ibm.com
9.12.17.151  testit
```

The equivalent MVS data set is more complex; there are several data sets involved. They are TCPIP.HOSTS.LOCAL, TCPIP.HOSTS.ADDRINFO, and TCPIP.HOSTS.SITEINFO. (The HLQ TCPIP is controlled by the DATASET PREFIX value in the DATA data set. We will assume this value is set to TCPIP.) You create and edit the TCPIP.HOSTS.LOCAL file and then execute the MAKESITE program to create the other two data sets.

Our TCPIP.HOSTS.LOCAL data sets were like this:

```
HOST : 9.12.17.150 : os2a :::
HOST : 9.12.17.151 : mvsea.itso.ibm.com :::
```

(We intentionally left out the *testit* name.) All the colons are needed, exactly as shown. After creating this data set, issue this command from a TSO command line:

```
MAKESITE HLQ=TCPIP,VOLSER=vvvvvv,UNIT=uuuu
```

For the AD CD-ROM system, we used VOLSER=OS39M1 and UNIT=3390. You need to stop and restart TCP/IP in order to use the new data.

You will normally want to keep your /etc/hosts and your HOSTS data the same. Remember to run the MAKESITE command every time you change the HOSTS file. The *IP Configuration Reference* manual contains information about /etc/hosts and the HOSTS data set in the chapter "Operator Commands and System Administration". The TSO command TESTSITE is provided to try the results of MAKESITE.

6.22.10 TCP/IP services definitions

The /etc/services file is used to associate port numbers with various OE-based TCP/IP services. A similar data set, TCPIP.ETC.SERVICES, is used to associate port numbers with various MVS-based TCP/IP services. The file and data set are similar, but not identical. In particular, the port for otelnet is assigned in the /etc/services file. You should not need to change the file and data set provided with your AD CD-ROM system.

Our AC CD-ROM had the following /etc/services file:

```
echo          7/tcp
echo          7/udp
discard      9/tcp          sink null
discard      9/udp          sink null
systat       11/tcp          users
daytime      13/tcp
daytime      13/udp
netstat      15/tcp
qotd         17/tcp          quote
chargen      19/tcp          ttytst source
chargen      19/udp          ttytst source
ftp          21/tcp
otelnet      1023/tcp          <-----
```



```

smtp          25/tcp      mail
time         37/tcp      timserver
time         37/udp      timserver
rlp          39/udp      resource      # resource location
nameserver   42/tcp      name          # IEN 116
whois        43/tcp      nickname
domain       53/tcp      nameserver    # name-domain server
domain       53/udp      nameserver
mtp          57/tcp      # deprecated
tftp         69/udp
rje          77/tcp      netrjs
finger       79/tcp
http         80/tcp      www
link         87/tcp      ttylink
supdup       95/tcp
hostnames    101/tcp     hostname      # usually from sri-nic
#csnet-cs    105/?
pop          109/tcp     postoffice
sunrpc       111/tcp
sunrpc       111/udp
auth         113/tcp     authentication
sftp         115/tcp
uucp-path    117/tcp
nntp         119/tcp     readnews untp # USENET News Transfer
snmp         161/udp     # snmp request port
snmp-trap    162/udp     # snmp monitor trap por
# UNIX specific services
exec         512/tcp
biff         512/udp     comsat
login        513/tcp
who          513/udp     whod
shell        514/tcp     cmd           # no passwords used
syslog       514/udp
printer      515/tcp     spooler       # line printer spooler
talk         517/udp
ntalk        518/udp
efs          520/tcp     # for LucasFilm
timed        525/udp     timeserver
tempo        526/tcp     newdate
courier      530/tcp     rpc
conference   531/tcp     chat
netnews      532/tcp     readnews
netwall      533/udp     # -for emergency broadcast
uucp         540/tcp     # uucp daemon
remotefs     556/tcp     rfs_server rfs # Brunhoff remote filesystem
ingreslock   1524/tcp
# Start of IBM added services ...
router       520/udp     routd
ncprout      580/udp     ncproute
# RVD service
rvd-control  531/udp     # rvd control port
# Andrew File System services
filesrv      2001/tcp
console      2018/udp
venus.itc    2106/tcp
# For file server backup and migration
client       2030/tcp
# Andrew File System Authenticated Services
vexec        712/tcp     vice-exec
vlogin       713/tcp     vice-login

```

```

rshell          714/tcp  vice-shell
# for the Venus process
venus.itc      2106/tcp
rauth2         2001/udp
rfilebulk     2002/udp
rfilesrv      2003/udp
ropcons       2115/udp
# in pairs
rupdsrv       2131/udp
rupdbulk      2132/udp
# (followed by a number of Kerberos definitions)

```

The same system had the following data set for TCPIP.ETC.SERVICES. The contents are in lowercase, which is unusual for MVS-based parameter files:

```

echo           7/tcp
echo           7/udp
discard       9/tcp          sink null
discard       9/udp          sink null
systat        11/tcp          users
daytime       13/tcp
daytime       13/udp
netstat       15/tcp
qotd          17/tcp          quote
chargen       19/tcp          ttytst source
chargen       19/udp          ttytst source
ftp           21/tcp
telnet        23/tcp
smtp          25/tcp          mail
time          37/tcp          timserver
time          37/udp          timserver
rlp           39/udp          resource      # resource location
nameserver    42/tcp          name          # IEN 116
whois         43/tcp          nickname
domain        53/tcp          nameserver    # name-domain server
domain        53/udp          nameserver
mtp           57/tcp
tftp          69/udp
rje           77/tcp          netrjs
finger        79/tcp
link          87/tcp          ttylink
supdup        95/tcp
hostnames     101/tcp          hostname      # usually from sri-nic
#csnet-cs     105/?
pop           109/tcp          postoffice
sunrpc        111/tcp
sunrpc        111/udp
auth          113/tcp          authentication
sftp          115/tcp
uucp-path     117/tcp
nntp          119/tcp          readnews untp # USENET News Transfer Protocol
snmp          161/udp          # snmp request port
snmp-trap     162/udp          # snmp monitor trap port
# UNIX specific services
exec          512/tcp
biff          512/udp          comsat
login         513/tcp
who           513/udp          whod
shell         514/tcp          cmd           # no passwords used
syslog        514/udp
printer       515/tcp          spooler       # line printer spooler

```

```

talk          517/udp
ntalk        518/udp
efs          520/tcp          # for LucasFilm
timed        525/udp          timeserver
tempo        526/tcp          newdate
courier      530/tcp          rpc
conference   531/tcp          chat
netnews      532/tcp          readnews
netwall      533/udp          # -for emergency broadcasts
uucp         540/tcp          uucpd          # uucp daemon
remotefs     556/tcp          rfs_server rfs # Brunhoff remote filesystem

ingreslock   1524/tcp
# Start of IBM added services ...
route        520/udp          router routed
ncprout      580/udp          ncproute
# RVD service
rvd-control  531/udp          # rvd control port
# Andrew File System services
filesrv      2001/tcp
console      2018/udp
venus.itc    2106/tcp
# For file server backup and migration
client       2030/tcp
# Andrew File System Authenticated services
vexec        712/tcp          vice-exec
vlogin       713/tcp          vice-login
vshell       714/tcp          vice-shell
# For the Venus process.
venus.itc    2106/tcp
rauth2       2001/udp
rfilebulk    2002/udp
rfilesrv     2003/udp
ropcons      2115/udp
# The following are assigned in pairs and the bulk must be the srv +1
rupdsrv      2131/udp
rupdbulk     2132/udp
rupdsrv1     2133/udp
rupdbulk1    2134/udp
# (followed by a number of Kerberos definitions)

```

Both versions contain port assignments for services you may not want or need. We suggest you use the file and data set as provided, unless you have specific reasons for changing them. You might want to change the port number used for otelnet. If you switch the ports for otelent (port 1023) and INTCLIEN (port 23), be certain to change TELNETPARMS in the PROFILE data set as well.

6.22.11 Other setups and customizations

Several entries have been added to the SCHEDxx and IEFSSNxx members in PARMLIB. This has already been done for the AD CD-ROM, but you might want to verify it. Remember that there are multiple PARMLIBs, and finding the members actually used for a particular IPL may require some effort in tracing the effects of the IPLPARM parameter used.

PARMLIB(SCHEDxx) should have these sections:

```

PPT PGMNAME(MVPTNF)
    KEY(0)          (storage protection key)
    PRIV           (special WLM handling)
    NOCANCEL       (operator cannot cancel)

```

```

        SYST                (no timing)
        NOSWAP              (never swap out completely)
PPT PGMNAME(MVPXVMCF)
        KEY(0)              (storage protection key)
        PRIV                (special WLM handling)
        NOCANCEL            (operator cannot cancel)
        SYST                (no timing)
        NOSWAP              (never swap out completely)
PPT PGMNAME(EZBTCPIP)
        KEY(6)              (storage protection key)
        PRIV                (special WLM handling)
        NOCANCEL            (operator cannot cancel)
        SYST                (no timing)
        NOSWAP              (never swap out completely)
        SPREF               (preferred storage)
        LPREF               (preferred storage)
PPT PGMNAME(EZAPPFS)
        KEY(1)
        NOSWAP
PPT PGMNAME(EZAPPAAA)
        NOSWAP

```

The active PARMLIB(IEFSSNxx) member should include these two lines:

```

SUBSYS SUBNAME(TNF)
SUBSYS SUBNAME(VMCF)

```

You may find IEFSSNxx data in an older format, without keywords such as SUBNAME. Either format can be used.

Member EZAZSSI should exist in SYS1.PROCLIB. It normally is like this:

```

//EZZAZSSI PROC P=' '
//STARTVT EXEC PGM=EZZAZSSI,PARM=&P

```

This procedure (started task) must be started before TCP/IP can be started. This is normally done with a command such as S EZAZSSI,P=P390 in a COMMNDxx member of PARMLIB.⁶ The AD CD-ROM systems are set up this way. If you change COMMNDxx processing you must be certain that EZAZSSI is started before the start command for TCPIP. The program parameter (set to P390) in the AD CD-ROM systems is the *NJE name* of your system, if one exists, or the SID from PARMLIB.

6.22.12 Advanced TCP/IP customization

Obtaining TN3270 and ftp service can be considered part of basic TCP/IP customization. Additional services, such as dynamic routing, nfs, name servers, and so forth involves more advanced customization and is generally beyond the scope of this document. The following comments may be useful as a starting point for additional functions for your OS/390 TCP/IP.

Dynamic routing

OS/390 includes three dynamic routing programs: OROUTED, NCPROUTE, and OMPROUTE. See the full OS/390 TCP/IP documentation for operational details and full configuration information.

To use dynamic routing, you *must* disable static routing by removing (or commenting out) all the GATEWAY statements (or BEGINROUTES statements) in your PROFILE data set. To use OROUTED, you need to enable BSDROUTINGPARMS. In our examples, this might be:

⁶ It could also be started from a VTAMAPPL script.

```

; GATEWAY
; 9      =   ETH1  1492      0.255.255.0  0.12.17.0
BSDROUTINGPARMS false
      ETH1  1492      0      255.255.255.0  0
ENDBSDROUTINGPARMS

```

The false operand means that a maximum size of 576 will be used for packets. (A *true* means that your specified maximum packet size will be used.) The first 0 in the line beginning ETH1 is the hop metric. This value is normally zero. The subnet mask should be whatever is appropriate for your local network. The last 0 is required unless you have a point-to-point (non-LAN) link.

To use OROUTED, you must run the OE ROUTEDD server on the same OS/390 system as your TCP/IP stack.

How to start using the network file system

OS/390 TCP/IP includes both an nfs client and nfs server. The OS/390 AD CD-ROM system provides initial customization for both elements. IBM distributes nfs functions as part of DFSMS services, and not as part of TCP/IP. The documentation for nfs is completely separate from the TCP/IP documentation, and different data sets are used for control parameters. The following manuals are important:

- ▶ *OS/390 Network File System Customization and Operation, SC26-7253*
- ▶ *Network File System User's Guide, SC26-7254*

In the AD system, the key control data set is MVS NFS.CNTL. This is a partitioned data set with two key members: NFSATTR and EXPORTS. The NFSATTR member contains all the controls for the nfs server. One of the controls (as set up in the AD system) selects the *exports* level of security. This means that any data set or file listed in the EXPORTS member can be accessed from external nfs clients without going through RACF controls. (nfs access, in UNIX environments, is partly controlled by UID and GID values. The meaning of UIDs and GIDs across multiple systems can be a complex topic; issues involved often place bounds on what can be reasonably exposed to nfs accesses. OS/390 nfs adds options to use RACF controls to force use of OS/390-defined UIDs and GIDs. These options are well designed and potentially permit wider use of nfs while retaining reasonable security.) The list of files and data sets in MVS NFS.CNTL(EXPORTS) is intended only as an example. You will need to revise it (or move to a different security option) for any serious use of nfs.

The AD system provides two started tasks, NFSS and NFSC (nfs server and nfs client). NFSS is started via VTAMAPPL. NFSC is started indirectly. (The author does not know exactly how it is started.) For reasons we did not explore, the NFSC started task does not appear in D A,L displays on the MVS master console, while NFSS does appear. Both appear in D A,ALL displays. We used the nfs server with OS/2 and AIX clients, and had no problems.

We strongly suggest you read the nfs manuals listed above if you plan to use either client or server functions.

6.22.13 How to begin diagnosing TCP/IP problems

A structured approach to problem solving is important when dealing with TCP/IP problems on P/390 or MP3000 systems. The most common situation is that:

- ▶ OS/390 is installed and working.
- ▶ The MVS master console and local VTAM (TSO) sessions are working on the server, via the AWS3274 or LAN3274 emulated I/O device manager.
- ▶ OS/390 TCP/IP is not working.

This set of symptoms means that the server (OS/2) TCP/IP is correctly configured—at least for the loopback address.

Do you have the most recent P/390 (or R/390) device managers? This is the starting point for debugging, especially with P/390 systems. Updates and fixes are available at the p390.ibm.com ftp site.

If you have a LAN attached to your server TCP/IP (OS/2), does it work? Can you ping it from OS/2? (This would be address 9.12.17.150 in our examples.) If not, you have problems at the server level. For a P/390 or MP3000, check C:\IBMCOM\LANTRAN.LOG to see if the LAN adapters are being started. If there are errors here, check your MPTS configuration. If there are no LANTRAN errors, and the adapters are starting, check OS/2 TCP/IP definitions.

Check \P390\AWS3172.ERR for messages. This file is more difficult to interpret, but it may contain useful clues about why the device managers could not access the LAN.

If the problem appears to be only with OS/390 TCP/IP, use the MVS master console to display the unit status. (We assume the unit addresses are E20 and E21 for the rest of this section.)

```
D U,,,E20,2          (Use your correct addresses, of course)
```

This should display:

```
IEE457I 11:04:42 UNIT STATUS 700
UNIT TYPE STATUS  VOLSER  VOLSTATE
OE20 CTC  0
OE21 CTC  0
```

If the status is O (online) or A (allocated) or A-BSY (allocated and busy), your problem is probably with OS/390 TCP/IP definitions, or with your network (such as routing problems).

If the status is OFFLINE, try to vary the units online:

```
V (E20-E21),ONLINE
```

If the result is NO PHYSICAL PATH or a similar message, your problems are probably with MPTS, server TCP/IP, HCD, or the P/390 Configurator. In this case:

- ▶ Use MPTS to confirm that the adapter you want to use for OS/390 TCP/IP *does not* have TCP/IP associated with it in the MPTS Current Configuration window. It should have only IEEE802.2. Note the adapter number before the IEEE802.2 line.
- ▶ Review C:\IBMCOM\LANTRAN.LOG, looking for any obvious error messages associated with this line. There should be messages indicating the adapter started.
- ▶ Have you used the right S/390 address? Does MVS think this address is a CTC? (This should be indicated under TYPE in the DISPLAY UNITS output.) If not, you will need to use an address that MVS knows as CTCs (even/odd pair) or change MVS, using the HCD function.

Start the P/390 Configurator. Be certain the appropriate addresses are defined. The even/odd pair of S/390 addresses you want to use must be defined to the LCS3172 device manager.

Are these addresses defined as CTC units (or equivalent) in your OS/390 system? This can be verified by the D U,,,E20,2 command shown earlier.

If OS/390 indicates that the addresses are ONLINE, but not allocated, try stopping and starting OS/390 TCP/IP.

```
P TCP/IP
```

(wait for ENDED message)
S TCPIP

(This assumes your started procedure name is TCPIP.) After TCPIP starts, the D U,,,E20,2 message should change from O (online) to A-BSY (allocated) for E20 and A (allocated) for E21. (You can also use NETSTAT DEV and NETSTAT GATE for this information.) If the addresses remain O (online), you probably have a problem with the DEVICE, LINK, or START lines in your PROFILE data set. Be certain the correct adapter number (from the IEEE802.3 line in the MPTS Current Configuration window) is specified as the third operand of the LINK statement in your PROFILE data set.

Do you have the right adapter selected in MPTS? The names in the Network Adapter window of MPTS can be quite similar and may not be easy to relate to the adapter you think you have. Can you verify which adapter you have? By finding the packing box, for example?

If you are using **ping** to test your OS/390 TCP/IP, you should confine your test (both sides of the **ping**) to the *local LAN connected to your OS/390 TCP/IP system*. If you try to **ping** to a more remote system, the ping may fail due to routing problems *in the return direction*. There must be defined routes *from* your system (using GATEWAY statements, for example) and *to* your system (and these must be defined in the remote system and any intermediate routers). *Always do your initial pinging to a system directly on your local LAN, in the same subnet.*

There is no point in trying TN3270, telnet, or ftp until you have **ping** working. Again, we stress perform **ping** tests with another system connected directly to *your local subnet*. If this works, but remote pings fail, there is a routing problem with intermediate or remote nodes.

If basic functions work, such as **ping** and TN3270 logon, but there are erratic errors, check your MTU size. For Ethernet, start with 1492, not the default 1500. If nothing works, verify your DIX or IEEE802.3 selection in MPTS. If you are in a complete IBM environment, you may be in an IEEE802.3 environment. Otherwise, you are probably in a DIX environment and your system should be configured for it. Note that OS/390 can have it both ways by using the ETHERNETOR802.3 parameter in the PROFILE data set. However, your server low-level adapter routine (OS/2) does not support this dual operation and should be set to the correct protocol.

Verify that your MTU size for Ethernet is not larger than 1492. A few users have reported that they needed to set the size to something in the 1450 range; we assume this was due to intermediate node limitations in their network. (If you think your problems are in this area, try setting the MTU to 576; all IP nodes should handle this size correctly.) MTU sizes are set in the GATEWAY statements of the OS/390 PROFILE data set or in the OS/2 TCP/IP configurator (for OS/2 TCP/IP).

Useful commands for TCP/IP debugging are:

- ▶ **ping** from a TSO command line
- ▶ **oping** from an OMVS command line
- ▶ **oe ping** from an OMVS command line (more options)
- ▶ **tracerte** from a TSO command line
- ▶ **otracer** from an OMVS command line
- ▶ **netstat** from a TSO command line
- ▶ **onetstat** from an OMVS command line
- ▶ D TCPIP,,N,xxxx where xxxx is a netstat-type option, from the MVS master console.

These commands offer help information; the new MVS operator commands are described in the IP Configuration document.

If you are unable to connect to **oteln** (the direct Open Edition ASCII telnet connection, at port 1023 in the AD system), the problem may be understanding your telnet client. By default, telnet clients attempt to connect to port 23. (Connecting to port 23 in ASCII telnet mode will attempt to use TSO in line mode. It is very unlikely that you really want to do this.) To connect to oteln, you need to specify a port number when starting your telnet client. The method of doing this varies with different telnet client packages. If you are starting your client with a line command, it might be something like this (using port 1023 as the example):

```
telnet -p 1023 9.12.17.151      or
telnet 9.12.17.151 1023       or
telnet 9.12.17.151:1023
```

or something similar.

6.23 How to compile TCP/IP applications

We used the following JCL to compile simple TCP/IP client and server applications:

```
//OGDEN7 JOB 1,OGDEN,MSGCLASS=X
// EXEC EDCCL,PARM.COMPILE='SOURCE'
//COMPILE.STEPLIB DD DISP=SHR,DSN=CEE.SCEERUN
//              DD DISP=SHR,DSN=CBC.SCBCCMP
//COMPILE.SYSLIB DD DISP=SHR,DSN=CEE.SCEEH.H
//              DD DISP=SHR,DSN=CEE.SCEEH.ARPA.H
//              DD DISP=SHR,DSN=CEE.SCEEH.NET.H
//              DD DISP=SHR,DSN=CEE.SCEEH.NETINET.H
//              DD DISP=SHR,DSN=CEE.SCEEH.SYS.H
//COMPILE.SYSIN DD *,DLM='<>'
               source code
<>
//LKED.SYSLIB   DD DISP=SHR,DSN=CEE.SCEELKEX
//              DD DISP=SHR,DSN=CEE.SCEELKED
//LKED.SYSLMOD DD DISP=OLD,DSN=OGDEN.LIB.LOAD(PSERVER)
```

Using this JCL, we did not have any private INCLUDE libraries. You would need to add such libraries to the SYSLIB concatenation. And the following JCL was later used to execute the programs in MVS batch:

```
: //OGDEN8 JOB 1,OGDEN,MSGCLASS=X
// EXEC PGM=PSERVER,PARM='POSIX(ON)'/
//STEPLIB DD DISP=SHR,DSN=OGDEN.LIB.LOAD(PSERVER)
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
//CEESNAP DD SYSOUT=*
```

All the SYSOUT statements may be overkill, but this is a good starting point for execution.

Under OMVS, we used the same programs (without the JCL, of course) and compiled with a simple **cc** command.

6.24 How to set FTP data set parameters

If you use ftp to send a file to MVS, it is better to use a preallocated target data set, if possible. In this way, the disk allocation and DCB parameters are already established. If your target data set does not exist, the ftp server will use its default parameters. In the AD system, these are set in TCPIP.FTP.DATA; other systems may have the parameters stored in a different data set.

The default parameters may not be what you want. In the AD system, the default parameters are SPACE=(1,1),TRK, with BLKSIZE=6233, RECFM=VB, and LRECL=256. You can modify these default parameters for the duration of your ftp session. Use one of the following formats in your ftp client session, before you transfer the file to MVS:

```
site RECFM=FB LRECL=80 BLKSIZE=0 PRIMARY=10 SECONDARY=5
or quote site RECFM=FB LRECL=80 etc....
```

Some ftp clients will recognize the **site** command and others may not. If the site command is not accepted, try placing **quote** before it. It appears most clients will accept the **quote** command; the OS/390 ftp server will then process the site command contained in the quote record. The **quote site** words may need to be in lower case.

If your ftp client appears to connect to the OS/390 ftp server, but you are unable to obtain responses from DIR, GET, MGET, or similar commands, the problem may be a SOCKS firewall between the client and server. In this case you may need an ftp client that can perform *passive* connections. Most newer ftp clients have this ability, but you may need to enable passive mode.

6.25 How to FTP from a batch job

You can run the FTP client from an MVS batch job. You will need the userid and password that are required by the target system's ftp server. The job might look like this:

```
//OGDENF JOB 1,OGDEN,MSGCLASS=X
// EXEC PGM=FTP,REGION=4096K
//OUTPUT DD SYSOUT=*
//INPUT DD *
10.1.1.151 (or use a domain name)
p390a (a userid)
p390apw (the password)
CD /u/p390a (directory on remote system)
dir (for my listing, if there is a problem)
get remotefile
quit
/*
```

You would not enter the comments shown in parentheses, of course. Such usage is not limited to OpenEdition files; you can access MVS data sets the same way. We suggest you use FTP interactively to determine the commands you need, and then create a batch job with the same commands. (This assumes you have a need to perform the same FTP function frequently.)

If the INPUT and OUTPUT DD statement does not use spooled files, the INPUT LRECL must be 80 and the OUTPUT LRECL must be 160. More examples are given in *OS/390 IBM Communications Server IP User's Guide*, GC31-8514.

6.26 How to establish TN3270E connections

TN3270E operation is an extended function within TN3270 protocols. It has several functions, but a primary one is the ability to connect to a specific VTAM LU. The TN3270E function is initiated by the client, not the server. If you use the PCOM client, you can use the following setup (from a PCOM 3270 session):

```
Click on COMMUNICATION
Click on CONFIGURE
  On Select Connection, click on CONFIGURE
    On Customize Configuration, click on CONFIGURE LINK
      Click on ADVANCED
        Specify an LU name
      Exit
    Exit
  Exit
Exit. The session will restart.
```

The LU name you specify must be defined in the BEGINVTAM parameters of the OS/390 TCP/IP PROFILE, and must not be in use at the time you establish the session. Unless there is something special about selected LU definitions, or unless you are using LU names as part of your security controls, there is little purpose in using TN3270E to connect to a specific LU.

From the server side, in order to use the TN3270E protocol you must have a second *logmode* specified in the TELNETDEVICE statements in your OS/390 TCP/IP PROFILE. (This is TCPIP.PROFILE.TCPIP for AD systems.) For example:

```
TELNETDEVICE 3278-3-E NSX32703, SNX32703 ; comments....
```

says that, if the PCOM connection claims to be device 3278-3-E then logmode table NSX32703 will be used for TN3270 mode and SNX32703 will be used for TN3270E mode. Logmodes for TN3270E connections should be SNA logmodes and their names begin with S; non-SNA logmodes begin with N. See the *IP Configuration* manual for more information.

Note that the E in the terminal type name (3278-3-E) has nothing to do with TN3270E mode. The E in the name indicates extended attributes for the terminal, and indicates that the application should query the screen size.

Also note that, strictly speaking, TN3270/TN3270E are concerned with the connection protocol and the session is concerned with screen size and attribute byte capabilities. There is a fine shade of difference here that can make discussions/descriptions even more confusing than they already are.

You can use the MVS command:

```
D TCPIP,,TELNET,CONNECTION,LUNAME=xxxx
```

to verify the protocol and logmode of a specific connection.

6.27 How to have several TN3270 ports

Starting with OS/390 V2R6, the TN3270 server that is part of OS/390 TCP/IP can listen on several ports. You can specify this by adding several TELENTPARMS sections to your PROFILE for TCP/IP. For example,

```
...
TELENTPARMS
  PORT 23
  INACTIVE 600
```

```

ENDTELNETPARMS
TELNETPARMS
  PORT 2323
  ...
ENDTELNETPARMS
...

```

You can also have several BEGINVTAM ... ENDVTAM blocks. See the IP Configuration manual (SC31-8726) for more details.

Multiple TN3270 ports can be useful for a variety of purposes, especially since you can start and stop individual port functions while TCP/IP is operational.

6.28 How to FTP partitioned data sets

MVS load modules are stored in partitioned data sets, or PDSs. (This discussion ignores details that differ with PDSEs.) Partitioned data sets are also commonly used for JCL and many MVS control files. A PDS is not a sequential file; you cannot send or receive a PDS using FTP, because FTP deals only with sequential files.

To send (or receive) a PDS, it must be converted to a sequential file. The IEBCOPY utility does this. However, IEBCOPY -- when left to its own defaults -- uses variable spanned records. These are fine for disk or tape storage, but present problems for FTP. (Some FTP implementations support MODE BLOCK operands that can be used in this situation.)

TSO has two commands, XMIT and RECEIVE, that can be used to convert a PDS to sequential format (XMIT) or convert the sequential form back to a PDS (RECEIVE). These commands use IEBCOPY internally, but store the sequential file as fixed-length records (LRECL=80). This is a good format for FTP.

The intended use of XMIT/RECEIVE is to send data over an NJE network. For this it uses a *node.userid* form of addressing. By the addition of a data set name parameter, XMIT/RECEIVE can work without actually sending the data anywhere via NJE. In this case, we must still supply a *node.userid* address, but it is just a dummy address.

When dealing with MVS data sets, usage might be (using the TSO READY prompt or ISPF option 6):

```

XMIT node.userid OUTDSN('MY.HOLDING.FILE') DA('MY.PDS.TO.SEND')

FTP .....          (start FTP, with node name, userid, password)
binary
put 'MY.HOLDING.FILE' remote.file.name
end
(On the remote system)
RECEIVE node.userid INDSN('REMOTE.FILE.NAME')

```

You can enter the *node.userid* operands exactly as shown here; they are meaningless, but must be in the right format. The OUTDSN specifies a sequential file. In this example, the XMIT command will convert the PDS MY.PDS.TO.SEND to sequential form and store this in MY.HOLDING.FILE. If this data set does not exist, XMIT will create it with LRECL=80 BLKSIZE=3120 RECFM=FB.

You can FTP this file in BINARY mode. The smoothest operation occurs if the target data set is preallocated with LRECL=80 RECFM=FB BLKSIZE=3120. You could also specify these parameters with the SITE subcommand of FTP. The FTP server at the target location might be configured to default to these DCB parameters for new files, but you cannot depend on this.

At the receiving location, use the RECEIVE command to convert the sequential file back to a PDS. It will be converted to the same name as the original data set if you do not specify another name (via a DA(...) parameter).

6.29 Why some browsers do not work with FTP

A web page may contain an ftp link to a downloadable file. In some cases this appears not to work if the ftp server is OS/390. Whether or not it works is related to the particular browser version.

For example, the link might be:

```
<a href="ftp://www.favorite.com/os390/oe/port/starwars"
```

Standards (RFC 1738) state that the browser should issue these commands (once within ftp):

```
cwd os390
cwd oe
etc
```

However, several current browsers issue this command (in specific violation of the RFC):

```
cwd /os390/oe
etc
```

The leading slash causes the problem, because it indicates that the path names starts from the root directory. If the OS/390 ftp server is configured to start all paths from the root directory, this second version will work. If the ftp server is configured to start clients at a preselected spot in the directory tree (and the URL was coded with this in mind) then forcing the path to start from the root may not find the desired file.

The solution is to make all potential ftp files located relative to the root directory. This has security exposures, since only your permission bits will stand between other HFS files (not intended for ftp usage) and the outside world. (A more general solution is to use the chroot() function as part of the ftp server startup. OS/390 does not provide convenient way to implement this.)

6.30 How to use nonstandard 3270 screen sizes

This is an interesting topic and there are clearly a variety of ways for defining and using non-standard screen sizes. We report here what we used on our P/390, using PCOM for TN3270 connections. Your experience may vary.

Standard 3270 screen sizes are:

Model	lines x columns	
2	24 x 80	
3	32 x 80	<-- most commonly used size
4	43 x 80	
5	27 x 132	

3270 displays usually initialize in 24x80 mode and later software (such as ISPF) switches it to its *alternate* size. Some emulators are always in their *alternate* (larger) size. For our testing, we decided to use a screen size of 37x147 and this is what was used for the steps described below. (This size was quite arbitrary.)

We should note that this description does not address SNA connections, even though VTAM definitions are described at some length. By SNA, we mean connections using the LAN3172 device manager for a P/390 or MP3000. We address two forms of TN3270 connections here:

- ▶ TN3270 connections to LAN3274 (or AWS3274). These connections are through OS/2 TCP/IP and appear to MVS VTAM as local, non-SNA 3270s -- as if connected through coax to a local 3174 control unit.
- ▶ TN3270 connections through LCS3174 to OS/390 TCP/IP. These use the OS/390 TCP/IP stack and are affected by the VTAM parameters in the TCP/IP PROFILE data set.

In both cases, we used PCOM for the 3270 emulators. We could not define a non-standard screen size using the PCOM configuration panels. Instead, we used an OS/2 editor to change the WS file for the PCOM session. In the our OS/2 system this was C:\PCOMOS2\PRIVATE\C.WS (for our C session). Find the stanza with the heading [3270]. Within it is a line such as "ScreenSize=32x80". Change the size to whatever you wish to try. After doing this, *do not use the PCOM configurator again for that session*. The configurator apparently does not like non-standard screen sizes and automatically changes them to a standard size.

This was all that was needed for LAN3274 connections, via OS/2 TCP/IP. It worked because the VTAMLST member that defined the terminals contained DLOGMOD=DYNAMICB. (For the AD system, this is SYS1.LOCAL.VTAMLST(NSNA90X) and (NSNA70X).) We do not claim to understand exactly what DYNAMICB contains, but it worked for the terminal sizes we tried.

Understanding how ISPF deals with screen sizes is important. Most of the standard ISPF menus are used only in 80-column mode. You will see only 80 columns, unless you use ISPF Option 0 to change the screen format to MAX. (We recommend leaving it as DATA; you can try it both ways, of course.) You will not see a wider screen (assuming your non-standard size has a wider screen) until you browse or edit a data set that has records longer than 80 bytes. The screen will then change to the wider view. SDSF and OMVS also use the wider view.

In order to use non-standard sizes with OS/390 TCP/IP, we edited the TCP/IP PROFILE data set (TCPIP.PROFILE.TCPIP for the AD system) and changed the BEGINVTAM section to appear as follows:

```
BEGINVTAM
  TELNETDEVICE 3278-2-E D4A32XX3, D4A32XX3 ; for dynamic sizes
  TELNETDEVICE 3279-2-E D4A32XX3, D4A32XX3 ; for dynamic sizes
  TELNETDEVICE 3278-3-E NSX32703           ; 32 x 80
  TELNETDEVICE 3279-3-E NSX32703           ; 32 x 80
  TELNETDEVICE 3278-4-E NSX32703           ; 43 x 80
  TELNETDEVICE 3279-4-E NSX32703           ; 43 x 80
  TELNETDEVICE 3278-5-E NSX32703           ; 27 x 132
  TELNETDEVICE 3279-5-E NSX32703           ; 27 x 132
DEFAULTLUS
  unchanged .....
```

In order to edit the TCP/IP PROFILE, you will need to stop TCP/IP. The primary change was to use the D4A32XX3 mode table entry for both TN3270 and TN3270E connections for 3278-2 (and 3279-2, although we might not need this one) devices. Apparently odd-size terminals claim to be *model two* units when negotiating a VTAM bind.

We also used displays (TN3270 sessions on IBM Network Stations) that needed the explicit model definitions for the standard sizes, so we left those in the modetable list.

We do not know what reasonable limits exist for workable screen sizes. TSO and ISPF appear able to adapt to reasonable sizes. Other applications, such as a CICS transaction that closely manipulate 3270 controls, might not work.

6.31 How to set up a test TCP/IP

This section discusses running different TCP/IP procedures, one at a time. It does not discuss running multiple OS/390 TCP/IP *stacks* at the same time. Setting up a multiple stack environment is beyond the scope of this document and is described in detail in the *IP Configuration* document.

Testing TCP/IP is a term that covers many possibilities. In this case, we will consider the simple case of testing an alternate PROFILE. That is, you will use all your current TCP/IP parameter data sets except for PROFILE.

There are a number of considerations here:

- ▶ The name of the started procedure (usually *TCPIP*) is specified in BPXPRMxx. This means you cannot simply start another TCP/IP procedure with a different name.
- ▶ The naming conventions used to find the various TCP/IP parameter and configuration files are complex. You should not disturb them without good cause.
- ▶ It is possible to run multiple TCP/IP address spaces. However this is more complex to set up and we suggest you avoid this unless it is necessary in your situation. (OS/390 systems before V2R5 typically ran two TCP/IP stacks because the same TCP/IP was not used for both MVS and OE. Starting with OS/390 V2R5, a single TCP/IP can service both MVS and OE programs.)

If you simply want to test changes in the PROFILE (and this is the most common point for changes), you can proceed as follows:

- ▶ Copy your existing PROFILE to another data set (or another member, if it is a PDS). For the AD system, the distributed one is a sequential data set named TCPIP.PROFILE.TCPIP. Keep the same HLQ; you might create TCPIP.PROFILE2.TCPIP, for example.
- ▶ Edit your new PROFILE2, making the desired changes.
- ▶ Create a new member in SYS1.PROCLIB named, for example, TCPIP2. Copy the existing TCPIP member and change the DSN name in the PROFILE DD statement to point to your new profile.
- ▶ You will probably need to add TCPIP2 as a new RACF userid (or add an entry to your RACF STARTED class to cause started task TCPIP2 to be run under userid TCPIP.) The userid should have an OMVS segment with UID=0 and belong to a group with a GID.
- ▶ Stop the active TCPIP. The MVS command is **P TCPIP**.
- ▶ Start your test TCPIP with the command:

```
S TCPIP2.TCPIP
```

This format will pass the name TCPIP (instead of TCPIP2) through the BPXPRMxx specifications. This is necessary because the name TCPIP2 would not be recognized in BPXPRMxx.

- ▶ Perform whatever tests are necessary. When finished, stop TCPIP2 and start TCPIP.

Using this technique avoids making changes to BPXPRMxx. If you will be using several TCP/IP started procedures (one at a time), and if you do not want to depend on a special form of the MVS start command, you can change BPXPRMxx to include multiple started task names. For example:

```

.....
FILESYSTYPE TYPE(INET) ENTRYPPOINT(EZBPFINI)
SUBFILESYSTYPE NAME(TCPIP) TYPE(INET) ENTRYPPOINT(EZBPFINI)
SUBFILESYSTYPE NAME(TCPIP2) TYPE(INET) ENTRYPPOINT(EZBPFINI)
SUBFILESYSTYPE NAME(TCPIP3) TYPE(INET) ENTRYPPOINT(EZBPFINI)
NETWORK DOMAIN(AF_INET)
etc
etc

```

In this example, you could use started task names TCPIP, TCPIP2, or TCPIP3 to start TCP/IP. (Be certain to find the BPXPRMxx member in your PARMLIBs that is actually being used.)

A third method, if your PROFILE is a member in a PDS, is to make the member name (in the DD statement in the TCPIP started procedure) a symbolic parameter and pass the parameter in the start command. For example, change SYS1.PROCLIB(TCPIP) like this:

```

//TCPIP PROC PROFILE='TCPIP' <--- default member name
// EXEC PGM=EZBTCPIP,PARM='CTRACE(CTIEZ00)'
// ...

//PROFILE DD DISP=SHR,DSN=SYS1.TCPPARMS(&PROFILE) <--- changed
// ...

```

and then you could S TCPIP,PROFILE=TCPIP2 using whatever member name contains your PROFILE to be tested.

6.32 P/390, IBM Integrated Server

This chapter contains information unique to P/390-based systems. It does not apply to other, larger S/390 machines.

6.32.1 How to verify P/390 disk integrity

The device manager that emulates CKD disks, AWSCKD.EXE, uses a special (and undocumented) format in the OS/2 files it uses for emulated drives. While it rarely happens, these files can become corrupted. If this happens, the AWSCKD.EXE program may abnormally terminate (producing an OS/2 pop-up message) and MVS will then crash (because it can no longer access disks).

The CKDCHECK program can verify the format of existing CKD disks. This should be done when MVS (or any S/390 operating system) is not active. This program can simply look for errors (and report them), or it can reformat any invalid tracks that are found. The /F operand enables the reformatting function. To use this function, do the following:

- ▶ Note the full OS/2 names of each of your emulated CKD volumes, such as E:\OS390\WORK02.A91.
- ▶ Using an OS/2 window, change to your P390 directory.
- ▶ Using this OS/2 window, issue the line command:

```
CKDCHECK D:\OS390\WORK02.A91 /F (for example)
```

- ▶ Wait while the program reads the file. This may take some time because these files are quite large. On our Server 500 test system, it processed approximately 500 cylinders (3390) per minute.
- ▶ If no errors are found, the program will display a message and end.
- ▶ The messages are also appended to the CKDCHECK.LOG file, which is created in the current directory if it does not already exist.

If errors are found, and if the /F operand was specified, CKDCHECK will reformat the CKD track. This prevents the CKD device manager (AWSCKD.EXE) from crashing sometime later, while OS/390 is running, as is likely if it attempts to process an invalid format. If AWSCKD crashes while OS/390 is running, OS/390 will also crash (because all its emulated disks have effectively disappeared).

OS/390 data may be lost when a track is reformatted. If the emulated track format is invalid, any data on it is effectively lost anyway, unless heroic (and undocumented and unsupported) recovery is attempted using OS/2 access to the file. Reformatting the track (with CKDCHECK) prevents AWSCKD crashes (invariably followed by an MVS crash). If the bad track was in a critical OS/390 data set, you must restore the data set or volume from a backup tape. If it was not in a critical system data set, there are several possibilities:

- ▶ The track was in unallocated space on an emulated disk. In this case there is no problem.
- ▶ The track was in a user data set. In this case, an application program (which might be the ISPF editor, for example) will either ABEND or provide an error message when it attempts to read the data. You will need to recreate the data in some way. If the bad track is in the middle of a PDS, you will be able to use all the other members of the PDS; you should immediately copy the PDS (member-by-member, not a bulk copy) to a new PDS, skipping the damaged member.

You can use the IEHLIST utility to determine which data set was damaged when a track was reformatted by CKDCHECK. (This assumes that OS/390 is sufficiently operational to IPL and run a job.) If the damaged volume was TSOPAK, for example, run a job like this:

```
//OGDENL JOB 1,OGDEN,MSGCLASS=X
//          EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=*
//DISK1   DD VOL=SER=TSOPAK,DISP=OLD,UNIT=3390
//SYSIN   DD *
          LISTVTOC VOL=3390=TSOPAK,FORMAT
/*
```

This provides a rather messy listing of all the extents of all the data sets on the volume. The error message from CKDCHECK provides a CCHH (cylinder-head) address of the track-in-error. This is a two-part number, such as 603 12 (cylinder 603, head 12); the IEHLIST output provides similar information for all the data set extents on the volume. You need to search the listing to find which data set (if any) has an extent that includes the reformatted track. That data set is no longer valid. If the reformatted track does not appear in any extent in the IEHLIST output, then it is in unallocated space, and no action is needed.

Searching IEHLIST output for a specific track address is tedious, at best, because there is no option to obtain a listing sorted by track address. However, CKD format errors are very rare and this procedure should seldom be needed.

The CKDCHECK function is also useful for emulated volumes on an MP3000 system.

6.32.2 Hardware and server setup for TCP/IP

The following discussion references P/390 machines. Much of the material also applies to MP3000 systems. A separate redbook, *Multiprise 3000: Basic Emulated I/O Definitions*, SG24-5669 should be used for more specific MP3000 details.

TCP/IP setup for a P/390 or Integrated Server is complicated by the fact that the server side of the system also uses TCP/IP. For a P/390, for example, OS/2 TCP/IP is used to provide emulated local 3270 paths to OS/390. This path is normally used by the MVS master console and can be used for local TSO, CICS, and so forth. The TCP/IP used by OS/2 is independent of the TCP/IP that is used by OS/390. Furthermore, the LAN adapter(s) used by the server TCP/IP (OS/2) cannot be used for OS/390 TCP/IP.

Each of the platforms discussed here will normally have two (or more) LAN adapters; one for the server (OS/2) and one for OS/390. You must clearly distinguish between LAN connections for the server TCP/IP and LAN connections for OS/390 TCP/IP. Failure to understand this separation is a common cause of confusion when setting up OS/390 TCP/IP on these platforms.

It is possible to define a server TCP/IP network that does not use a LAN. That is, it uses only the internal loopback address. (It might also use the serial ports for SLIP or PPP connections, but this is not addressed here.) This is sufficient for supporting an MVS master console and several TSO windows on the attached server display. In this situation, two TCP/IP stacks (OS/2 and OS/390) could be used with only one LAN adapter, with the LAN adapter belonging to OS/390 TCP/IP. We believe that most P/390 and Integrated Server owners will elect to install two (or more) LAN adapters, and the following descriptions assume two or more adapters.

Many types of TCP/IP links exist: standard and fast Ethernet (DIX or IEEE802.3), 4/16/100 Mbps token ring, FDDI, channel-to-channel, SLIP, PPP, and so forth. Our discussions will concentrate on 10 Mbps DIX Ethernet, as this appears to be the most common LAN protocol.

Some IBM-configured P/390 systems have one Ethernet adapter (such as the integrated adapter in the Server 330) and one token ring adapter. This is not a useful mix in most situations. We suggest and assume that you have two LAN adapters of the same type --- either Ethernet or token ring.

6.32.3 How to prepare a P/390 for TCP/IP

Although the following description is for a P/390 system, the same concepts apply to MP3000 systems using emulated I/O for LAN connections.

The most common P/390 systems are based on IBM Server 330-PB0 machines. These have PCI busses. Earlier P/390 systems were based on Server 500 machines with Micro Channel busses. This discussion assumes you are using Server 330 machines. (The Server 500 did not have an integrated Ethernet adapter. You needed Micro Channel adapters for each of the Ethernet or token ring adapters you wish to install. As far as we know, any IBM Micro Channel Ethernet or token ring adapter that is supported by OS/2 for a Server 500 can be used for a Server 500-based P/390 system.)

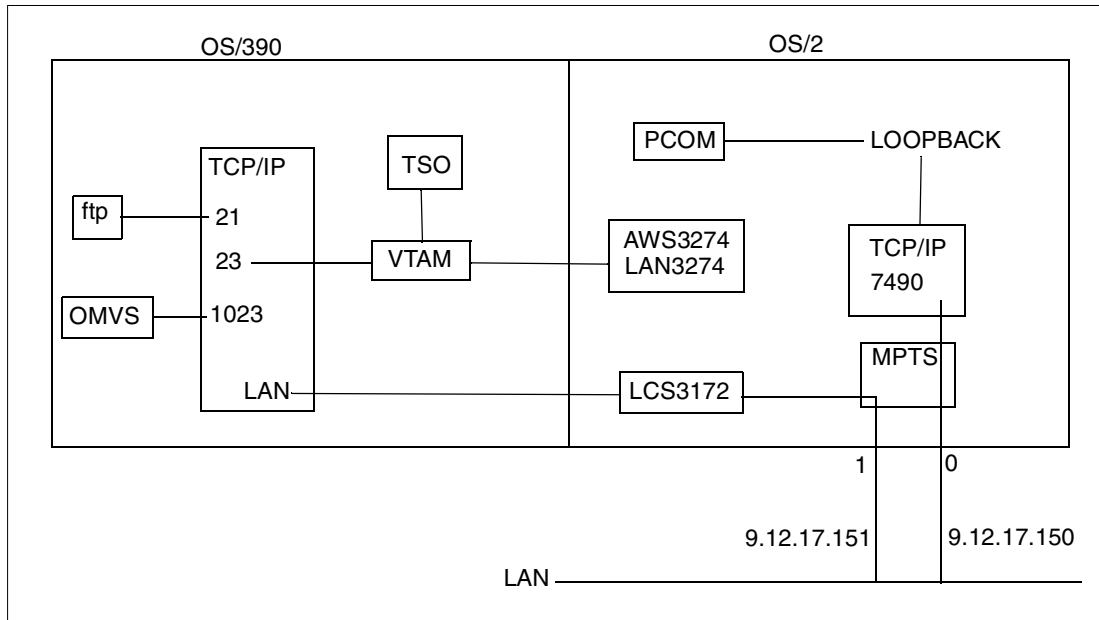


Figure 6-6 Goal for Basic P/390

The Server 330 has an integrated Ethernet adapter that can be used for either OS/2 TCP/IP or OS/390 TCP/IP. If you add another adapter, we strongly recommend the *IBM 100/10 EtherJet PCI Adapter*; this is available in several forms. We do not recommend the following:

- ▶ *IBM PCI Ethernet Adapter*. (This has connections for 10Base-T and AUI (15-pin connector)). We have had difficulties using this adapter with P/390 systems.
- ▶ Any ISA bus adapter. We have had difficulty configuring these if there is more than one ISA bus LAN adapter.

If you use token ring adapters, we recommend *IBM PCI Token-Ring Adapter*. We do not recommend *IBM Turbo 16/4 Token-Ring ISA Adapter*. Again, we have had problems configuring these if more than one ISA LAN adapter exists.

One of the P/390 systems used frequently for ITSO projects has one integrated Ethernet adapter, one PCI EtherJet adapter and two PCI Token-Ring Adapters. We have used these adapters in various combinations for OS/2 and OS/390 TCP/IP connections, without problems.

Setting up the P/390 Server for TCP/IP consists of:

- ▶ Installing the adapters
- ▶ Configuring the adapters with MPTS
- ▶ Configuring one or more adapters for OS/2 TCP/IP
- ▶ A minor edit for CONFIG.SYS
- ▶ Update the P/390 configuration (DEVMAP)
- ▶ Rebooting OS/2 and verifying the adapters started correctly
- ▶ Configure PCOM (or another 3270 emulator)

In Figure 6-6 on page 324 the P/390 system has two LAN adapters. Adapter 0 is connected to OS2 TCP/IP and the LAN3274 device manager. TN3270 sessions to port 7490 at 9.12.17.150 will appear to be local 3270 displays to OS/390. TN3270 sessions to port 23 at 9.12.17.151 will connect to OS/390 TCP/IP through the INTCLIEN server that connects them to VTAM. Telnet sessions to port 1023 at address 9.12.17.151 will connect directly to OMVS (as ASCII terminals). The PCOM session used for the MVS master console (on the OS/2 side) connects to OS/2 TCP/IP via the loopback address.

Installing the adapters

The integrated Ethernet adapter is already installed in Server 330 systems. Other PCI adapters are installed simply by placing them in available PCI slots. Most adapters have their MAC address noted on a paper label somewhere on the adapter. We suggest you note which MAC address is located in which PC slot number. You may need this later to identify which adapter is for OS/2 and which is for OS/390.

Configuring adapters with MPTS

This can be the most confusing step of this process. We suggest you step through it carefully and write down the final configuration, for future reference. Neither your server TCP/IP nor your OS/390 TCP/IP will work unless you have the correct definitions in MPTS.

Start with an OS/2 window and enter the command MPTS. (This stands for Multi-Protocol Transport Support; in earlier versions of OS/2 it was known as LAPS.) On the first panel select **Configure**; on the second panel select **Configure LANs** and then select **Configure**. This should produce a panel similar to Figure 6-7. Note that there are three windows in this panel: Network Adapter, Protocols, and Current Configuration. The initial Current Configuration window will be empty. (All three windows can scroll up and down, and may contain many more lines of data than shown here.)

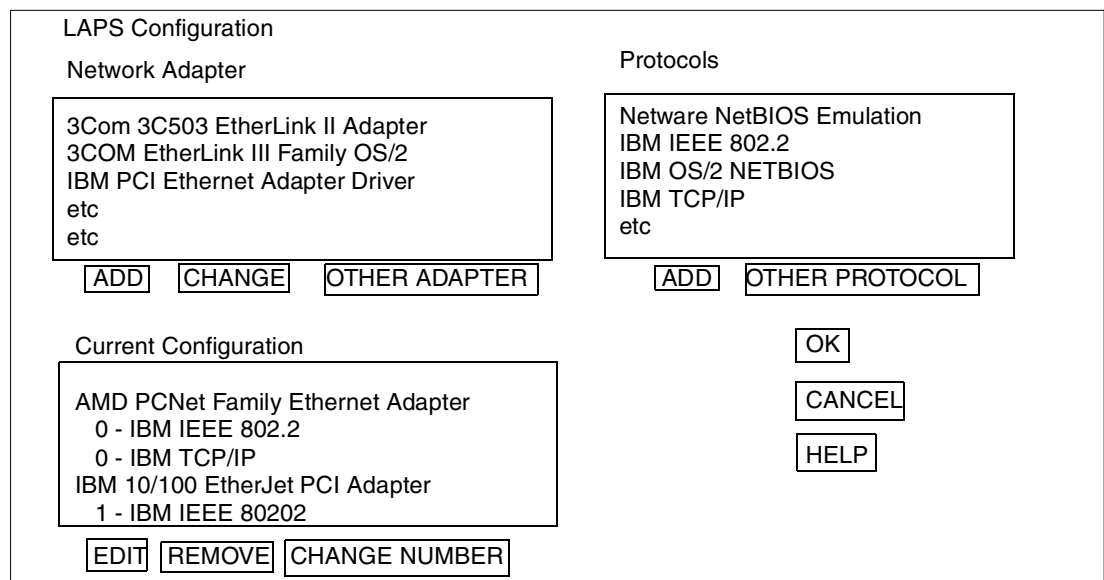


Figure 6-7 Primary MPTS Working panel

The general operation of this panel is:

- ▶ Select a Network Adapter that matches one of your adapters and ADD it to your Current Configuration.
- ▶ Then select the protocol(s) that should be associated with this adapter and ADD each to the Current Configuration (where the correct adapter is selected).

- ▶ Then select another Network Adapter that matches another of your adapters and ADD it and its intended protocols to your configuration. And so forth.

However, there are some details to consider before you start. Perhaps the most difficult detail is to find your adapter in the Network Adapter window. Many of the adapter names listed in the first window are quite similar (especially among all the IBM adapters included). Furthermore, your adapter may not be included in the list. For example, the integrated Ethernet adapter included with Server 330 systems is not in this list.

You should have received a diskette of OS/2 drivers with your LAN adapter (and with the Server 330). You need this to add your LAN adapter to the list in the Network Adapter window. (Even if your adapter is already in the list, we suggest you follow the process of adding the adapter in order to copy the latest driver from your diskette.) (We assume your diskette will have a later version of the driver than the base OS/2 system is likely to have.) If you have the diskette, select (double click with the mouse) the OTHER ADAPTER button and follow the instructions to insert your diskette in drive A. MPTS will search the diskette for appropriate drivers and setup information and add your LAN adapter to the list of adapters in the Network Adapter window.

Installing new adapters for MPTS is done only once. If you are reconfiguring your MPTS and the proper adapter drivers were previously installed, you do not need to install them again.

If your LAN adapter is already in the list somewhere in the Network Adapter window (remember to scroll through the whole list), then you do not need the diskette of drivers (unless you want to update the drivers). Read the Network Adapters list *carefully*. The names are confusing. If you select the wrong adapter your TCP/IP network will not work. (Fortunately, it is easy to reconfigure MPTS, so you can try again if you make a mistake.)

If you have a Server 330 P/390, we suggest you make the integrated Ethernet adapter the first adapter for MPTS. You will need the Ethernet diskette that came with the system to do this. Follow the process just described to add this Ethernet adapter to the MPTS list. When you are finished there should be a new line in the Network Adapters window:

AMD PCNet Family Ethernet Adapter

This is the proper name of the integrated Ethernet adapter on the Server 330-PB0 machine.

Next, with the proper adapter name highlighted in the Current Configuration window, move to the Protocol window. Select (single mouse click) the IBM IEEE 802.2 protocol, and then ADD it. This protocol must be included for every adapter intended for use by OS/390 TCP/IP; IEEE 802.2 is also selected for adapters intended for OS/390 SNA or OS/2 SNA.

If this adapter will be used for OS/2 TCP/IP, then select IBM TCP/IP in the Protocol window and ADD it. Do not ADD the TCP/IP protocol if the adapter will be used for OS/390 TCP/IP. Be careful! The TCP/IP in the MPTS window refers only to OS/2 TCP/IP. If you select this protocol for an adapter, that adapter cannot be used with OS/390 TCP/IP. (To be more precise, if OS/2 TCP/IP is started and this interface is active, the adapter cannot be used by OS/390 TCP/IP.)

After you have selected the protocols for your first LAN adapter, return to the Network Adapter window and select the line matching your second LAN adapter. ADD this line and then ADD the protocols for it.

Figure 6-7 on page 325 illustrates a system with two LAN adapters. The first is the integrated AMD Ethernet adapter and the second is an IBM PCI EtherJet adapter. The AMD adapter will be used for OS/2 TCP/IP and the EtherJet adapter will be used for OS/390 TCP/IP. (As an aside, note that in some cases either (or both) adapters can be used for SNA at the same

time they are being used for TCP/IP. SNA connections can potentially share adapters between OS/390 and OS/2 and can sometimes use adapters that are running TCP/IP at the same time. This is usually true for token ring adapters. It can be true for Ethernet adapters, but is usually not true for the most common Ethernet configurations)

In the Current Configuration window, note the number before the protocol name(s) for each adapter. You will need these numbers later for TCP/IP configurations in both OS/2 and OS/390. We suggest you print or write down the information in the Current Configuration window.

There are two slightly different Ethernet protocols in use today. These are DIX Ethernet and IEEE 802.3 ethernet. These different low-level protocols apply to all higher-level protocols that might be sent over Ethernet. For example, there you might have:

TCP/IP with DIX	(commonly used)
TCP/IP with IEEE802.3	(rarely used)
SNA with DIX	(rarely used)
SNA with IEEE802.3	(more commonly used)

Furthermore, the IEEE802.3 or DIX selection for an OS/2 Ethernet adapter may be specified in two places: in MPTS or in the OS/2 TCP/IP configurator. We will discuss the OS/2 TCP/IP configuration option later. The choice you set for the 802.2 protocol via MPTS is used by the LCS3172 device manager (for OS/390 TCP/IP) and the LAN3172 device manager (for OS/390 SNA, which we will not further discuss here). MPTS defaults to IEEE 802.3. If you have an existing (non-IBM) IP network and you are not certain which Ethernet form you use, you are probably using the DIX form. *If this is confusing to you, then select DIX for your first try.*

Do not confuse IEEE 802.2 (an internal frame format between various parts of a LAN protocol implementation) and IEEE 802.3 (an external packet format; an alternative to DIX).

To change your MPTS 802.2 Ethernet protocol to DIX:

- ▶ In the Current Configuration window, double-click the IEEE 802.2 protocol line associated with the appropriate adapter. For example:

```

AMD PCNet Family Ethernet Adapter
  0 - IBM IEEE 802.2          <---- double click this line
  0 - IBM TCP/IP
IBM 100/10 EtherJet PCI Adapter
  1 - IBM IEEE 802.2

```

- ▶ This should produce a large display (scrollable) titled *Parameters for IEEE 802.2*. This display contains lines such as:

```

* ..... (more options)

*network adapter address      ____

*Type ethernet driver         I      <-- find this line

*System key value             ____

* ..... (more options)

```

Find the indicated line and change the value from I to D.

- ▶ Do not change any other parameters in this panel unless you have a specific requirement and understand what you are changing.
- ▶ Click **OK** to exit.

- ▶ Repeat this change for any other adapter that is used with OS/2 TCP/IP. (Remember: an adapter is used with OS/2 TCP/IP if it has the *IBM TCP/IP* protocol assigned to it in the Current Configuration window and, in a later step, is activated by the OS/2 TCP/IP configurator.) We also suggest making the same change for the adapter intended for OS/390 TCP/IP.

There may be one other change in the LAPS Configuration if you are using token ring adapters that have automatic speed sensing (4 or 16 Mbps) or have full duplex capability. If you have autosensing token ring adapters AND if you are connecting to a token ring with no other active systems, then the autosense function can cause problems. It switches speeds every few seconds (looking for packets from other systems on the LAN, in order to select a permanent speed), and this causes OS/2 to pause at intervals. (A token ring adapter can also cause OS/2 to pause if the adapter is not connected to a ring (such as an IBM 8228 MAU) or does not have a loop plug at the adapter connector.)

There have been situations in which the use of full duplex token ring operation has been associated with problems. We do not have definitive information about this, but we usually disable full duplex operation in our small, lightly-loaded token rings used with our P/390 systems. (You may have different experience with full duplex in other situations, and want to ignore our suggestion to disable it. It requires a switching hub, so if you have this you may want to work with full duplex.)

To remove autosensing and/or full duplex:

- ▶ In the Current Configuration window, double click (with the mouse) on the title line (not a protocol line) for the token ring adapter.
- ▶ This produces a panel with many token ring options. Find the line named *PCI adapter data rate* and change this from AUTO to M16 (or M4 if you want 4 Mbps).
- ▶ If you wish, find the line named *Enable full-duplex* and change it from YES to NO.
- ▶ Do not change any other lines on this panel unless you have a specific requirement.

In general, the MPTS screens are easy to use once you understand the general operation. You can ADD and REMOVE adapters and protocols, in the Current Configuration window, until you are satisfied with the configuration. At this point select OK, and step through several screens to exit from MPTS. When you exit from the last screen, it will ask permission to update CONFIG.SYS. You should reply YES to this question.

6.32.4 Configuring selected adapters for OS/2 TCP/IP

Next you need to configure OS/2 TCP/IP. You should already have TCP/IP installed as part of your OS/2 installation. It should be represented as an icon on the primary OS/2 desktop. Select this icon (double click) and it should open to display icons for about 18 different TCP/IP options. One of these is TCP/IP CONFIGURATION. Select this icon (double-click) and it should produce a display similar to that in Figure 6-8 on page 329

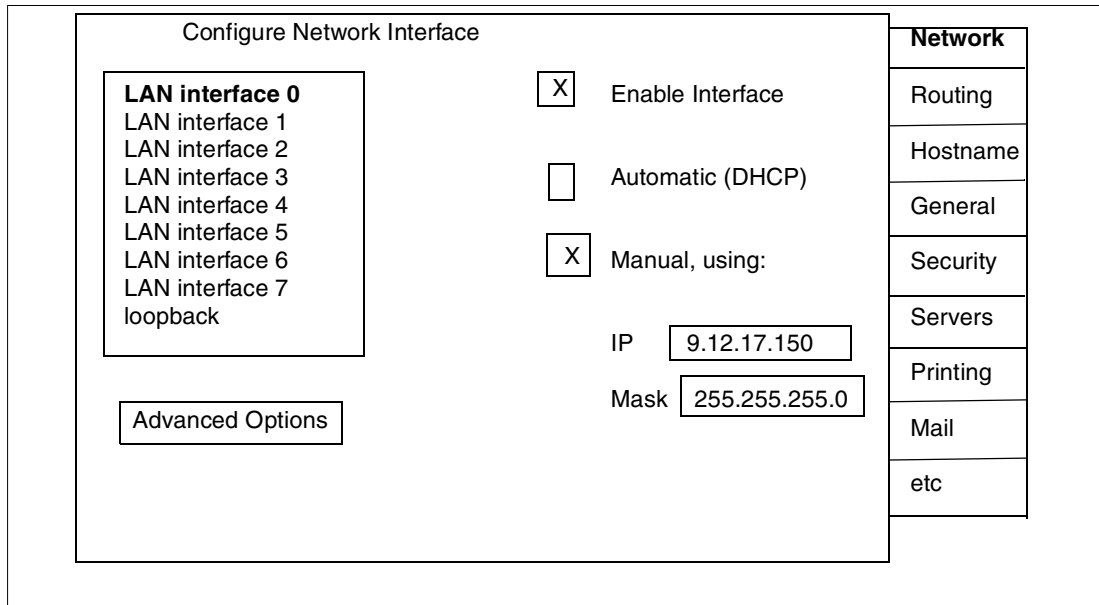


Figure 6-8 Primary Configuration panel for OS/2 TCP/IP

Later versions of OS/2 TCP/IP have a different format for these panels, but the basic operation remains about the same.

Assuming you will use the TCP/IP interface for AWS3274,⁷ you should first enable the loopback interface:

- ▶ Select (single click) the LOOPBACK INTERFACE line on the Configure Network Interface display.
- ▶ Select the ENABLE INTERFACE option.
- ▶ Select the MANUAL address option and enter 127.0.0.1 as the IP address. No entry is needed in the MASK field.

When you later reboot (or restart OS/2 TCP/IP) this will enable the loopback interface at address 127.0.0.1.

You should enable one or more LAN interfaces for OS/2 TCP/IP. To do this you must relate the TCP/IP interface numbers (as illustrated in Figure 6-8 on page 329) to the MPTS adapter numbers (as illustrated in Figure 6-7 on page 325). This can be confusing and is best explained with an example:

MPTS Adapter	OS/2 TCP/IP Interface

AMD PCNet Family Ethernet Adapter	
0 - IBM IEEE 802.2	
0 - IBM TCP/IP	Interface 0
IBM 100/10 EtherJet PCI Adapter	
1 - IBM IEEE 802.2	
IBM PCI Token-Ring Adapter	
2 - IBM IEEE 802.2	
2 - IBM TCP/IP	Interface 1
IBM PCI Token-Ring Adapter	
3 - IBM IEEE 802.2	

⁷ That is, assuming you are not using the obsolete CM/2 interface to your P/390 system.

In this example, four LAN adapters are defined to MPTS. The first two are Ethernet and the last two are token ring --- although the LAN types are irrelevant to the numbering scheme. The MPTS adapter numbers are 0 - 3; these numbers are used later by OS/390 TCP/IP and VTAM. The OS/2 TCP/IP interface numbers are assigned, starting with interface 0, only to MPTS adapters having the (OS/2) TCP/IP protocol associated with them.

The typical P/390 we are considering has only two LAN adapters and only the first is used for OS/2 TCP/IP. In this case the MPTS adapter number (zero) matches the TCP/IP interface number (zero). To enable this adapter for OS/2 TCP/IP:

- ▶ Select (single click) the LAN INTERFACE 0 line on the Configure Network Interface display.
- ▶ Select the ENABLE INTERFACE option.
- ▶ Select the MANUAL address option and enter your selected IP address. Make an appropriate entry in the MASK field, if necessary. The example address in the figure is a class A address, and we subnet it to a class C network; this requires a mask of 255.255.255.0.
- ▶ If this interface is Ethernet (IEEE or DIX), select the ADVANCED OPTIONS button. In the advanced options display, change the MTU size from 1500 to 1492. Do not make entries in the Broadcast Address or Destination fields in the Advanced Options display. Select OK to return to the Configure Network Interface display. (In the unlikely event that you want to use IEEE802.3 frames instead of DIX frames, you must select the NEXT button in the ADVANCED OPTIONS frame. This provides a second panel of advanced options, where you can select IEEE802.3.)

Repeat these steps if you have a second OS/2 TCP/IP adapter (which is unlikely unless you are connecting to both Ethernet and token ring networks).

The Configure Network Interface display has a number of *tabs* on the right-hand side. (Later versions of OS2 TCP/IP have a somewhat different layout, but the operation is about the same.) The labels are Network, Routing, Hostname, and so forth. You need to select these tabs, one at a time, and make entries for several of the functions. The most important of these is the Autostart function.

- ▶ Select the Autostart tab. This will display the Configure Automatic Starting of Services screen.
- ▶ Select *inetd* in the left-hand window of the Autostart screen. Then select *Autostart service* and *Foreground* on the right-hand side of the screen.
- ▶ If you want OS/2 telnet (not required for P/390 operation), select *telnetd* in the left-hand window and select *Autostart service* and *inetd super server daemon* on the right-hand side.
- ▶ Select *portmap* in the left-hand window and select only *Autostart service* on the right-hand side.
- ▶ You should not need *ftp* or *tftp* unless you require these for your server (OS/2) operation. (We used *tftp* to boot IBM Network Stations, but this was for our specific setup.)

Select and set the other tabs, as follows:

- ▶ Routing. We normally make no entry here. However, you may want to configure IP Forwarding active on this screen.
- ▶ Hostname. We normally make no entries here.
- ▶ General. We normally make no entries here.

- ▶ **Security.** You should enter a telnet password on this screen if you enabled OS/2 telnet. Any user on your network can telnet to port 23 on OS/2 TCP/IP and obtain an OS/2 character-oriented window. He could then alter or destroy OS/2, since OS/2 has no security controls. The telnet password you set here is stored (in plain language) in CONFIG.SYS if you are using an older release of OS/2 TCP/IP that is packaged with an OS/2 CD-ROM. Newer versions of OS/2 TCP/IP can have multiple passwords and they are stored in an encrypted format. However, most P/390 owners use the older OS/2 TCP/IP that was preloaded or installed from a basic OS/2 CD-ROM.
- ▶ **Servers.** We normally make no entries here.
- ▶ **Printing.** We normally make no entries here.
- ▶ **Mail.** We normally make no entries here.
- ▶ **Sendmail.** We normally make no entries here.
- ▶ **snmp.** We normally make no entries here.

This completes your TCP/IP configuration. Double click on the top-left button of the window to exit. Read the exit prompts carefully. One prompt will offer to configure mail functions; we normally reply NO to this prompt. Another prompt requests permission to write in CONFIG.SYS; reply YES to this prompt.

Correcting CONFIG.SYS

Both MPTS Configuration and TCP/IP Configuration write additional lines at the end of the CONFIG.SYS file. You must edit CONFIG.SYS and move these lines to a location just before the beginning of the P/390 group of lines in CONFIG.SYS. (Moving these lines is required for P/390-based systems, but not for MP3000 systems.) We normally use the **e** editor and its cut-and-paste functions to do this. You will need to move 5 - 10 lines. Do not use an editor that may truncate lines.

If you later reconfigure MPTS or TCP/IP, *always check CONFIG.SYS* to see if any lines were added at the end of the file. If so, move them to a point above the P/390 lines in CONFIG.SYS. The P/390 group of lines *must* be the last lines in CONFIG.SYS.

In addition to adding lines to CONFIG.SYS, MPTS also changes the PATH, DPATH, and LIBPATH statements, and places itself first in the search list. This has slight performance implications: The author has never bothered to alter this placement.

Update the P/390 configuration

Three P/390 device managers are associated with TCP/IP connections.

- ▶ AWS3274 device manager (uses OS/2 TCP/IP)
- ▶ LAN3274 device manager (uses OS/2 TCP/IP)
- ▶ LCS3174 device manager (uses OS/390 TCP/IP)

AWS3274 and LAN3274 use OS/2 TCP/IP services, not OS/390 TCP/IP services. (Obsolete versions used CM/2 instead. We ignore these here.) S/390 addresses associated with these device managers appear (to OS/390) to be locally-attached non-SNA 3270 terminals. If you use these two device managers perform the same function and use the same code. AWS3274 automatically uses the loopback interface to connect to PCOM clients on the P/390 itself, while LAN3274 connects to TN\TN3270 clients that are connected via the OS/2 TCP/IP LAN. OS/2 TCP/IP, in turn, will use the adapters and interfaces that you defined for OS/2 TCP/IP in MPTS and the OS/2 TCP/IP configuration.

The LCS3172 device manager is associated with OS/390 TCP/IP connections. This will use adapters that were not defined for (OS/2) TCP/IP in MPTS.

Addr	Device	Label	Atype	Size	Mgr	FN/P
>	>	>	>	>	>	>
00C	2540				7	D:\RDR*.JOB
700	3278		DSPY		3	Master Console
701	3278		DSPY		3	
702	3278		DSPY		4	
703	3278		DSPY		4	
A80	3390	OS39RA	CKD	3339C	2	F:\OS39RA.A80
A81	3390	OS3RAA	CKD	3390C	2	F:\OS3RAA.A81
E20	3088				A	
E21	3088				A	

Mgr Codes: 1=AWSFBA 2=AWSCKD 3=AWS3274 4=LAN3274 5=AWS3215 6=AWS2821
7=AWS2540 8=LAN3088 9=LAN3172 A=LCS3172 B=MGR3172 C=WAN3172 D=AWS2703
E=AWSICA F=AWSPBS G=AWSTAPE H=SCSI3480 I=SCSI3420 J=AWS3480 K=AWS3420
L=AWSOMA M=AWSTFA N=AWSPCSR V O=CHAN370 P=AWSICE

Figure 6-9 P/390 Device Configuration display

The configuration process is started from the P/390 icons on the OS/2 desktop. Double click on the Configuration icon. This should produce the *Configuration Password* screen. Press Enter (or enter a password if one has been set) to obtain the *Function* screen. Press F2 to obtain the *Update System Devices* screen similar to that shown in Figure 6-9. You should be somewhat familiar with this *device map* function from your initial P/390 installation and use.

Your display may differ from the example in the figure. You will probably have more devices defined. (You can scroll through them with PageUp and PageDown keys.) More importantly, the list of device managers at the bottom of the display may have different letters assigned. In the figure, letter A is associated with device manager LCS3172, for example. You may have different letters in your display; use them. You can create a new line in the devices table by entering the appropriate parameters in the line containing the > symbols and pressing Enter.

You may first need to configure devices for AWS3274 and LAN3274. The device map delivered with the AD CD-ROM already has some of these devices at appropriate addresses. The OS/390 system on the CD-ROM expects to find a local 3270 at address 700 to use as the MVS master console. You should define at least three AWS3274 addresses at 700, 701, and 702. (The two extra addresses are for VTAM sessions, such as TSO or CICS.)

You should define several LAN3274 addresses. The number of LAN3274 addresses you define represents the maximum number of concurrent 3270 connections via OS/2 TCP/IP. Client systems (your potential TSO users, for example) connect to these addresses by establishing a TN3270 session with OS/2 TCP/IP at port address 7490. (You can change this port address by changing default LAN3274 parameters, but you cannot use port 23 unless you stop OS/2 telnet or reassign its port address. Port 7490 was selected during P/390 development (for historical reasons) and most P/390 sites continue to use port 7490.) The OS/390 on the AD CD-ROM has a large number of local 3270 devices defined starting at addresses 700 and 900.

To configure and use OS/390 TCP/IP, we need to assign S/390 addresses to the emulated 3172 channel addresses that will be used for OS/390 TCP/IP connections. The 3172 emulation function is provided by the LCS3172 device manager. OS/390 TCP/IP uses two addresses, an even/odd pair of contiguous addresses, to communicate with a 3172 control unit. As with almost all P/390 emulated devices, you could select any three-digit address pair you wish. In a practical sense, you should select an address pair that is already generated in the OS/390 AD CD-ROM system (or in any other OS/390 system you may be using).

OS/390 uses a device definition known as a Channel-To-Channel (CTC) adapter to communicate with a 3172 control unit. A CTC is also known as a 3088 control unit. The AD CD-ROM has addresses E20-E27 and E40-E47 generated as CTCs, and you can use any even/odd pair in this range. For our examples we will use E20 and E21 for our first (and only) OS/390 TCP/IP connection. This will correspond to MPTS adapter number one -- the EtherJet adapter in the examples earlier.

You want to add two lines in the device map to define addresses E20 and E21 as device type 3088 (another name for CTC devices) and assign these to the LCS3172 device manager. Two entries like this are shown in Figure 6-9 on page 332.

When you are finished, press F10 (to exit from this function) and then F6 (to save your changes). You must restart the P/390 subsystem for changes to be effective.

If you have two OS/390 TCP/IP adapters, you would need four lines in the device map, and so forth. You can use different S/390 addresses, but the addresses you use must be generated as 3088 or CTC device types in OS/390. (You can change or add OS/390 devices by using the HCD utility under TSO. We suggest you do not try this unless you are familiar with HCD or have a pressing requirement to use different addresses.) There is no connection between the P/390 device map definitions and any specific LAN adapter. This connection is made later, in the OS/390 TCP/IP profile definitions.

Rebooting OS/2 and checking results

At this point, we suggest you shut down OS/2 and reboot. If all the changes were correct, there should be no errors during the rebooting. Booting may take longer; we assume this is due to startup times for the LAN adapters.

After OS/2 is restarted, open an OS/2 window and examine the file C:\IBMCOM\LANTRAN.LOG. This should contain lines such as:

```
IBM LANBB is accessing IBM DIX LAN Interface
Adapter 0 is using node address 006629050660
...
Port out: The IBMPCI Token Ring Adapter opened at 16 Mbps, half duplex
...
```

You can use this listing (and your notes about which MAC address is associated with which adapter) to determine which LAN adapter is adapter 0, which is adapter 1, and so forth. If significant errors are noted in this file, you must resolve these errors before proceeding with your TCP/IP configuration.

You can also examine \P390\AWS3172.ERR for error messages. Some errors reported in this file are acceptable. The return codes and module names reported here are not readily available in routine P/390 documentation.

After your reboot, you should see a window for inetd and another for portmap. You can minimize these windows. You should be able to ping the OS/2 system from another machine on the same IP subnet. You should be able to telnet (not TN3270) to the OS/2 TCP/IP address (to port 23) and log onto OS/2. You will need the password you entered in the OS/2

TCP/IP configuration panel. (OS/2 can support several simultaneous telnet users. However, we strongly suggest you do not make this available to other users. Once logged onto OS/2, there are no security controls for OS/2. You should make the OS/2 TCP/IP telnet password as obscure as possible if your system will appear on a large network. You can change it in CONFIG.SYS (older OS/2 TCP/IP releases) or through the TCP/IP configurator (newer releases)). The OS/2 telnet function can be used to remotely IPL OS/390, among other useful tasks.

6.32.5 Configuring PCOM

You need a 3270 emulator if you want to access AWS3274, LAN3270, or the OS/390 TN3270 interface. You need a 3270 emulator on the P/390 system itself, for the MVS master console and for local TSO sessions. There are many 3270 emulator products available from many vendors. One is the IBM Personal Communications AS/400 and 3270,⁸ with versions available for OS/2, Windows 95, and other systems. This product is often known as PCOM. This is the emulator most commonly used with P/390 systems and we briefly describe the customization of a 3270 session here.

PCOM can be installed from diskettes or a CD-ROM. Once installed, you need to execute a configuration function for each 3270 session you plan to use. The first GUI screen of the configuration process permits you to select a path for connecting your PC to a host system. You should select a **PC -> LAN -> TCP/IP -> S/390** path on this GUI screen, then select the CONFIGURE button.

In our examples, we are using IP address 9.12.17.150 for our OS/2 TCP/IP adapter and will use 9.12.17.151 for our OS/390 TCP/IP interface. (The setup for this is described later.) Using these IP addresses, we may need TN3270 connections for:

- ▶ 127.0.0.1, port 7490. TN3270 emulator windows on the P/390 itself would use this loopback address and port number for the MVS master console and local VTAM (TSO) sessions.
- ▶ 9.12.17.150, port 7490. TN3270 clients on the LAN can use this combination to access the LAN3274 device manager. This device manager makes the connections appear as local 3270s to OS/390.
- ▶ 9.12.17.151, port 23. TN3270 clients use this address and port to connect to OS/390 TCP/IP.

A given emulator session would use only one of these connections at a time, of course. Depending on its purpose, a given emulator session might have only one of these connections defined.

Within the CONFIGURE screens you should select:

- ▶ Screen size 32x80 (this is optional, but very common).
- ▶ Link Parameter - select CONFIGURE LINK.
 - Host name - 127.0.0.1 (or any other appropriate address).
 - Select ADVANCED SETUP.
 - Change the port number to 7490 (or port 23 for direct OS/390 TCP/IP connections).
 - Select AUTO-RECONNECT.

When you exit from the PCOM configurator, it may offer to create a desktop icon for the session. We suggest you accept this. You can create several PCOM sessions. We usually create three sessions for PCOM installed on the P/390 itself, and two sessions for PCOM installed on other personal computers.

⁸ Also known as *eNetwork Personal Communications*.

Start one of the sessions. There is a pull-down menu named ASSIST; under this is an option CUSTOMIZE KEYBOARD. If appropriate, use the Customize Keyboard function to map the common C characters, such as square brackets, onto your keyboard. You can also remap the ENTER key; the default location for the 3270 ENTER function is the right-hand Ctrl key of the PC keyboard. (Defining the square brackets as “apl ad” “apl bd” (which results in the hex values ad and bd for the characters) worked well for us.)

After starting one of the PCOM sessions, you can use the COMMUNICATIONS pull-down menu, with the CONFIGURE suboption, to change the IP address and port number for the TN3270 session. Using the addresses in our examples, you could change from 9.12.17.151 port 7420 to 9.12.17.151 port 23 to change from a local 3270 connection (via OS/2 TCP/IP and LAN3274) to a direct OS/390 TCP/IP connection.

A side note: PCOM 3270 sessions apparently expect you to have a printer configured for the session. You may see a timeout of about ten seconds (100% CPU usage on the OS/2 pulse display) before a popup window appears asking about a printer. At this point we select the CANCEL option in the popup window. This usually happens only once per PCOM 3270 session.

You can eliminate the attempt to use a printer by editing the WS file associated with the PCOM session. These are usually found in the C:\PCOMOS2\PRIVATE directory. If a PCOM session name is A, the associate file is A.WS. Within the WS file, find (or create) a section beginning [LT] and add the following line to this section:

```
IgnoreWCCStartPrint=Y
```

The file format is obvious once you list or edit it.

Summary and next step

If you have followed the steps in this section, your P/390 should:

- ▶ Have LAN adapter cards installed.
- ▶ Have MPTS configured to cover both OS/2 TCP/IP and OS/390 TCP/IP (using separate adapter cards, of course).
- ▶ Have OS/2 TCP/IP configured and operational.
- ▶ Have the P/390 device map updated to provide two addresses for the LCS3172 device manager. This will provide the path from the adapters/MPTS to OS/390 TCP/IP.
- ▶ Have OS/2 PCOM installed and configured.

You should be able to IPL OS/390 and use the PCOM sessions defined on the P/390 for the MVS master console and local TSO sessions. You will need a local TSO session to complete the OS/390 TCP/IP customization necessary before it can be used. The next steps for customizing OS/390 TCP/IP start in “TCP/IP” on page 292.

6.32.6 IBM Integrated Server setup

The setup for the IBM S/390 Integrated Server, machine type 3006, is generally the same as for a P/390, with the following changes:

- ▶ The installation CD-ROM provided with the Integrated Server already has much of the setup completed.
- ▶ The integrated Ethernet adapter is a different adapter than the one used with the PC Server 330-based P/390 systems.

If you have an Integrated Server, we suggest you examine the MPTS configuration. (From an OS/2 window, enter the command MPTS, and then follow the earlier P/390 description of MPTS usage.) The setup CD-ROM will include the definition of the integrated Ethernet adapter. If you purchase additional LAN adapters, you must define them—just as described in the P/390 section.

6.33 Application StarterPak LAN setup

TCP/IP setup for an Application StarterPak should be much simpler than for a P/390 or R/390. We will assume you have the OSA-2 adapter that has two LAN ports; this is commonly known as the ENTR (EtherNet - TokenRing adapter. Each port can be used as either Ethernet (10 Mbps) or token ring.

If the LAN ports are used only for OS/390 TCP/IP, very little setup is needed. If the ports are also used for SNA connections, then you will probably need to use the OSA/SF utility, and this is outside the scope of the current discussion. By default, the ENTR adapter will support TCP/IP, without any additional customization with the OSA/SF utility.

The primary setup challenge is to assign the desired protocol (Ethernet or token ring) to each of the two ports. This is done from the Support Element (SE) console, and will probably involve a Power-On-Reset (POR) cycle. To begin, you need to examine the connectors on the ENTR adapter.

Connections to the ENTR OSA-2 adapter

Figure 6-10 on page 337 roughly illustrates the external connections of an ENTR adapter. There are two ports, 0 and 1. Each port has three connectors, two for Ethernet and one for token ring. Only one of the connectors may be used for each port. For Ethernet, you select which connector to use based on the type of wiring you have; use the UTP connector for 10BaseT or use the AUI connector for an external Ethernet transceiver. Only one connector on each port should have a connection. (A wrap plug is considered a connection.)

The Application StarterPak will sense which connection, for each port, is used when you *first* power up the system. It will remember this choice thereafter. This is often not acceptable, because you probably did not have your LANs connected when you first brought up the system; you probably left the IBM-provided wrap plugs in place.

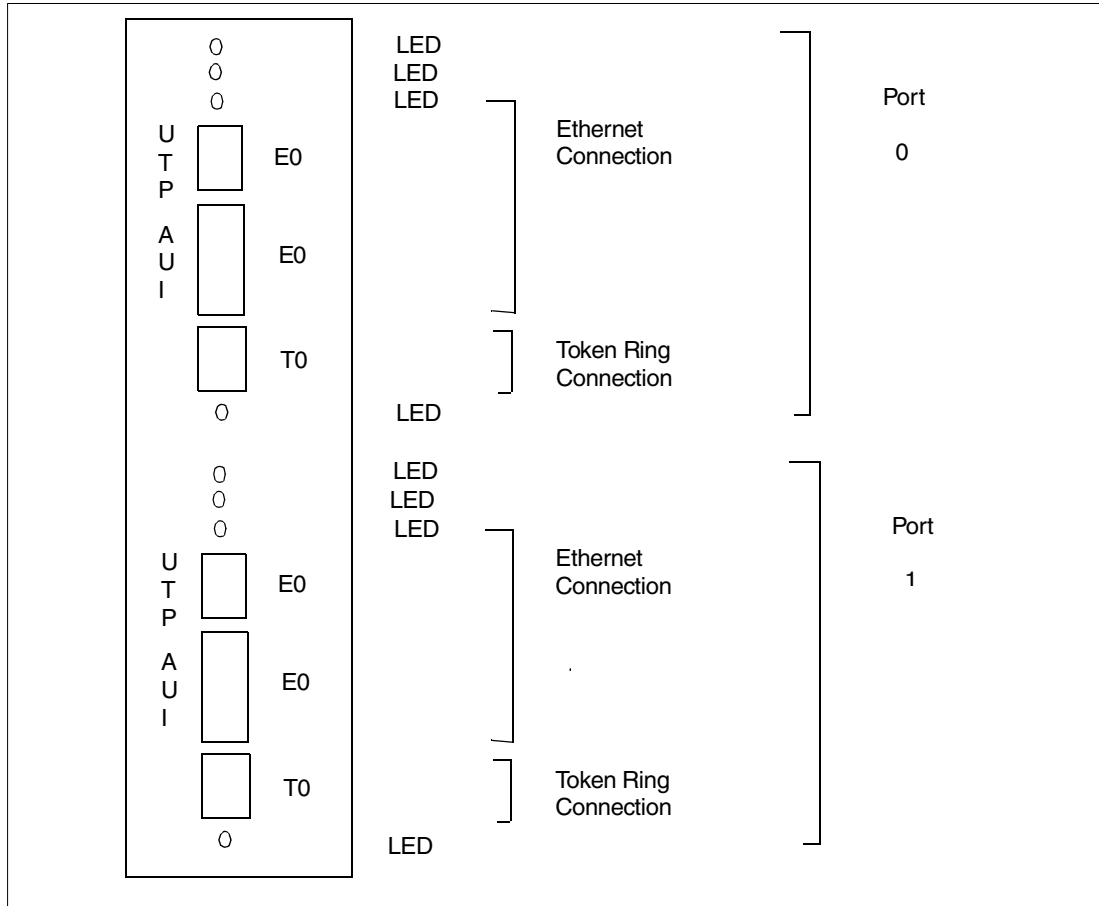


Figure 6-10 OSA-2 Ethernet/Token Ring Adapter Connections

Resetting the Ethernet vs. token ring selection

To set (or reset) the ENTR adapter to your desired configuration (that is, your choice of either Token Ring or Ethernet for each of the two ports) you need to:

- ▶ Remove any connections from each of the six connectors on the ENTR adapter.
- ▶ Attach one connector to each port, going to an appropriate LAN. (For Token Ring, the LAN might be an IBM 8228 MAU with no other connections. For 10BaseT Ethernet, it might be a connection to a hub.) If no connection exists for a port, the default setting is to Ethernet.
- ▶ Use the SE tools to enable a reset function for the OSA adapter. This is described below.
- ▶ Perform a POR (Power On Reset) function, during which the machine will sense the LAN connections and remember the settings.

There are four LEDs (indicator lights) on each half of the ENTR adapter face. These are (from top to bottom):

- ▶ A2 - Not Operational. This will be on when the card is first reset; it should not be on during normal operation.
- ▶ B2 - Operational. This light should flash during normal operation of the adapter.
- ▶ C2 - Ethernet port not operational on LAN.
- ▶ D2 - Token Ring port not operational on LAN.

There are a great many Service Element (SE) *panels* provided with the Application StarterPak, used from the personal computer that is usually placed on top of the StarterPak. Use of the panels involves much mouse clicking to navigate from display to display. There are often several ways to navigate to a particular display. The following instructions should guide you through one path to reach the panel where you can enable the *Enable autosense on next reset* function to change the protocols for ENTR ports. These instructions are meant to be used while you are performing the indicated functions at the SE display:

- ▶ Select the *Groups* option on the left-hand side of the display.
- ▶ Double click on the *CPC* icon.
- ▶ This should produce a small icon that, in a sense, appears as a box within a box. Use the *right mouse button* to select the right-hand (green) portion of the outer box.
- ▶ This should produce a small window offering a choice of CPs or CHPIDs. Click on the *CHPIDs* option.
- ▶ This will produce a window with an icon for each chpid on your system. OSA adapters have a section of pipe as the icon. The first OSA adapter is normally chpid 5C.
- ▶ On the right-hand side of the SE display, find the *CHPID Operations* display.
- ▶ Use the *right mouse button* to drag the 5C chpid icon (or whichever OSA chpid you need) to the *Advanced Facilities* icon in the CHPID Operations display.
- ▶ You should receive a window with a single choice: *OSA Specific Advanced Facilities*. Select this option and click OK.
- ▶ You should then receive a menu with options such as:
 - Enable or disable ports
 - Query port status
 - Run port diagnostics
 - View port parameters
 - Display or alter MAC address
 - Set ethernet mode
 - Enable autosense on next reset <--- this one
 - Cancel command
- ▶ You can try the various options to learn more about your OSA port.
- ▶ Before exiting, select *Enable autosense on next reset*, and click *apply*.
- ▶ Exit from the OSA and CHPID panels on the left-hand side of the screen.
- ▶ Be certain you have your desired LAN connections for the ENTR ports.
- ▶ Perform a POR, using this option in one of the right-hand displays on the SE console.

Finding addresses of OSA interfaces

The Application StarterPak uses IOCDS files to associate channels, control units, and devices with the addresses (device numbers) seen by OS/390. There is a very rough equivalent to the P/390 Configurator function, but an IOCDS can be considerably more complex. The best way to find the addresses for the OSA-2 ports you want to use for TCP/IP is to ask the systems programmer who defined the IOCDS for your system.

If this is not practical, and if your StarterPak is OS/390 working (implying that a valid IOCDS is being used), you can use the SE console as follows:

On the left-hand side, select *Groups*. On the right-hand side select the *CPC Configuration* display. Using the right mouse button (RMB), drag the CPC icon over the Input/Output (I/O) Configuration icon on the right-hand side of the screen.

This should display the I/O Configuration window. Examine the top few lines to determine which IOCDS data set is active; it will be A0, A1, A2, or A3. Use the mouse to select the data set line corresponding to this name. Select the **View** option from the action line of the I/O Configuration window; then select **Channel Path Configuration** from the drop-down menu.

This should produce a window listing all the chpids on your system. The first OSA-2 adapter is normally chpid 5C and the second is 5A. We will assume your ENTR OSA-2 adapter is the first one. (This implies it is located in the right-most slot of the available slots for OSA adapters.) Use the mouse to select the line corresponding to your OSA chpid. Then select **View** in the action line and **Device Information** from the drop-down menu.

- ▶ This should produce a window containing something like this:

```

UNITADD= ADDRESS UNIT MODEL STADET= TIMEOUT=
      00      0900   OSA          Y         Y
      01      0901   OSA          Y         Y
      ...
      FE      099A   OSA          Y         Y

```

This says that the first OS/390 address for the OSA adapter is 900, and so forth. By default the first two addresses (900, 901) are for TCP/IP connections to the first port of the adapter (port 0) and the next two addresses (902, 903) are for TCP/IP connections to the second port of the adapter (port 1). That is, by default, the first four addresses of an ENTR adapter are reserved for TCP/IP access. Additional addresses, after the first four, might be used for SNA access. A single ENTR adapter can be shared with multiple LPARs and can be used for both TCP/IP and SNA concurrently.

- ▶ Exit from the SE display screens.

If you have access to the IOCDS used for your system, you can determine your OSA addresses from it. A typical IOCDS listing might be:

```

CHPID PATH=(40),TYPE=CVC
CHPID PATH=(41),TYPE=CNC
CHPID PATH=(42),TYPE=CTC
CHPID PATH=(43),TYPE=CTC
CHPID PATH=(58),TYPE=OSA
CHPID PATH=(5A),TYPE=ISD
CHPID PATH=(5B),TYPE=ISD
CHPID PATH=(5C),TYPE=OSA          <-----
CHPID PATH=(5D),TYPE=ISD
CHPID PATH=(5E),TYPE=ISD
CHPID PATH=(5F),TYPE=ISD
CNTLUNIT CUNUMBR=0300,PATH=(5A,5E),UNITADD=((00,064)),      *
        UNIT=3990
CNTLUNIT CUNUMBR=0500,PATH=(5D),UNITADD=((00,001)),UNIT=3490
CNTLUNIT CUNUMBR=0600,PATH=(5B,5F),UNITADD=((00,064)),      *
        UNIT=3990
CNTLUNIT CUNUMBR=0890,PATH=(58),UNIT=OSA
CNTLUNIT CUNUMBR=0900,PATH=(40),UNITADD=((00,032)),SHARED=N, *
        PROTOCL=D,UNIT=3174
CNTLUNIT CUNUMBR=0990,PATH=(5C),UNIT=OSA <-----
CNTLUNIT CUNUMBR=0F00,PATH=(41),UNITADD=((00,032)),UNIT=3174
IODEVICE ADDRESS=(300,064),CUNUMBR=(0300),UNIT=3390
IODEVICE ADDRESS=(500,001),CUNUMBR=(0500),UNIT=3490
IODEVICE ADDRESS=(600,064),CUNUMBR=(0600),UNIT=3390
IODEVICE ADDRESS=(890,010),UNITADD=00,CUNUMBR=(0890),STADET=N,*
        UNIT=OSA
IODEVICE ADDRESS=89A,UNITADD=FE,CUNUMBR=(0890),STADET=Y,    *
        UNIT=OSAD
IODEVICE ADDRESS=(900,032),MODEL=X,CUNUMBR=(0900),TIMEOUT=Y,*
        STADET=N,UNIT=3270
IODEVICE ADDRESS=(990,010),UNITADD=00,CUNUMBR=(0990),STADET=Y,*
        UNIT=OSA          <-----
IODEVICE ADDRESS=99A,UNITADD=FE,CUNUMBR=(0990),STADET=Y,    *
        UNIT=OSAD
IODEVICE ADDRESS=(F00,032),MODEL=X,CUNUMBR=(0F00),STADET=Y, *

```

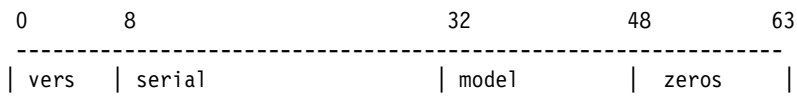
Information in this listing should match the information from the SE displays.

If OSA/SF has been used to customize your OSA adapter(s), you may have a completely different usage of OSA ports. Using OSA/SF is normally associated with SNA setup for the OSA ports. If your StarterPak was preloaded by IBM, there was probably no OSA/SF customization and you can assume the first four addresses (for an ENTR adapter) can be used for TCP/IP. For more information about OSA/SF see *Open Systems Adapter 2 Implementation Guide*, SG24-4770.

The following sections of this document assume that you will use S/390 addresses E20 and E21 for your TCP/IP connections. These addresses correspond to most P/390 and R/390 systems. If you have a StarterPak, substitute your addresses for E20 and E21. (In the example above, the StarterPak addresses are 900 and 901.)

6.34 How to use the CPU ID of a P/390

A P/390 (and R/390) system does not have a unique serial number. In this characteristic it differs from other S/390 machines. A program can read the CPU ID (also known as the CPU serial number) with the STIDP instruction. This is a privileged instruction, so the user must be in supervisor state. STIDP stores eight bytes of information (on a doubleword boundary):



For a P/390 adapter:

- ▶ vers (8 bits):
 - x'01' = Micro Channel P/390 with no daughter card
 - x'03' = Micro Channel P/390 with 32MB daughter card
 - x'04' = Micro Channel P/390 with 96MB daughter card
 - x'08' = PCI P/390 with 32MB memory
 - x'09' = PCI P/390 with 128MB memory (64 Mbit DRAMS)
 - x'0A' = PCI P/390 with 128MB memory (16 Mbit DRAMS) (rare)
 - x'11' = PCI P/390E with 256MB memory (64 Mbit DRAMS)
 - x'13' = PCI P/390E with 1GB memory (stacked 64Mbit) (not generally available)
- ▶ serial is a 24-bit value (6 hex digits) that can be set from the P/390 configurator.
- ▶ model is always the value x'7490' (16 bits)

There is no unique serial number per P/390 adapter. Any and all P/390 adapters can be set to any serial number by their owner. The serial number can be changed at any time (although the P/390 subsystem must be restarted to make the change effective). Software that depends on serial numbers for license protection will not be effective on a P/390 adapter. Such software could check bits 32-47 of the STIDP results for the value x'7490' and then conclude that it is executing on a P/390 and skip any additional serial number check. Unless the product contains very unusual code, we suggest that the version data not be used. Whether a P/390 adapter is Micro Channel or PCI should not be relevant to a program, and OS/390 facilities should be used to determine memory characteristics.

The version numbers are not intended as a programming interface. Also, note that some of the fields appear to change when running under VM.

6.35 How to perform AWSTAPE mounts

Tapes emulated with the AWSTAPE device manager can be used as labelled or unlabelled tapes. Suppose you have a job that will write a tape, and the JCL specifies a standard labelled tape with volume serial number TAPEAA.

```
//SYSUT2 DD UNIT=583,DISP=NEW,DSN=TAPEDATA,LABEL=(1,SL),
//          VOL=SER=TAPEAA
```

The JCL assigns the tape to unit 583, which (in our P/390 configuration file) is associated with device manager AWSTAPE. The entry in the configuration table does not associate an OS/2 file with this device. Note that AWSTAPE only emulates 3420 drives; be certain to use the appropriate OS/390 addresses.

Addr	Device	Label	Atype	Size	Mgr	FN/P
>	>	>	>	>	>	>
583	3420				G	

The key MVS console interaction will look like this:

```
$HASP373 OGDEN5   STARTED - INIT 1
IEF403I  OGDEN5   STARTED
IEF503I  INCORRECT VOLUME LABEL OR I/O ERROR
*IEF233A M 583,TAPEAA,,OGDEN5
<----- at this point, switch to
          an OS/2 window and enter:
          AWSMOUNT 583 D:\DATA12 /C

IEC512I  LBL ERR 583,,      ,NL,TAPEAA,SL,OGDEN5
IEC704A  L 583,TAPEAA,SL,6250 BPI,OGDEN5
*05 IEC704A REPLY 'VOLSER,OWNER INFORMATION', 'M' or 'U'
5,'TAPEAA' <----- reply
IEE600I  REPLY TO 05 IS; 'TAPEAA'
IEC705I  TAPE ON 583, TAPEAA,SL,6250 BPI,OGDEN5
IEF234   K 583,TAPEAA,PVT,OGDEN5
IEF404I  OGDEN5 - ENDED
<----- in OS/2 window enter:
          AWSMOUNT 583 /RUN
```

The OS/2 command `AWSMOUNT 583 D:\DATA12 /C` will create a file on your D drive with the name `DATA12`. (You can use any valid OS/2 name and directory, of course.) The `/C` option is required when creating a *new* emulated tape volume. You need to respond to the standard MVS messages to create the appropriate standard labels. After the job ends, you probably want to dismount the tape; use the `AWSMOUNT 583 /RUN` command. (*RUN* means Rewind UNload.) The file that was written (`D:\DATA12` in this example) can later be mounted to satisfy a mount request for another job, or used with OS/2 programs (if these programs are specially written to handle the AWSTAPE format).

You should enter a full path name when specifying a file name with the `AWSMOUNT` command. If you do not, it defaults to the P390 directory and you probably do not want it there.

The file can be on diskette; diskettes are convenient for storage or moving data to another P/390 MVS system. *Remember to use the `AWSMOUNT xxx /RUN` command before removing the diskette.* In general, we suggest that you do not use diskettes for AWSTAPE output. Operation is slow (about 12 minutes to write a full diskette, versus 6 seconds for the same job writing to disk) because the AWSTAPE device manager must constantly verify that enough space is available to write the next block. (This is necessary because there is no way to duplicate the end-of-tape (EOT) condition associated with a “real” tape drive. The EOT condition warns that the end of tape is near, but the program can continue writing the current

block (plus a few more, for labels). AWSTAPE must verify the space available frequently because other programs (OS/2) might be writing to the same disk or diskette and consuming free space.) If you need an AWSTAPE file on diskette, you can create it on disk and later (with OS/2 commands) copy it to diskette. Reading an AWSTAPE file from diskette took 4 minutes (for a 1.4MB file). We do not know why this is so slow.

If your OS/390 tape drive definitions include (or default to) OFFLINE=NO, then MVS will consider any tape drives it detects at IPL to be ONLINE. (If you rework your configuration, we suggest that you might want to specify OFFLINE=YES for your tape drives.) OS/390 will detect all tape drives defined in your P/390 configuration file (if they correspond to tape addresses known to OS/390, of course), and they will be ONLINE after IPL. When OS/390 needs to mount a tape (in response to a submitted job), it will sense/read tape drives that are ONLINE to determine if the requested tape has been premounted. This is the AVR (automatic volume recognition) function of OS/390. If AWSTAPE drives are defined (via the P/390 configuration file) with no associated file (as in our example above), OS/390 will produce a variety of error messages when it attempts to sense/read these drives. A typical message is:

```
IOS000I 584,01,FPR,02,0E40,,01000000 .....
```

You can ignore these messages. You will sometimes get a message like this:

```
*06 IEC613A OGDEN5, ,583,TAPEAA TAPE POSITION ERROR
      REPLY 'R' RETRY OR 'U' CONTINUE WITH ABEND
```

This may occur before you issue the AWSMOUNT command. After you issue your AWSMOUNT command, reply R to this message. OS/390 should then issue the IEF503 message about an incorrect label, and you would proceed as usual. If your output disk (or diskette) becomes full, AWSTAPE signals an end-of-tape (EOT) condition. OS/390 will call for another tape volume, and you must use the AWSMOUNT command to mount another volume (probably a diskette, but possibly a file on another disk). This process creates a multivolume file, just as it would if real tapes were used. AWSTAPE cannot fully emulate the EOT environment of a tape drive. There may be situations that cannot be handled; the results (to MVS) would be similar to a tape running off the end of a reel before the last blocks were written. We suggest that, when possible, you do not use AWSTAPE if your expected tape size is larger than the free space on your disk (or diskette).

Note that AWSMOUNT is not related to the MVS MOUNT command.

In general, AWSTAPE is slower than any other tape function of the P/390 for writing output and faster than any other for reading input (except when using diskettes drives).

The RUN option of AWSMOUNT does not completely remove the link between the last OS/2 file used and the AWSTAPE device manager. Instead, it effectively makes the emulated tape drive *not ready*. If OS/390 attempts to use the drive, it will get an *intervention required*. The *next* OS/390 access after this will work and will connect to the last-used OS/2 AWSTAPE file and make it available for use. The only way to completely disconnect the OS/2 file is to use AWSMOUNT to mount another OS/2 file.

6.36 How many token ring adapters do I need

The following diagram may help. It indicates which P/390 device managers are involved in the discussion, and indicates which MPTS protocol is used. (The CM/2 entry is for completeness; we will not discuss it further.)

```
MVS TCP/IP -----> LCS3172 ----> 802.2 MPTS -----> TR adapter
```

MVS SNA -----> LAN3172 ----> 802.2 MPTS -----> TR adapter

OS/2 TCP/IP -----> TCP/IP MPTS -----> TR adapter

OS/2 CM/2 (SNA)----> 802.2 MPTS -----> TR adapter

OS/2 TCP/IP includes the use of AWS3274 and LAN3274.
Combinations, except for the two TCP/IPs, can share the same
adapter

The MPTS TCP/IP protocol is only for OS/2 TCP/IP. MVS TCP/IP operates through the LCS3172 device manager, which uses the 802.2 protocol interface to MPTS. You cannot share a single adapter between MVS TCP/IP and OS/2 TCP/IP. (Note that the token ring adapter can also be used for OS/2 NetBios protocols at the same time it is used for other protocols, although NETBios is not further discussed here.)

A single adapter can be used for any *one* of these purposes:

- ▶ LCS3172
- ▶ LAN3172
- ▶ OS/2 TCP/IP
- ▶ OS/2 SNA (usually via CM/2)
- ▶ LCS3172 and LAN3172
- ▶ LCS3172 and OS/2 SNA
- ▶ LAN3172 and OS/2 TCP/IP
- ▶ LAN3172 and OS/2 SNA (if different SAPs) are used
- ▶ OS/2 TCP/IP and OS/2 SNA
- ▶ LCS3172 and LAN3172 and OS/2 SNA (different SAPs)
- ▶ LAN3172 and OS/2 SNA (different SAPs) and OS/2 TCP/IP

If you need other combinations, you will need multiple token ring adapters. This is most likely to happen in two circumstances: (1) you want to use both OS/2 TCP/IP and MVS TCP/IP, or (2) you want to use MVS SNA and CM/2, both with SAP=4.

For token ring, the general rule is: (number of adapters) = (number of TCP/IP stacks).

6.37 How many Ethernet adapters do I need

The following diagram may help. It indicates which P/390 device managers are involved in the discussion, and indicates which MPTS protocol is used. (The CM/2 entry is for completeness; we will not discuss it further.)

MVS TCP/IP -----> LCS3172 ----> 802.2 MPTS -----> En adapter
(DIX or 802.3)

MVS SNA -----> LAN3172 ----> 802.2 MPTS -----> En adapter
(DIX or 802.3)

OS/2 TCP/IP -----> TCP/IP MPTS -----> En adapter
(DIX or 802.3)

OS/2 CM/2 (SNA)-----> 802.2 MPTS -----> En adapter
(DIX or 802.3)

OS/2 TCP/IP includes the use of AWS3274 and LAN3274.
Some combinations can share the same adapter.

The situation with Ethernet is more complex than the situation with token ring because there are two types of Ethernet frames: DIX and 802.3. You must select one or the other. You can have these combinations:

TCP/IP with DIX	(commonly used)
TCP/IP with IEEE802.3	(rarely used)
SNA with DIX	(rarely used)
SNA with IEEE802.3	(more commonly used)

(Do not confuse 802.2 with 802.3. The P/390 device managers always use the 802.2 protocol to communicate with MPTS. Within the 802.2 protocol there may be 802.3 or DIX frames.) The MPTS selection of DIX or 802.3 affects all 802.2 frames (LCS3172, LAN3172, and OS/2 SNA).

A single adapter can be used for any *one* of these purposes:

- ▶ LCS3172 (with MPTS 802.2 set to either DIX or IEEE802.3. You probably want DIX)
- ▶ LAN3172 (you probably want MPTS 802.2 set to 802.3, although some SNA systems can use DIX) (OS/2 and Windows NT (but not Windows 95) support SNA via DIX Frame Types).
- ▶ OS/2 TCP/IP (either DIX or IEEE802.3. The choice is made in the TCP/IP configurator ADVANCED OPTIONS second panel. The default is DIX.)
- ▶ OS/2 SNA (probably CM/2). (You could set MPTS 802.2 to either 802.3 or DIX; you probably would use 802.3.)
- ▶ LCS3172 and LAN3172 (You would need to set MPTS 802.2 to either DIX or 802.3, and this would apply to both protocols. Since you probably want DIX for LCS3172 and 802.3 for LAN3172, you are unlikely to want this combination.)
- ▶ LCS3172 and OS/2 SNA. (You would need to set MPTS 802.2 to either DIX or 802.3, and this would apply to both protocols. Since you probably want DIX for LCS3172 and 802.3 for SNA, you are unlikely to want this combination.)
- ▶ LAN3172 and OS/2 TCP/IP (in this case, MPTS 802.2 could be set to either 802.3 or DIX; you probably would want 802.3. OS/2 TCP/IP could, independently, be set to DIX or 802.3. You probably would want DIX. This can be a useful combined use.)
- ▶ LAN3172 and OS/2 SNA (in this case you could use either 802.3 or DIX for the 802.2 MPTS specification; you probably want 802.3. You would need to use a different SAP number for either VTAM or OS/2 SNA.)
- ▶ OS/2 TCP/IP and OS/2 SNA (you can select either DIX or 802.3 independently for the two protocols. You probably want DIX and 802.3, respectively, and these are the default settings.)
- ▶ LCS3172 and LAN3172 and OS/2 SNA (You would need to set MPTS 802.2 to either DIX or 802.3, and this would apply to both LCS3172 and LAN3172. Since you probably want DIX for LCS3172 and 802.3 for SNA, you are unlikely to want this combination. If you do use it, you will need different SAP numbers for VTAM and OS/2 SNA.)
- ▶ LAN3172 and OS/2 TCP/IP and OS/2 SNA (the DIX or 802.3 frame chosen for MPTS 802.2 would apply to both LAN3172 and OS/2 SNA; you would probably select 802.3. The DIX or 802.3 selection for OS/2 TCP/IP is independent of MPTS; you would probably want DIX. You will need two different SAP numbers.)

If we can assume that you want DIX TCP/IP, and do not want SNA in DIX frames, we can summarize common situations thus:

- ▶ You need two adapters to use both MVS TCP/IP and OS/2 TCP/IP.
- ▶ You need two adapters to use MVS TCP/IP and VTAM.
- ▶ You need three adapters to use MVS TCP/IP, VTAM, and OS/2 TCP/IP.

In words, the general rule is: (number of adapters) = (number of TCP/IP stacks) if, and only if, SNA and TCP/IP use the same frame format (DIX or 802.3), else SNA and TCP/IP will need separate adapters.

The above discussion applies to P/390 and Integrated Servers. An R/390 cannot run SNA over DIX frames, so that option is removed from the above lists. The summary (in the last paragraph above) should also apply to R/390 systems.

6.38 How to display square brackets with PCOM

The problem of the “C characters” or “UNIX characters” within traditional MVS environments (including TSO) is complicated because many components are involved. There is no central function in OS/390 that manages character sets.

If you are using PCOM to provide TN3270 sessions for TSO and OMVS, you can solve part of the “square bracket” problem, as follows:

- ▶ In your ISPF session, go to option 0. In your TEMINAL TYPE definition you probably have 3278. Change this to 3278A. (This probably means changing the selection from 3 to 4). Exit from option 0. This simple change may help display square brackets properly, once they are entered.
- ▶ To enter square brackets from the PCOM keyboard, you will need to remap the keyboard. Display the Menu Bar for the TN3270 window.

```
Select the ASSIST item from the Menu Bar
Select KEYBOARD SETUP from the pull-down menu
Select CUSTOMIZE from the keyboard setup menu. You
may need to specify a local Keyboard File Name if you
have used this function before.
On the keyboard map displayed, select (mouse, click)
the key that should have a left square bracket. In the
Base box (toward lower right corner), type “apl ad”.
(without the quote marks).
Click CHANGE KEY.
On the keyboard map, select the key that should have a
right square bracket. In the Base box, type “apl bd”.
Click CHANGE KEY.
Exit by double clicking the top left-hand icon.
If you are asked for a new file name for the modified keyboard
definition, use something like A.WS.
Exit back to the TN3270 session.
```

If this works correctly, you can specify the A.WS file as a customized keyboard file for other TN3270 sessions, without having to go through the detailed customization again.

For purposes of C or shell programming, the square brackets should be mapped to x'ad' and x'bd'.

6.39 How to edit a PDS from OS/2

OS/390 volumes (for a P/390 based system) are emulated CKD devices. The emulator runs under OS/2 (or AIX, in the case of an R/390). From the OS/2 point of view, an OS/390 volume (with all its data sets) is a single OS/2 file; a very large OS/2 file. The internal format of this file is not documented. You cannot edit MVS data sets or OpenEdition files from OS/2 or AIX, because the OS/2 and AIX editors do not understand the format of the files used for emulation. (And they would not understand the EBCDIC data in the files.)

An independent vendor product, *Facilitator*, runs under OS/2 and can access OS/390 PDS and sequential files from emulated CKD volumes. It does this by performing much the same logic that MVS uses to process disk labels, VTOCs, extents, and so forth. The specific functions offered are: copy Sequential or PDS member to OS/2, copy OS/2 to Sequential or PDS member, superzap, rename data set, list VTOC, list PDS directory, and rename MVS volume.

Contact ASTCO Ltd., PO Box 10826, Rockville, MD, 20850; 301-469-5429 or 301-424-9455; astcoltd@erols.com for prices and details.

While there are many uses for such a product, the most apparent is to correct PARMLIB (or VTAMLST or another critical library) errors that are preventing an IPL. The alternative may be a length restore and rebuilding process if the backup was not current.

6.40 OS/2 does not see my CD-ROM drive

If you install OS/2 (Server, Advanced Server, Warp), and if you have a RAID system, you may find that the installation process cannot find your CD ROM drive partway through the installation. This is because the standard installation diskettes do not include the driver needed to use a CD-ROM drive attached through a RAID adapter.

There are several ways to proceed. The most basic is to copy IBMRAID.ADD from your RAID service driver diskette onto one of the OS/2 installation diskettes and add BASEDEV=IBMRAID.ADD to the CONFIG.SYS on the installation diskette. However there are several considerations, depending on which model of P/390 you have and which RAID adapter you have.

This question is addressed at length in the P/390 Hints and Tips (FAQ document. This document is required reading for all P/390 owners, and can be found at <ftp://p390.ibm.com>. When you access this site, go to the DOC directory, and then obtain FAQP390.PDF. (It is also available in PS and HTML formats.)

6.41 How to disconnect a token ring

You should think before disconnecting a token-ring cable. Try to observe the following guidelines:

- ▶ Token ring connections with the IEEE connector (the big, clunky connector) can often be disconnected while the system and token ring are operational. The connector is self-shorting and acts like a loop plug. Disconnect the IEEE connector, not the adapter connector (the 4-pin D-shell at the computer end).
- ▶ Do not disconnect the computer end of a token ring cable while the system is operational. The results range from a trap or OS/2 hang to normal operation, depending on your software level and the phase of the moon.

- ▶ Do not disconnect either end of a token ring adapter cable that has a telephone-type connector (RJ-45) at the LAN end. This has the same effect as disconnecting the computer end of the adapter cable because there is no automatic shorting to present a loop circuit.
- ▶ If you want to operate with the token ring disconnected, remove it from your MPTS and P/390 configuration. If the P/390 attempts to start the token ring and it is not connected to a ring (or does not have a loop plug inserted), the system will beep at you every 30 seconds. Furthermore, MPTS may appear to hang the system for a second or so, every few seconds, while it attempts to connect to the token ring.
- ▶ Obtain a token ring loop plug. Use it (before booting OS/2) if you want to temporarily operate with token ring configured (MPTS, P/390) but not attached to a ring.

In this sense, Ethernet is easier to deal with; you can simply disconnect the cable and not (normally) have any bad effects on OS/2. (For both cases, token ring and Ethernet, we assume you do not have active LAN sessions when you want to disconnect them.)

6.42 Should I use HPFS386 for my P/390

In earlier years IBM recommended the use of Warp Server Advanced Version 4 as the base operating system for a P/390. A primary attraction of Warp Server Advanced was the HPFS386 file system. HPFS386 can use large amounts of PC memory as a disk cache, and this was seen as a potential boost for P/390 performance. For a variety of reasons, IBM no longer recommends using HPFS386 for the P/390.

You may make better use of extra PC memory by using it as S/390 expanded memory. VM, in particular, uses this effectively. MVS will use it effectively to lower paging rates; if your paging rate is almost zero (common in many P/390 situations) the addition of expanded memory has less impact. The use of PC memory for expanded memory is controlled in the PF4 section of the P/390 configurator.

Read the latest DOC files first, because you need to plan the amount of OS/2 memory to use and you must set up CONFIG.SYS correctly.

If you want to use HPFS386 with your P/390, we currently recommend the following:

- ▶ Use OS/2 Warp Fixpack 35 or later. (Use a later one, if available. Some early releases of Fixpack 35 had bad copies of IPSRAID.ADD.)
- ▶ Use LAN Server Advanced Fixpack IP08508 (IP_8508) or later. (IP08506 is a prereq for IP08508).
- ▶ In the C:\IBM386FS\HPFS386.INI parameters, set MAXHEAP=4000, CACHESIZE=xxxx (see next step), and USEALLMEM=YES. (Do not use USEALLMEM unless both CACHESIZE and MAXHEAP are set correctly.)
- ▶ For the CACHESIZE:

TOTAL PC MEMORY	CACHESIZE
64 MB	12 MB
96 MB	36 MB
128 MB	60 MB
192 MB	108 MB
256 MB	156 MB
512 MB	384 MB

These numbers assume no PC storage was used for S/390 expanded memory. If some storage was used for expanded memory, subtract that amount from the CACHESIZE storage.

If you use HPFS instead of HPFS386, verify that the correct parameters are set in your CONFIG.SYS:

```
IFS=C:\OS2\HPFS.IFS /CACHE:2048 /CRECL:64 etc...
```

This says to use 2048K for a disk cache (this is the maximum that HPFS can handle, unfortunately) and cache records up to 64K. The 64 is critical; without it your P/390 performance will suffer badly.

6.43 How to determine the level of your P/390 programs

Open an OS/2 window. Enter the command SYSLEVEL. This will display several pages of output. Examine it carefully, looking for the line

```
D:\P390\SYSLEVEL.AWS IBM P/370 & P/390 Support Programs  
Version 2.05
```

....

Version 2.05 (often referenced as 2.5) was the current version at the time this was written.

6.44 How to remove PCOM's printer startup

The IBM Personal Communications 3270 product that may be used to provide 3270 emulator sessions for PCs is informally known as PCOM or PCOMM. When it starts, it may try to start a default printer. If you do not have a printer, or the printer is not configured for PCOM, this can be annoying.

You can add the line:

```
IgnoreWCCStartPrint=Y
```

to the 3270 emulator profile to prevent this. The profile usually has a suffix name of WS. It might be found in \PROGRA~1\PERSON~\PRIVATE and typically has a name like A.WS or B.WS, where the A or B corresponds to your 3270 session name. Depending on the installation procedure and the PC operating system you are using, you may need to search for the PCOM profile(s).

Once you find the profile, edit it and place the above line in the [!t] or [print] or [telnet3270] sections of the profile (or in all three; extra appearances appear to be harmless). You might want to add it to all the WS profiles if you have several.

6.45 How to use PC memory as S/390 expanded storage

The recent releases of the P/390 support programs offer an option to use PC memory as S/390 expanded storage. PC memory has become relatively inexpensive, and 256 MB or even 512 MB of PC memory is not unreasonable. There are two steps involved in using this memory as S/390 expanded memory:

- ▶ Edit your CONFIG.SYS and find the AWS370DD driver statement. Change it like this:

```
DEVICE=D:\P390\AWS370DD.SYS /X=nn
```

where nn is the number of megabytes of PC OS/2 memory to use as S/390 expanded memory.

- ▶ Go to the F4 option of the P/390 configurator. Set the Expanded Store Size to the same nn value. Select the P option on the same line. (You cannot allocate part of the P390 adapter storage to expanded storage if you use PC memory for expanded storage.)
- ▶ You will need to reboot to pick up the CONFIG.SYS change.

Experience has shown that, for a P/390 adapter, using PC memory for expanded storage is about ten time slower than using P/390 adapter storage for expanded storage. (This is still much faster than paging to a disk.) For a P/390E adapter, using PC memory is only 15% slower than using adapter storage.

This option is available for P/390 and Integrated Server systems, but not for R/390 systems. It is available only with the PCI adapters; it is not available for Micro Channel P/390 adapters.

How effective is it? The answer depends on your MVS workload. If your normal paging rate is close to zero, you may not gain much. IPL will be faster, because there is considerable paging during IPL. If your paging rate, during important periods, is above a few pages/second, you should see a useful benefit.

How much PC OS/2 memory to assign to this function? Reserve at least 32MB for OS/2. If you are using HPFS386, subtract whatever memory it takes for its cache. (HPFS386 obtains its memory before AWS370DD.) How much expanded storage does your MVS need? There is no way for us to answer this. We suggest you try this option if you can give at least 32 MB for expanded storage.

6.46 How to check LAN3274 connections

The command AWSLANUL (issued from an OS/2 window of a P/390 system or MP3000) will list current TN3270 connections to the AWS3274 device manager. (The programs displays an error if there are no connections.) This can be very helpful in debugging problems with complex DEVMAPs.

6.47 Communication line adapters for MP3000

The following table outlines the BSC and SDLC adapters available for emulated I/O operation in an MP3000 system.

Protocol	Adapter	Device Manager	Operating System	Emulating:
BSC	WAC	AWSICA	VM, VSE	ICA
SDLC	WAC	AWSICA	VM, VSE	ICA
SDLC	WAC	WAN3172	MVS, VM, VSE	3172
SDLC	Multiport -2	WAN3172	MVS, VM,VSE	3172

The interfaces can be RS232, V.35, or X.21 for the WAC adapter, and RS232 (up to 38.4 Kbps) for the Multiport adapter. WAC is the common term for the Wide Area Connector adapter. ICA is the common term for the Integrated Communications Adapter found on older S/390 machines.

The Multiport Adapter can handle 8 lines at 9600 bps/line, or 4 lines at 19200; its limitation is total bandwidth, not per-line bandwidth.

6.48 How to access more PUs for LAN3172

The LAN3172 device manager (MP3000, P/390) is used to access SNA remote units. It defaults to a limited number of LINK stations. A link station is normally a PU. The maximum number of LINK stations is 255 per LAN adapter. You can change the default by editing C:\IBMCOM\PROTOCOL.INI in OS/2:

```
.....
LINKS = 8,8,8,8           <== default in our OS/2
LINKS = 255,255,255,255  <== change to this
```

The four numbers are for four LAN adapters (the maximum for normal OS/2 use). You may not have four LAN adapters, but all four parameters are specified.

6.49 3490-F01 device settings

The IBM 3490-F01 SCSI tape drive has a number of parameters that are set using switches on the drive. To use this drive with the MP3000 or P/390 systems, use the following settings:

Option	Value	Comments
31	Disp WRT format	
44	Sync 2 FMK	write synchronization
46	32	number of read retries
47	16	number of write retries
48	Host/panel unlocked	host and front panel active
49	Variable	block size
50	ID 5	SCSI ID <==== set as needed
51	Yes	interface parity enabled
52	Dflt dev (80h)	Overland 3480/3490E (real device type)
53	No	ready signaled during rewind
54	No	synchronous data transfer
55	Std	EOT mode reporting
57	No	enable twos-complement residue
59	Default S/N	not used
60	Overland	vendor ID
61	L490E	product ID
62	3490E	format selection
63	SCSI-2	interface option
65	18-track write unlocked	
66	Status on unload	signal status on unload operation
67	Retry MTW errors	not used
68	Sync 5 MB/sec	default SCSI transfer rate
69	3490E compatible	power-up default
72	Busy	SCSI response during rewind
74	Read to EOD	
75	Auto clean OFF	
76	Pack after RTYS	
77	18-track write OFF	initial state
78	No medium type	
79	Standard mode	cartridge sensing
88	0	

The option number is used when setting parameters through the device switches. Option 72 is available only if you have firmware level *flash 15* or later. Options 48, 62, and 69 must be set as shown if you intend to use the multidevice support for writing 18 and 36 track output tapes.

6.50 How to improve emulated printer performance

A P/390 or MP3000 can use a PC printer, attached to the PC parallel port, to emulate a S/390 line printer. The OS/2 printer drivers can usually run in either of two modes. (We say “usually” because there are a number of printer drivers and they do not all work the same way.) The default mode is for OS/2 to frequently poll the printer and ask if it needs more data. This can drive the printer faster, but can use an excessive number of PC processor cycles. The alternate mode uses interrupts from the printer to trigger more data to the printer.

The interrupt mode should be superior but there is a history of PC printers not working well in interrupt mode, for a variety of reasons.

If your system runs much slower when you are printing, try switching to interrupt mode. This is done by editing CONFIG.SYS. Find the PRINT01.SYS statement and add /IRQ as a parameter:

```
BASEDEV=PRINT01.SYS /IRQ
```

You will need to reboot to make the change effective. This should improve system S/390 performance while OS/2 is printing. If the printer hangs or is very erratic, you should examine the interrupt assignments on your system.

6.51 How to install a second EtherJet adapter

Installing a second EtherJet adapter requires additional information. In our particular case, we were working with a PC Server that had an EtherJet-equivalent adapter on the system board. When we installed a PCI EtherJet adapter, this adapter became the second EtherJet adapter. (The adapter we installed was 10/100 EtherJet PCI Adapter with Wake on LAN. We ignored the Wake on LAN feature.)

As part of the installation process, we used the *Other Adapters* function of the *Network Adapters* window in MPTS to read the device driver diskette supplied with the adapter. This updated the driver for MPTS.

In MPTS, the adapter was added to the Current Configuration window. We intended to use it for OS/390 TCP/IP. We added the IEEE802.2 protocol to the adapter (in the *Current Configuration* window) and changed the frame protocol to DIX. If you double-click on the adapter name in the *Current Configuration* window, the popup window contains a field for a Slot/Device number. You must supply the correct number or you will have errors during OS/2 startup. Your first (original) EtherJet adapter also now needs a Slot/Device number. You must specify both of these in order for your LAN adapters to work correctly. (Both numbers must be specified in hexadecimal.)

We were unable to obtain useful help information from MPTS. If we booted OS/2 (without specifying slot numbers) we had error messages suggesting the slot numbers might be 0x03 and 0x0E. The README files on the diskette suggested booting OS/2 with no LAN drivers active and then running a utility on the diskette. Changing OS/2 this way was not attractive. Instead we booted a DOS system and executed the SETUP utility on the adapter driver diskette. This ran a variety of tests confirming the adapters worked. It also specified the adapter slot numbers as 3 and 14 (decimal). By running the adapter tests and watching lights on an Ethernet hub, we could determine which adapter was which. (Slot 3 was the integrated adapter and slot 14 was the new adapter we just installed.)

We specified these numbers (3 and E, in hexadecimal) in MPTS and the adapters then worked without problems. In our setup, the integrated adapter is LAN adapter 0 and is used for OS/2 TCP/IP. The new PCI adapter card is LAN adapter 1 and is used for OS/390 TCP/IP. Both adapters are capable of 100 Mbps operation, but our hubs are 10 Mbps and we did not try 100 Mbps operation. We did not specify a speed in MPTS, so we assume they automatically sense the LAN speed at startup (probably making OS/2 startup a little slower).

6.52 Multiple Ethernet adapters for MP3000

MPTS (using an OS/2 window) can be used to force the speed (10 or 100) and duplex mode (full duplex = 2 or half duplex = 1) for an Ethernet adapter. Most users leave these fields blank and let the adapter automatically set itself. We have had a few reported cases where it was necessary to set explicit values for these parameters. In at least one case, this was necessary because there was no other traffic on the LAN and the adapter was unable to find a reference packet in order to set its own parameters.

6.53 How to unlock MPTS

If something goes wrong with an MPTS installation or update, you may not be able to use MPTS. It will produce messages about *locked files* and suggest a reboot to clear the locks. If rebooting does not solve the problem, search for files named IBMLANLK.LST or IBMCSFLK.LST and delete these files. Rebooting should then resolve the problem. This problem can occur with P/390, Integrated Server, or (in rare cases) an MP3000 system.

6.54 LAN adapter plugging for MP3000

LAN adapters in the MP3000 must be placed in a certain order. The first adapter goes in slot 12, the second in slot 11, and so forth. A maximum of four adapters may be used, and they go in slots 12 - 9. Ethernet adapters (if any) must be installed first, followed by token ring adapters (if any). A maximum of three Ethernet adapters may be installed; in conjunction with the built-in Ethernet port, this provides four Ethernet interfaces. If four token-ring adapters are installed, the built-in Ethernet interface must be disabled in the system BIOS. If a mixed LAN set (Ethernet and token ring) is used, a maximum of three adapters may be used plus the built-in Ethernet interface.

6.55 Canonical form of MAC addresses

Some setup functions want the canonical form of token ring MAC addresses and others want the non-canonical form. Token ring MAC addresses are usually non-canonical and Ethernet addresses are often canonical. What does this mean? *Canonical* means a “primary” or “legal” or “sanctioned” source and implies that this form is more fundamental in some way. (We do not know who made the decision that a certain MAC format is more primary or fundamental than the other.)

If a token ring MAC address begins 40... then it is in non-canonical format. If it begins 02... then it is in canonical form. To convert, take each byte (two hex digits, working left to right) and read the bits “backward” or right to left in that byte. For example:

Canonical Hex	Canonical bits	Reversed bits	Non-canonical
403B	0100 0000 0011 1011	0000 0010 1101 1100	02DC

Ethernet MAC addresses are usually given and used in canonical format and no conversion is needed. We know of no absolute rule here, and apparently some LAN adapters do address conversion automatically. A standard technique when setting up an SNA LAN definition in VTAM is to try the MAC address both ways.

6.56 Understanding mainframe printing

A requirement to have something printed on paper exists in all environments. This chapter addresses some of the ways to print with an OS/390 system, dealing with both traditional line printing and Advanced Function Presentation (AFP) printing. While these concepts are well understood by traditional MVS users, they may be new to someone moving to OS/390 from a UNIX or PC environment.

To understand some of the printing issues, we must discuss printer data streams. The original printer for the S/360 was the IBM 1403 Printer, a relatively fast impact line printer. It used a data stream that is still used by many programs, including all the system utilities, compilers, and so forth.

The 1403 data stream:

- ▶ Is a simple character record, with a maximum of 120 or 132 characters. The extra 12 character positions for the printer were an option on early 1403 printers. To this day, many utilities create 120-column output because the original, base 1403 printer had 120 columns.
- ▶ Has a *carriage control character* that can be used to skip to the top of the next page or to skip to any of several other positions on the page that are indicated by the current carriage control tape. An operator can change the carriage control tape, and different jobs might call for different carriage control tapes. The different positions on the carriage control tape are known as channels. On a give tape, channel two might be used to make a half-page skip. Later printers implemented these functions using Forms Control Buffers (FCBs), but the carriage control tape terminology remains.

How to create FCBs is described in the manual MVS DFP Utilities SC26-4559. FCBs are stored in the system library called SYS1.IMAGELIB.

- ▶ Later 1403 printers had a Universal Character Set (UCS) option. This permitted a job, by *loading UCS* to equate any byte value to a given position on a print chain. For example, a byte containing 'x'C1' is usually printed as an A, but this could be changed through the UCS option. UCS is particularly useful for non-English languages.
- ▶ If a line longer than 120 (or 132) characters is sent to the printer, the excess is simply thrown away. There is no automatic wrapping of lines.
- ▶ If data lines, in the most basic mode, with no other control options specified, are sent to the printer, they are simply printed and followed by an automatic skip to the next line. Data will be printed across page separations and so forth. JES2 contains a parameter intended to skip over page breaks if a print stream contains no controls of its own.
- ▶ Software (OS/390 and all other S/390 operating systems) can be told to regard column 1 of every output line as a carriage control byte. This is not a requirement. Such controls are indicated in DCB parameters by an A (ANSI) or M (machine). The exact details of ANSI and machine control characters can be found in the manual DFP/MVS Macro Instruction for Data Sets, SC26-4747.
- ▶ Using ANSI controls (with a DCB parameter such as RECFM=FA), a 1 in the first column of output means to skip to carriage control channel 1 before printing the line. A blank as a carriage control character means to print with single line spacing; for this reason, a blank is the most commonly used control character. By convention, channel 1 always means the

top of the next page. A 2 means to skip to whatever position is indicated in channel 2 of the carriage tape or FCB; there are no standard conventions for anything than channel 1 (top of page) and sometimes channel 9 (almost at the bottom of the page).

- ▶ The carriage control character is not actually printed. JES2 (or some other printer program) strips it from the data and converts it into the CCWs needed to do the job. A *full length* 1403 line, in DCB=FA form, would have 133 bytes; one byte for the control character and 132 bytes of print data.

The control character may be omitted in the data stream, in that case the printing, depending on the printer or the system, either prints continuously, even over the forms border, or skipping is based on the line count set up in the system parameters. By omitting the control character, the file is a *flat file*, no controls imbedded. It is similar to a flat ASCII file in the PC environment. In the PC environment, the functions of the 1403 data carriage control byte are available in some control characters and ESC sequences, for example, Form Feed is used for the same purpose as “skip to channel 1” in S/390 line printer operations.

With the introduction of the IBM 3800 Page Printer, another optional control byte, Table Reference Character (TRC), was added after the carriage control character and before the the data bytes. The TRC is used only when printing line data, and is not used for printing page-mode data. The purpose of this character is to control the font used to print the line. In the IBM 3800-1 printer, it was possible to use only one to four different fonts in one print data set. When this function was transferred to AFP, the choice became larger. TRCs were used for a relatively short time and are seldom encountered today.

When used in the IBM 3800 way, there is a parameter in the MVS JCL to handle this font function. The user may specify one to four fonts in the CHARS keyword in either OUTPUT JCL statement or DD statement for a data set. The format is CHARS=(FON0,FON1,FON2,FON3), where fonts FON0-3 are coded fonts that are to be used when TRC 0,...3, is specified in TRC respectively. To make the system understand that the second byte is not data, but TRC, you have to use some parameters, either you specify TRC=YES in the DD statement, or you code the subparameter OPTCD=J in your DCB for the data set.

The IBM 3800-1 printer was a transitional step between previous line printers and future page printers. It is not considered to be a full member of the AFP family (and few 3800-1 printers still exist). In general, all other IBM 38xx and 39xx printers are AFP printers, although some can also be used in other modes. The most general characteristic of an AFP printer is that it is a page printer instead of a line printer.

When printing line mode data sets on an AFP printer, the TRC byte is still optional, but you can use in the same way or even extend its use to allow more fonts. If the TRC is used in the new way provided by AFP, the JCL above works as long as only four fonts are needed. If more fonts are required, these have to be specified in a an AFP resource called page definition. In that case the TRC must be between hexadecimal '00' and '7E' and in the page definition (see later) these TRC characters are mapped to fonts.

6.56.1 AFP data streams

With the coming of AFP printers, a new type of data stream was needed to have more control of the functions available in the printers. The structure of an AFP Data Stream record is the following:

- ▶ Column 1: instead of a carriage control character as in the 1403 data stream above, this is always hexadecimal '5A'.
- ▶ Columns 2-3: the length of the remaining record including these length bytes.
- ▶ Columns 4-6: AFP data stream command indicating the nature and purpose of the record.

- ▶ Column 7: flag byte is used for some special records, see details in the AFP Data Stream Reference.
- ▶ Columns 8-9: sequence number of the data stream record. Can be anything, but as this is used in error messages, it is better to make these sequence numbers unique.
- ▶ Columns 10-n: AFP data, depending on the AFP record, it is either commands and text or additional information. It is all documented in great detail in the AFP Data Stream Reference manual.

In a 1403 data stream, the data is line by line. In AFP printing, it is page by page. In general a record represent a page, although it is possible to use multiple records to produce a single page.

An AFP data stream needs Print Services Facility (PSF) for printing. PSF is an optional program product, and versions exist for OS/390, VM, VSE, AIX, OS/2, and so forth. PSF is able to handle 1403 data stream printing on an AFP printer as well. It does this by converting the line-mode data to page mode, using a default conversion that you can control. In most OS/390 installations, much of the printed output is handled this way. For example, a COBOL compiler generates a listing in 1403 mode. JES2 routes this listing to PSF, which converts it to page mode and prints it on a 38xx printer.

PSF is included with the AD CD-ROM OS/390 systems. It is not part of basic OS/390, but is an additional program product. As a practical matter, almost all OS/390 installations have the PSF product.

AFP data stream files are sometimes called LIST3820 files referring to one of the early AFP printers.

6.56.2 How to print AFP data stream (LIST3820) files

To be able to print AFP data stream data sets you have to have the *resources* referred to in the data set. AFP *resources* are usually fonts, although images and other controls can also be considered AFP resources. The most common thing missing is one or more fonts that are required by the AFP data stream data set. Given that you have the resources needed, the printing itself is easy.

You may use the IEBGENER system utility to copy the file to a printer, SYSUT1 DD refers to the input data set, SYSUT2 DD to the output. The only thing that may be less obvious, is that the DCB for the output file has to use either ANSI or machine control characters. You specify this by entering as 'A' or 'M' the last letter of the RECFM subparameter, for example, VBA or VBM. If the input data set's DCB already has this, everything works fine.

If you omit the A or M parameter, and your input data set does indicate the control character, the hexadecimal '5A' in the beginning of the data record is considered to be data and PSF does not recognize that the file is an AFP file. Then you will get text that looks more like garbage than anything else on your printed page.

What are AFP resources?

AFP uses special resources to provide advanced functions. These resources are:

- ▶ Formatting resources
 - Form Definition - is a resource mostly used to control the physical form that is used for printing. Functions defined in the form definition are:
 - Which paper source (if multiple sources are available) is used.
 - Is the printing only one on side of the paper (simplex), or on both sides of the paper (duplex or tumble depending on the edge around which the paper is turned).
 - Use of electronic overlays; you can include up to 8 electronic overlays per side.

- Copy count of similar sheets of paper -- how many copies of single sheet are produced. To produce collated copies, in which the whole data set is printed and then printed again, you must use the COPIES keyword in your JCL OUTPUT statement.
 - Positioning of the composed page, created by an application or by PSF when line data is formatted using a page definition, on the paper
- Page Definition - formats line data on a page. This resource is not used for AFP data stream files, as they are already formatted as composed pages. For line data sets printed on an AFP printer, a page definition is needed; it is either the one specified in the PSF startup procedure for that printer, the one set up as FCB in JES, or overridden by a JCL OUTPUT statement PAGEDEF parameter.

Within a Page Definition you can specify:

- Positioning of the line data on a page. Lines may be placed as whole entities, or the line may be split into fields the fields placed. This includes information related to the skipping to channels as described in the 1403 data stream. You specify in the page definition where to skip on the page when a skip to a certain channel is detected in the line data. Another function achieved with page definition is *multiple-up* printing. If skip to channel 1 is used to skip to a new page, by specifying in the page definition multiple places on the page where to skip with channel 1, you get more than one original application program output page to be printed on one physical page.
- Which font is used for printing. You can specify this by line or by field, or you can specify that all text is printed with the same font. If the page definition does not specify fonts, then the initial setting in the PSF startup or the one you specify in the CHARS parameter is used for all text.

You call for these resources by coding PAGEDEF and FORMDEF keywords in the JCL OUTPUT statement. In the OUTPUT statement, you use the resource names without the prefixes AFP attaches to them. If you code, for example, FORMDEF=A10110 and PAGEDEF=A06462, then PSF will use form definition F1A10110 and page definition P1A06462 respectively. This dropping of a two-character prefix is always confusing to PSF users and we do not know why AFP/PSF development produced this result.

However, this operation has been in use for many years.

- ▶ Electronic overlays are resources that contain constant text and may also contain images and graphics that are being printed on all (or a subset of) pages, as specified in the form definition. Overlays can also be included by proper coding of an Include Page Overlay (IPO) record within line data.
- ▶ Page segments (PSEGs) are pictures or graphics, most typical of which are company logos, signatures, and charts. They may be called by overlays or included by inserting a special Include Page Segment (IPS) AFP record in a line data stream.
- ▶ Fonts are needed if the page has text. Fonts are called by overlays and page definitions, or they can be specified in the CHARS parameter. As most of the AFP font names are longer than four characters (even without counting the X0 prefix) and as the maximum length you can specify in CHARS parameter is only four, you may have to copy or rename an AFP font to have only four letters to be able to use it in the CHARS parameter. If you code CHARS=(FON0,FON1,FON2,FON3) and the printout is going to an AFP printer, the coded font names accessed are X0FON1-4 respectively. If you want all your page to be printed with one and same font, you code only one font in the CHARS keyword.

This may be required if you use page definitions that have no font requests coded in them. In this case, the printout would use the default font which might be wrong for your purposes. For example, if you use the standard page definition W240F3 that indicates 4-up landscape printing, you have to use either 20 characters per inch (for example, GT20), or 24 characters per inch (for example, GT24) to make all the text fit on the area

An example of JCL to print a listing 4-up and duplex and using GT20 characters is.

```
//FI48590A JOB 1,MARKKULA,MSGLCASS=X
//COPY3820 EXEC PGM=IEBGENER
//OUTPUT1 OUTPUT PAGEDEF=W240F3,FORMDEF=A10111,CHARS=GT20
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT2 DD SYSOUT=A,OUTPUT=*.OUTPUT1
//SYSUT1 DD DSN=FI48590.LISTING,DISP=SHR
/*
```

How do I create AFP data streams?

AFP data stream are created with a number of application programs. The Document Composition Facility (DCF), or its follow-on products (often collectively known as Script) was a common source of AFP printing; this series of products is seldom used in OS/390 environments today.

An easy way to produce AFP data stream files is to use WINAFP in Windows or the respective driver in OS/2. These drivers allow you to create an AFP document, an AFP overlay, or an AFP page segment from any Windows or OS/2 program. The format for how AFP records are stored on a PC disk is different from what it is in OS/390. In a PC, the file is one entity; in OS/390, it is a set of variable length records. You must be careful when transferring the file from the PC to the host, because using CRLF option in the transfer may destroy it. If you transfer the file with no CRLF, you need a program or a REXX EXEC to split the resource into records in OS/390. In AFP redbooks (for example, *AFP Resources in a Cross-System Environment*), there are examples of REXX EXECs to do this.

An AFP document can be printed directly by PSF if it has the right fonts and so forth available. An overlay, or a page segment, cannot be printed directly. They must be included in a complete AFP printable file by using form definitions and, in the case of a page segment, either referring to it in an overlay or using IPS to include the segment. file

All AFP resources are in AFP data stream format; there are tools for creating AFP data stream for the resources. Tools are available in the OS/390 environment, but the tools in the workstations are more popular.

You can use a trick to get a kind of AFP data stream, or, at least, get different fonts for different lines. The TRC, the byte indicating the font that will be used for the printing of the line, can be manually edited in a text file. You would code carriage control information in the first column in the data records of your file and then a character 0,1,2, or 3 in the second column, and from the third column onwards you enter your data. Then, defining in your JCL that the file has control characters and TRC characters and by coding the keyword CHARS in the OUTPUT in the way you want, you can have up to four different fonts in your printout.

If you have created a nice picture on your workstation, created an AFP page segment (PSEG) from it, and transferred this to your OS/390 system (avoiding CRLF conversion, but making the necessary record segmentation), you can include it in your printed output. You may include the page segment on your printed page by entering an IPS command in your text data set. The format of the IPS AFP record is found in the AFP Data Stream Reference.

Here is a very simple example:

Character representation	S1MYSEGM
Hexadecimal representation	501DA5000EFDEECCDFFFFFFF A063FF00021468574FFFFFFF
column	1...5...10...15...20...

This record would include a page segment named 'S1MYSEGM' at the place where the printing is going on the page at the moment this record is encountered in the output data file. This method is fine, for example, for including a signature at the end of a letter. You would space a couple of lines after the last line of the text and then, with the IPS command, insert the page segment containing the signature. To place the segment in a fixed place on the page you simply substitute the six hexadecimal 'FF' bytes with the binary value of the horizontal and vertical distance from the top left corner of the page.

For example, to place a segment 100 pels (240 resolution pels) right from the left margin and 200 pels down from the top of the page, first, you have to convert numbers 100 and 200 to hexadecimal. They are '64' and 'C8'. So, you replace the 'FF's with '0000640000C8'.

To include an overlay in the same way, the overlay has to be mentioned in the page definition, so you have to create a page definition to do this. We do not go into details with this, if you really need it, you can find more information in AFP manuals,

6.56.3 What else do I get with AFP?

Although you may not have the tools to create AFP resources in your system, and maybe you do not have applications to create AFP data stream documents, you may still have some benefit of AFP. One of the nice things is printing multiple-up output, as described earlier. By combining multiple-up printing and duplex printing, you may reduce the amount of paper sheets printed significantly. This saves printing time, storing space, and paper costs. And, if you have to send the document, also shipping costs.

There are several standard page and form definitions available. For example, a page definition named P1M120C0 describes two application pages, 60 lines each, on one AFP page, printed simplex. The pages are positioned so that the first one takes the top half of a portrait 8.5x11 page and the second page takes the bottom half. Form definition F1A10111 defines duplex printing, using the printer's primary input bin as the paper source.

Using these resources in your JCL OUTPUT statement, you make your application printout, a compiler output listing, for example, print two application pages on both sides of the paper, reducing the amount of paper used to one quarter of the original. To do this, you must use a font that is small enough; for example, GT15 should be small enough.

6.56.4 Fonts

You do not need fonts when you print your output on a line printer or an emulated line printer. If, however, you are going to print your output on an AFP printer, or IBM 3800 printer with AFP, you need some fonts to print.

The IBM 3800 printer has two different ways of operation. The old 3800-1 was really a line-mode printer; the newer models 3800-3 and 3800-6 could be used like 3800-1 or as an AFP printer. When using the 3800 printer in line mode, the printer is driven by JES2; when using an AFP 3800 printer, you need PSF/MVS to drive the printer. To use 3800 printer (either 3800-1 or the new models) as a line data printer, you must have appropriate character sets available in SYS1.IMAGELIB. You can even create your own character sets with a utility shipped with the system software. This utility is IEBIMAGE. More information about IEBIMAGE is in the DFP/MVS Utilities.

If you want to use an 3800 AFP printer, you need fonts that are unique for the 3800. With OS/390, the following fonts for 3800 printing with AFP are available:

- ▶ The SYS1.FONTLIB library contains the most common fonts to be used for replacing traditional line printing.

- ▶ The SYS1.FONTLIBB library contains 240 pel fonts. The most common 240 pel printers are 3812, 3816, 3820, 3825, 3827, 3835, 3900, 3130 in 240 pel resolution, and 3160.
- ▶ The SYS1.FONT300 library contains 300 pel fonts. The most common 300 pel printers are 4028, 3112, 3116 3912, 3916, 3130 in 300 pel resolution, and 3935.

If you are not using 3800 AFP printers, or you are going to use them as 3800-1 (line mode) printers only, you can delete SYS1.FONTLIB; you do not need it. If you use only printers of the same resolution, you can free some disk space by deleting the unused library.

6.56.5 How to deal with different printer resolutions

AFP printers use different printing resolution, either 240 or 300 pels (dots) per inch. There are font resources available for both resolutions. In addition, PSF/MVS includes a conversion program to convert 240 pel fonts to 300 pel fonts.

When you refer to a font in CHARS keyword, you, in fact, refer to an object called a *coded font*. A coded font is simply an object to tie together a font and a code page.

A font is an object where you have the names of the characters (upper case 'A' being LA020000, lower case 'a' LA010000 and so on) and the graphical pictures of the respective characters. A code page is an object where you map hexadecimal codes to character names (for example, hex 'C1' is mapped to LA020000, hex '81' is mapped to LA010000 and so on)

Let us take an example. A coded font named GT10 can be found in one or more of the OS/390 font libraries. In fact, the name of the member in the library is X0GT10, where the X0 prefix indicates that this is a coded font resource. If you browse that resource, you will find two names, C0D0GT10 and T1D0BASE inside the file. In this, C0D0GT10 is the character set or font, both terms are used to mean the same object. This object contains the graphic representations of the Gothic 10 character per inch characters with their names. The other object referred to is T1D0BASE, the code page mapping a standard US character set to the character names in the character set.

There are multiple coded fonts referring to the same character set and different code pages allowing the user to use the appropriate national character set coding. For example, which you might use GT0E in the CHARS keyword, refers to the coded font X0GT0E, which then refers to the same character set described above and a code page T1DEBASE that is a Finnish/Swedish codepage.

6.56.6 How to consider NLS printing

One of the major problems with printing seems to be the printing of national language characters correctly. Depending on whether you are printing on a line printer, or on an emulated line printer, or on an AFP printer, the problem is handled differently.

When printing on a line printer, you have to make sure that the printer is loaded with the right character set (UCS) and the print train or belt has those characters available. Remember, a UCS is an object that is used to tell the printer which hex code is going to represent which character. There are probably a suitable UCS available in the system or you can create one. UCS modules are stored in the library SYS1.IMAGELIB. For further details, see the manual *MVS/DFP Utilities*.

If you are printing on an emulated line printer, such as a PC printer with a P/390, you may have to use `tr` commands to have the characters translated to correct printable characters.

The easiest task is when you are using an AFP printer. By design, AFP takes care of national language differences, so you just have to pick the right coded font and your output will be printed in the proper way.

6.56.7 How to print a file In OS/390

There are several ways to print in OS/390. The basic idea is that you have an application program, either one supplied with the system software or other software package, or a program written by yourself. Whichever it is, the application program creates a print file either printing it directly on a printer (very seldom done) or spooling it to the JES spool. From the JES spool, either JES prints the file on the printer, or in the case of a job to be printed on AFP printer, PSF takes care of the printing on the printer.

The OS/390 system comes with a large number of utilities you can use for printing. The most common utility you might want to use is IEBGENER. It is basically a utility that copies a file to another place. In the printing case, IEBGENER copies a file to printer. IEBGENER DD statements allow you to enter parameters to control printing; for example if the input data set includes carriage control characters, but the file was originally created with no carriage control characters, you can change the file characteristics by entering the DCB information on the DD statement.

Most of the ISPF functions have printing services included in one way or another. When you issue a print command, ISPF puts the printout to a temporary file. At any time, or when you exit ISPF, you may print this temporary file. At any other time you can issue a LIST command on the ISPF command line, and the panel for controlling printed output disposition is displayed. On that panel you can specify where you want to have the file printed, which print class and so on.

6.56.8 Printing from CICS

Usually, CICS applications print their output on a VTAM/CICS controlled printer, so that, in terms of OS/390, no action is required. Sometimes printing on a centralized printer or on an distributed AFP printer is wanted. There are several ways to do this. In JES2 environment, CICS has tools to write directly into the JES spool (SPOOLOPEN, SPOOLWRITE and SPOOLCLOSE). In general, printing from CICS requires a application program (transaction program) that was designed to do the printing, as opposed to TSO, where various standard print interfaces can be called from the terminal.

In both JES2 and JES3, it is possible to create a job in CICS to initiate a print job by writing it to a data set and then, after the data set is closed manually, or automatically at CICS shutdown, the data set is sent to the internal reader.

You can also write to a transient data set. Either when CICS is shut down, or during the time CICS is running, by closing the transient data file, it is directed to the printer as specified in the CICS startup JCL.

With PSF/MVS 2.2.0, the optional feature NETSPOOL is available. It allows you to print any file that was directed to a VTAM printer, to an AFP printer instead. Although the application still thinks it is printing on a VTAM printer, what really happens is that NETSPOOL, according to the setup parameters, grabs the print file and puts it into JES spool. From the JES spool, the file is then printed by PSF/MVS.

6.56.9 How to control line count

It is possible to specify the line count for a page in several places. By *line count*, we mean a count that determines when the next printed line will start a new page. In 1403 terminology, this is when a *skip to channel 1* is forced.

Many programs have parameters to set their own line counts, independent of any overall system controls. For example, most of the compilers allow the user to set a line count.

The most centralized point where you can set a line count is in the LINECT parameter of PRINTDEF in SYS1.PARMLIB(JES2PARM). This value is typically set to 61 in IBM-distributed systems. If a line-mode application prints more than this number of lines on a page, JES2 will force a skip to a new page. If your application has a higher line count, the JES2 count and the application count will interact and produce odd page skipping. You can override this JES2 count, for an individual job, using OUTPUT statements in your JCL.

6.56.10 Controlling printing

After a printed output is sent JES spool by any program, the actual printing can be controlled in several ways. If you are allowed to use an operator console or you can access the console services from your TSO screen, you can use JES console commands to control printing. You can, for example, forward space or backspace the print, to skip pages or reprint some pages. You can interrupt a print job, cancel it, you can start and stop printers, change the parameters for a printer and so on.

Whether it is an AFP printer or a JES controlled printer does not impact the operation significantly.

The Spool Display and Search Facility (SDSF) provides a more user-friendly interface for some printing controls, in addition to displaying print files. However, for line data you cannot see the skipping and spacing in the same way the printout would appear on a printed page. If it is an AFP file, you can see it almost in the same format as it would appear on the paper.

6.56.11 What to do when something goes wrong

Typical errors with printing are, for example, problems with national language characters, printing off the page, getting error messages, and so on.

► AFP error messages

You get AFP error messages starting with the message prefix "APS" in many cases. Some messages may be ignored, but many must be resolved. You should review each message and also check that there is nothing missing from the page. AFP messages contain detailed information about what happened and which objects were active at the time of the error. When you become familiar with these messages, you will quickly find the reason for most errors.

What are the most common messages? The following cases are very common:

- The font requested cannot be found in the resolution that is needed for the printer. Different AFP printers use different resolutions; for example, the 3812 and 3816 use 240 pel fonts while the 4028 uses 300 pel fonts. To ensure proper printing you must have the correct fonts for each printer. The fonts in 240 pel resolution are in SYS1.FONTLIBB and the fonts in 300 pel resolution are in SYS1.FONT300. There is a conversion tool to convert 240 pel fonts to 300 pel fonts; a sample job for the conversion can be found in SYS1.SAMPLIB(APSWCF30). When you set up the start procedure for 240 pel printers, you must have the 240 pel font library in front of the 300 pel fonts in the concatenation order. For 300 pel printers, use the opposite

concatenation, of course. Coded fonts as well as code pages are independent of font resolution. You may have a common library for coded fonts and code pages, but you must have separate libraries for fonts of different resolutions.

You can try to find a missing font that is available in the correct resolution, or if the message is caused by not finding a 300 pel font and there is a 240 pel font, you can use the conversion program to create the respective 300 pel font.

Fonts are the only resources that are resolution-dependent. PSF/MVS takes care of converting page segments and overlays from 240 to 300 pels while printing. Of course, you get better print quality if you create the resources at the right resolution.

- Font resource mismatch and/or characters missing: This is usually caused by a hexadecimal code in the data stream for which there is no defined character mapping. If there is no mapping in the code page for a specific hexadecimal code, the code will be replaced using the default character set code page. Depending on an indicator byte in the code page to determine whether this either causes a data check or not. The default character is usually a space. This is convenient, because the most common cause is due to a COBOL program initializing a character field to hex zeros. low-values (hex '00') being used instead of space. The hex code '00' should usually be mapped to a space to handle this problem. In some of the code pages supplied by IBM, especially those that are designed to be used in NLS applications, unfortunately, hex '00' is not mapped properly, and further, replacing a code with the default character causes a data check. Depending on the PSF startup setup or overriding OUTPUT JCL statement keyword settings, dropping part of a line may be the only consequence and there is usually an error mark indicating dropped data. If a data check is generated, an error message is also generated. You can control whether the code page gives a data check or not when a code has been replaced with the default character.

- ▶ Missing output on a page

Another very common error is that an application creates output that is too wide to be printed with default formatting definitions. In this case, output exceeds the valid printable area that is defined as the intersection of the physical paper and the area defined in the AFP page definition formatting resource. The part of the line exceeding the valid printable area is not printed. Depending on the PSF startup setup or overriding OUTPUT JCL statement keyword settings, dropping part of a line may be the only consequence. National language characters printed incorrectly

This is caused by using the wrong code page for printing. See the description earlier.

- ▶ Pages split incorrectly

An AFP printer handles a whole page at a time. Thus, the output meant to be printed on the page either prints on a page or gets lost exceeding the page area, but never splits onto two pages. If it appears that a page, originally created as one page, is split over the page boundary, the reason is that the page definition used does not accommodate enough lines so that the all the lines of the original page fit on the page. For example, if the page definition you use specifies the maximum number of lines per page is 60 but the application program generating the output writes 62 lines before issuing a skip to a new page, then 60 lines are printed on the first page and 2 lines are printed on the following page. The application then intends to skip to skip to page 2, the skip takes the next lines onto page 3, and so forth. This problem is easily solved, either by using a page definition with more lines per page, or changing the application so that it does not write too many lines on a page.

There is a relationship between FCBs used for line printers and page definitions used for AFP printers. When you set up an FCB in the JES parameters for an AFP printer, PSF will search for a page definition matching the FCB name prefixed with P1.

6.56.12 Handling AFP errors and error messages

In the PSF procedure for each printer, you can specify keyword parameters PIMSG and DATAACK. These two keywords control whether errors are reported, whether error messages are printed, and whether a certain number of error messages will cause printing to be terminated.

If you set DATAACK=UNBLOCK, all data checks, whether they are caused by invalid characters or exceeding the boundaries, are reported to PSF. If you set DATAACK=BLKCHAR, only errors caused by exceeding the boundaries are reported, invalid character errors are blocked. If you set DATAACK=BLKPOS, invalid character errors are reported, but exceeding the boundaries is not.

By setting PIMSG=(YES,8) you tell PSF to continue until 8 errors have been reported and then terminate the print job. All messages are printed. By setting PIMSG=(NO,8) you get the print job terminated after 8 error messages, but only the last message is printed. By setting PIMSG=(NO,0), you print the print job until the end, even though what you get on the paper may be far from what the intention was. And only the terminating message is printed.

For a production environment, DATAACK=UNBLOCK and PIMSG=(YES,n) where n might somewhere between 10 and 20, should be set. This guarantees that all errors are reported and marked on the page to tell the operator and user that the printing probably is not what it should be.

When testing your applications you may expect to have errors, so it may be reasonable to block both types of error reporting.

6.57 How to print with a PC laser printer and a P/390

Printed output, typically from JES2, can be handled by the P/390 AWS2821 device manager. Output is sent to a printer (LPT1 on the OS/2 system, by default) or to a file. Output handled by this device manager is automatically translated to ASCII. Other than translation to ASCII, it is not altered; it will contain JES2 separation pages and is probably oriented to 132 column printing. In this section, we are discussing only line-mode (1403 style) printing, and not AFP printing.

The operation described here can also be applied to an MP3000 system.

A Server file may be specified instead of a printer. Printed output is appended to this file. After writing to the file, if no device manager control parameters are specified, AWS2821 closes the file if no new print lines arrive within 10 seconds. When more print lines arrive, the file is reopened (in append mode) and the new output added. By default, AWS2821 is not aware of different OS/390 jobs. The output may contain separator pages from JES2, but these are merely print records that are placed in the output file. You (on the Server side) must separate the output for different jobs.

The output file might reside on a LAN or NFS server and be available to multiple users. Be careful using these files, because file servers usually do not offer much protection against two programs opening the same file for output. AWS2821 always writes to the file (in append mode) and if your Server session only reads the file, there is no conflict. However the file will grow forever, and will consist of the output from many jobs (if it is JES2 writing the file, as a normal output printer device). To trim the file, or divide it into job-related files (easy with a REXX program), you need to be certain that AWS2821 (JES2) does not attempt to write to the file while you are manipulating it.

AWS2821 emulates a mainframe line printer on a PC-type printer. The emulated printer is an IBM 1403 or a 2311. We recommend you emulate 1403 printers unless you need to work with FCBs.

To configure AWS2821, use the Configurator Update System Devices menu and enter the real address for the 1403 printer in the Addr field, the MGR code for AWS2821 in the Mgr field, and any optional parameters in the FN/P field:

```

Addr Device                      Mgr  FN/P
-----
> 00E > 1403                      > 6  > [parameters]

```

Optional Parameters:

- filename = Fully-qualified file name for printing to a file.
Default is to print to the PC printer LPT1.
LPTn (no colon) is also allowed.
The file is opened in append mode.
- /80 = Print in 80 column mode. Default is 132 columns.
Records are truncated to 80 columns. 80 column mode also sets a larger font, 132 column mode sets a smaller one.
- /I = Send a printer initialization string to an IBM 40x9 Laser Printer to switch it to PPDS mode. Default is not to send any initialization string.
- /C=filename specifies a fully-qualified file name, and this file contains advanced control parameters for AWS2821

Examples of FN/P field values:

```

LPT2
LPT1 /80
/C=D:\P390\PRCTL.00E
G:\printer\output.00E
D:\mydata\script.prt /80
LPT2 /I

```

The DMSTART command (in IPL.CMD) that starts this device manager may have one parameter. It is a time (in seconds) that applies when a printer output file is open. If no printing occurs during the specified number of seconds, the output file is closed. For example:

```
'AWSSTART AWS2821.EXE n 60' /* wait 1 minute before closing files */
```

AWS2821 can accept a control file. You must create this file yourself; the P390 configuration program cannot do it automatically. To use it, include the name of the control file in the configurator line defining the emulated printer. For example,

```

Addr Device                      Mgr  FN/P
-----
> 00E > 1403                      > 6  > /C=D:\P390\PRCTL.00E

```

In our examples, we assume that Manager 6 is AWS2821; you should use the number that corresponds to AWS2821 on your system. The /C= characters are required. The remainder must be a fully qualified file name; we suggest using a meaningful name, such as shown in the example.

The control file is a simple ASCII file, created with any of the basic OS/2 editors. The control file will contain several lines in any order; the basic control lines are as follows:

```

// Comment lines begin with two slashes
PC= xx xx xx xx xx xx xx xx xx
EOJ=nn,oo,'ssssssssssssssss'
LL=132      (or some other length)
FILE=C:\OUTLST.TXT
TR=xx xx xx, yy yy yy
LINE NUMBER

```

You are unlikely to have all of these control lines in any given instance of a printer control file. The meanings of the parameters are:

- ▶ PC provides a string of printer control characters (entered in hex) that are sent to the printer every time a printer output file is opened. You must provide this string; there are no defaults. AWS2821 does not verify these codes. The maximum length of the line is 255 characters. All control files will normally have this parameter line. You may have multiple PC lines in your control file. They will, in effect, be concatenated into one string before being sent to the OS/2 printer.
- ▶ EOJ, if present, checks line nn (after an emulated skip to channel 1), at offset oo in the line, and attempts to match string ssss. If a match is found, then this page is the ending page of a job. After the next skip to channel 1, AWS2821 will close the printer file. (If more printed output arrives from the P/390, a new printer file is opened.) The nn and oo parameters start with 1; a page begins with line 1 and column 1. If the nn or oo parameters are zero, AWS2821 will scan the whole page or line, as appropriate.
- ▶ LL., which defaults to 132, sets a truncation length for print lines.
- ▶ FILE provides the name of a Server file for output, instead of sending output to the printer. You can specify an output file name in the configurator line instead. You can also specify a printer name, such as LPT1 or LPT2, as the file name; this would be useful if you have more than one printer.
- ▶ TR permits individual character translation, and can serve some of the more basic national language requirements. The xx operands are hexadecimal values representing S/390 bytes. The yy operands are hexadecimal values sent to the PC printer or file. For example, TR=C1 C2 C3, 61 62 63 would result in S/390 bytes for A, B, and C being printed as lower case a, b, and c on the PC. All x'C1' bytes in incoming S/390 data would be translated to x'61' before being sent to the PC printer or file, and so forth. If no TR statement is present, a generic EBCDIC to ASCII translation is used. The TR statement is limited to about 254 characters in length, permitting up to about 31 character translations. There must be an equal number of xx and yy operands, of course. The comma signals the end of the xx operands. You may have multiple TR statements, and may thus translate all 256 possible byte values.
- ▶ LINE NUMBER causes AWS2821 to overlay the first 20 characters of every line with a field showing the line number (on the logical page, as seen by AWS2821). This is useful for determining the correct parameters for the EOJ operand.

With proper setup, AWS2821 plus a control file is a powerful combination. The ability to send control characters to the printer, before printing any output, is the key to the advanced functions. If your printer supports the necessary functions, you can send controls to rotate the output (to landscape mode), select a smaller character set (to hold more lines and characters, for example), and print on both sides of the paper (duplex).

We found that OS/2 reinitializes the printer between jobs (as OS/2 regards jobs). The initialization strings sent to the printer via AWS2821 did not affect the proper printing of output from other OS/2 applications.

Our most common control file looks like this:

```
PC=1b 45 1b 26 6c 31 53 1b 26 6c 31 4f 1b 26 6c 38 44 1b 28 73 31 33 56
```

We found no need for EOJ or TR statements. The above control string causes a Lexmark Optra printer (with the duplex feature) to print duplex, in landscape mode, with 12 characters per inch and 8 lines per inch. By default, a Courier font is used. The translation for this string is:

```
1b 45          Reset the printer
1b 26 6c 31 53 Enable duplex operation
1b 26 6c 31 4f Use landscape mode
1b 26 6c 38 44 Use 8 lines per inch
1b 28 73 31 34 56 Use 14 characters/inch. (Results in 12 cpi)
```

Another example, for an IBM 4019 laser printer (without duplex), is:

```
PC=1b 5b 4b 05 00 05 31 02 00 0a 1b 26 6c 31 4f 1b 26 6c 31 32 44
```

which results in Courier 16.67 (a very small font) and landscape mode with 12 lines/inch.

The following control file is intended for use with an IBM Network Printer model 12. It is listed here as several lines to make reading easier; in use the PC= statement should be a single line. The TR= statement is to make the *cent* and *not* symbols print correctly. We leave the bit-by-bit description of these controls as an exercise for the reader:

```
pc=1b 45
1b 28 30 4e
1b 28 73 30 70 31 36 2e 36 37 68 38 2e 35 76 30 73 30 62 30 54
1b 26 6c 31 6f
31 73
36 31 66
35 2e 35 63
32 30 30 55
tr=4a 5f,a2 ac
```

In general, this will print landscape, duplex, small font.

The most recent versions of AWS2821 for the P/390 will accept UCS and FCB commands, but ignore them. This means that JES2 can issue these commands without receiving a command reject, but that they have no effect on printed output.

You may need to experiment a little to find the best combination for your printer. If you use a small font and intend to have more than 61 lines per page, you will probably need to change the default line count parameter in SYS1.PARMLIB(JESPARM).

You can define multiple printers to OS/390, JES2, and AWS2821. We suggest you use 1403 definitions in OS/390. The AD system contains several 1403 definitions, but you may want to add more (using HCD). You can define a different control file for each OS/390 printer, and this can provide a large number of combinations—even if you have only one PC printer available. Unless you tell it otherwise, AWS2821 will route all output to eventual printing on LPT1. You can have several AWS2821-emulated printers sending output to LPT1 at the same time. The OS/2 spooling function will keep the jobs separate. For this reason we recommend that you do not disable OS/2 spooling, even though it results in dual spooling—once by JES2 and once by OS/2.

Special notices

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, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

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

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

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere., The Power To Manage., Anything. Anywhere., TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

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

PC Direct is a trademark of Ziff Communications Company in the United States and/or other

countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

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

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

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

Related publications

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

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 369.

- ▶ *Multiprise 3000 Technical Introduction*, SG24-5633
- ▶ *Multiprise 3000 Basic Emulated I/O Definition*, SG24-5669
- ▶ *NUMA-Q Enabled for S/390: Technical Introduction*, SG24-6215

Other resources

These publications are also relevant as further information sources:

- ▶ *OS/390 SecureWay Security Server RACF Security Administrator's Guide*, SC28-1915
- ▶ *OS/390 MVS JCL Reference*, GC28-1757
- ▶ *OS/390 MVS System Commands*, GC28-1781
- ▶ *OS/390 MVS Initialization and Tuning Reference*, SC28-1752
- ▶ *OS/390 UNIX System Services Planning*, SC28-1890
- ▶ *OS/390 IBM Communications Server: IP Configuration Reference*, SC31-8726

Referenced Web sites

These Web sites are also relevant as further information sources:

- ▶ Fixpaks for P/390-based systems:
<ftp://p390.ibm.com>
- ▶ Partners in Development support for AD systems
<http://etpgw02.dfw.ibm.com/adcd/html>

How to get IBM Redbooks

Search for additional Redbooks or Redpieces, view, download, or order hardcopy from the Redbooks Web site:

ibm.com/redbooks

Also download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

Redpieces are Redbooks in progress; not all Redbooks become Redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.

Index

Symbols

#rvary command 268
\$A and \$H commands 267
\$D 'jobname' 267
\$DI command 266
\$DN,ALL command 267
\$HASP395 message 197
\$P 'jobname' command 267
\$TERM=vt100 274
\$TI command 267
/etc/profile 177
/etc/rc, customization 275
/etc/resolv.conf 302, 306
/etc/services 306
/u 82
/u permissions 41, 71
/usr/lib/cron/at.allow 233
/usr/lib/cron/at.deny 233
_BPX_SHAREAS=YES 177

Numerics

1403 Printer 353
1403 printer 185
2540 card reader 185
3174, SDLC 282
3270 screen sizes 318
3270 screen, clearing, in C 230
3380 emulated disks 126
3390 emulated disks 126
3490-F01 device settings 350
4C6 ABEND 168
4mm tape drive 186
9345 emulated disks 126

A

ACCOUNT command 121, 122
ACCTNUM 91, 171
ACDS (Active Control Data Set) 139
ACF user, definition of 167
ACS routine 147
AD CD-ROM installation 6
AD installation for MP3000 11
AD system creation 5
add a program to OS/390 213
ADDSD (or AD) commands 125
administrative environment 203
ADMINISTRATOR keyword 161
ADRDSU 14
ADRDSU dump, copying 211
ADRDSU job 161
ADRDSU program 210
Advanced Function Presentation (AFP) printing 353
AFP data streams 357

AFP error messages 361
AFP resources 355
alias, PDS, editing 228
ALLOC command 216
ALTUSER command 90, 115, 120
AMASPZAP 226
AMODE24 and OpenEdition 237
Application StarterPak LAN setup 336
ardware Configuration Definition (HCD) 109
ASCH and APPC startups 39, 70
at command 233
ATCCON00 mem ber in VTAMLST 280
attention key 212
AU (ADDUSER) 89
authorized library list 114
authorized program execution 232
authorized programs 214
AUTO BACKUP 143
AUTO MIGRATE 143
Automatic Class Selection (ACS) 144
automatic IPL 106
Auxiliary Storage Manager 187
AWS2821 device manager 363
AWS3172 device manager 283
AWS370DD driver 348
AWSCFCON.DOC 4
AWSCFG command 10
AWSDEV908E message 20
AWSLCS350S message 20
AWSMOUNT command 342
AWSTAPE mounts 341

B

backup and restore OS/390 209
batch job, class 194
BDAM 238
BEGINROUTES statement 301
BLP authority 195
BPX.DAEMON 61
BPX.FILEATTR.APF 61
BPX.FILEATTR.PROGCTL 61
BPX.SERVER 61
BPX.SUPERUSER 61
BPX1WRT 221
BPXAS jobs 220
BPXPRM00, DB2 errors 74
BPXPRMDB, DB2 errors 74
BPXPRMxx 36, 66, 68
BPXPRMxx member 272, 321
BPXPRMxx members, DB2 74
BPXPRMxx, MAXASSIZE 189
BSC and SDLC adapters 349

C

- C characters, displaying 345
- C compiler procedures 40
- CALL statement (TSO) 213
- Cancel batch job, started task, or TSO user 260
- Canonical form of MAC addresses 352
- catalog protection 86
- Catalogs 35, 65
- CD-ROM drive and OS/2 346
- CD-ROMs 26, 54
- CEEEVDBG 239
- Changing your I/O configuration 18
- channel-attached drives 82
- CKDCHECK program 321
- class table entries 160
- clean address space 169
- clipping the volume 135
- CLIST allocations, ISPF 203
- CLIST, logon 177
- CLPA 14, 20
- COBOL library 226
- COFVLFXx 133, 225
- cold start 20, 184
- COMMNDxx 106, 113, 194
- COMPACT feature 33, 64
- COMPACT feature in HCD 273
- compress a PDS 190
- CONFIG.SYS 20, 331, 348
- Configurator 10
- CONNECT commands 91
- Consolidated Software Inventory (CSI) 240
- CONSOLxx 113
- continue a TSO command line 211
- copy and paste 219
- core dump size, OE 272
- CPU ID of a P/390 340
- CR, LF, NL 238
- CREATE authority 87
- cron command 233
- CSA sizes 274

D

- D A,A command 259
- D A,L command 258
- D GRS command 264
- D GRS,RES command 265
- D IOS,CONFIG 104
- D IOS,CONFIG command 264
- D IPLINFO command 264
- D LLA command 225
- D M=DEV(nnn) command 262
- D PARMLIB 264
- D R,L command 260
- D U,,nnn,m command 261
- D U,,ALLOC,nnn command 263
- D U,DASD,ONLINE command 262
- Daemons 162
- DASDVOL 161
- DATA data set 302

- Data Set In Use message 246
- data set names 150
- data set names, system 207
- data set prefixes, system 208
- data set user, identification 246
- DB2 37, 68
- DB2, mounts, HFS 74
- DBCLASS 138
- DBSPROC 62
- DBSPROC logon procedure 48, 79
- DBSYNC program 163
- dbx 239
- DEFAULTNET statement 300
- DEFINE ALIAS statements 134
- DELDSD (or DD) commands 125
- delete users 116
- device address 102
- device numbers 102
- DEVMAP 3, 9, 18
- devserv 263
- DFSMS configuration name 176
- DFSMS state 38, 69
- DFSMS storage group 136
- DFSMSshm 210
- digital certificate 164
- directory, PDS 190
- dirty address space 169
- disk integrity, P/390 321
- Disk space required 27, 55
- disk types 126
- disk volumes, new 82
- display area, operator screen 260
- domain name server 306
- dot command 234
- DS P,nnn command 263
- dub 168
- dump command 268
- dump data sets 192, 193
- dump data sets, security 194
- DUMPSRV 248
- Dynamic Reconfiguration 109
- Dynamic routing 310

E

- EDIT, TSO 197
- emergency userids 121
- emulated disk volumes 127
- emulated I/O 6, 82
- Enhanced Generic Naming (EGN) 42, 72
- ENTR OSA-2 Adapter 336
- error messages, IPL 18
- escape key function 212
- ESCON channel attachment 82
- EtherJet adapter, installing 351
- Ethernet adapter, how many 343
- Ethernet adapters for MP3000 352
- ETR timer 39, 70
- EXEC (execute) statement 215
- EXEC statement condition codes 224
- executable, moving from HFS 236

execute vs read permission 122
expanded storage, S/390 348
extattr command 230
extended attributes, setting 230
EZAZSSI job 310

F

Facilitator product 346
file system, adding 92
File Transfer program 219
files, invalid UID 167
FLEX-ES 2
FLEX-ES EFS 103
FLEX-ES emulated disks 82
FMID 240
FMID numbers 24, 52
Fonts 358
Forms Control Buffers 353
FREE command 216
FTP customization 303
FTP data set parameters 315
FTP failure with browsers 318
FTP from a batch job 315
FTP partitioned data sets 317
Fundamental Software, Incorporated 2

G

GATEWAY statement 299
general resource information 154
Global Access Checking (GAC) 42, 73
Global Access Control, RACF 165
Group definitions 62
group profile information 154
groups, user 86
GRS system name 176
GRS, display 246
GUARANTEED SPACE option 141

H

hardware address 102
HASPPARM=ALTPARM 174
HCD change 201
HFS allocation 217
HFS data sets, RACF protection 43, 73, 74
HFS file system, creating 92
HFS mounts in BPXPRMxx 68
HFS, maximum file system size 239
HFS, reclaiming space 235
HFS.WAS 68
HFSCCLASS 136, 138
High Level Qualifier 84
HLQ, definition 84
host name customization 306
Hot I/O 186
HPFS386 file system 347
HTTPD1 41, 71, 237

I

I/O devices, query or control 261
ICKDSF 14, 223
ID(*) vs UACC 167
IDCAMS commands 85, 199
IDCAMS job 136, 199
IDRC 273
IEAICSxx and IEAIPSxx 113
IEASYMxx 176
IEASYS00, listing 110
IEASYSxx 106, 114
IEBCOPY 191
IECIOSxx member 186
IEFSSNxx 114
IEFSSNxx member 309
IFASMFDP program 175
IGDDFPKG 114
IKJACCNT 62
IKJTSOxx 114
IMWEBSRV 41, 71
IND\$FILE 219
INDEX parameter, VTOC 139
inetd customization 305
inetd procedure 46, 74
inetd, restart 235
initiator, names, classes 195
initiators 185
initiators, number of 40, 71
inst3172 command 284
install command 8
Integrated Server 2
IOCDS 3, 12, 18, 109
IODF01 63
IODF99 63
IODFs 18, 33, 63, 109
IP addresses 293
IPCS inventory 41, 72
IPL 14
IPL information, display 246, 264
IPL parameters 28, 29, 56, 57
IPL the system 11
IPL, manual 105
IPL, running shell script 275
IRRDBU00 job 158
IRRDP100 61
IRRMIN00 utility 164
IRRUT200 program 157, 158
IRRUT400 utility 163
ISFPRMxx member 181
ISMF 137
ISP.SISPPENU 179
ISPF edit, line length 274
ISPF menu 179
ISPF primary menu 48, 78
ISPFLITE 62
ISPFLITE procedure 45, 75
ISPFPRCB 62
ISPFPROC 62
ISR@PRIM 179
ISRDDN, using 226

J

- JCL library 214
- JES2 checkpoint data set 174
- JES2 commands 244
- JES2 definitions 184
- JES2 member name 176
- JES2 security messages 164
- JES2 SPOOL space 174
- JES2 spool space 36, 66
- JES2 starts 107
- JES2, hot start 246
- JES2BACK 174, 185, 196
- JES2PARM 174, 184, 185, 194
- JES2PARM definitions 223
- job classes 194
- JOB ENDED messages 197
- JOB statement 214
- Job Wait Time 176
- JOBLIB statement 213
- JWT 176

K

- K commands 260

L

- LAN adapter plugging, MP3000 352
- LAN3172 350
- LAN3274 connections, verification 349
- laser printer 363
- LCS3172 device manager 44
- LCS3174.SYS 20
- LD command 168
- line count 361
- linkage editor vs binder 274
- Linux for S/390 4
- List of Groups 42, 72
- LIST3820 355
- LISTUSER 153
- LISTUSER command 120
- LLA (Library LookAside) 225
- LLA, REFRESH 214, 225
- LNKLST 213
- LNKLSTxx 114
- load libraries, blocksize 220
- load module, create from OE 233
- loaded program, size 235
- LOADxx member 84
- logical volumes 13, 82
- LOGOFF command 217
- logon CLIST 177
- logon ID, finding 222
- LOGON procedures 218
- lp command 46, 76, 231
- LPA 213
- LPALSTxx 114

M

- MAC addresses, canonical 352

- mainframe printing 353
- manual IPL 106
- master catalog 7, 84, 134, 160
- master catalog, protection 86
- Master JCL 114
- MAXASSIZE 189, 190
- MAXCORESIZE() parameter 272
- migration process 7, 8
- MIH 185
- Missing Interrupt Handler 185
- Missing interrupt handler 294
- mount attribute 83
- MP3000 printing 363
- MP3000, Ethernet adapters 352
- MP3000, LAN adapter plugging 352
- MPFLST00 197
- MPFLSTxx 114
- MPTS configuration 325
- MPTS interface number 299
- MPTS, locked 352
- MSTJCLxx 114
- Multiport -2 349
- Multiprise 3000 2
- MVS 1
- MVS console 247
- MVS data set, create in OE 273
- MVS.GLOBAL.CSI 272
- MVSADMIN.WLM.POLICY 61

N

- naming conventions, data set 92
- Netfinity/EFS, installation 15
- netstat 313
- network file system 311
- NFS operation 44
- NJE setup 288
- NLS printing 359
- NOEXPIRED option 120
- NOHUP to start a daemon 234
- NOINTERVAL option 120

O

- OE, preventing use 235
- oedit and permission bits 273
- Oedit and su 272
- OMVS segment 116
- OMVS segments 170
- OMVS, RACF protection 43, 73
- OMVS, SMF output 235
- OMVS, terminology 229
- OMVSPROC 62
- onetstat 313
- Open Edition, create MVS data set 273
- OpenEdition userid, default 166
- OpenEdition, terminology 229
- OPER authority 32, 61, 258
- Operating the system 258
- oping 313
- OPUT command 238

OS/2 CONFIG.SYS 20
OS/2 files 6
OS39DA 26
OS39M1 7, 20, 28, 56, 82
OS3RAA 82
OSA-2 adapter 336
otracer 313
Outlist 199

P

P/390 adapter card 2
P/390 device managers 44, 74
P/390 hardware generations 2
P/390 icon window 102
P/390 program levels 348
P/390 subsystem 102
P/390-based systems, z/OS 4
page data sets 189
Paging and spooling data sets 35, 66
paging data sets 82
paging space 187
PANELID command 180
PARMLIB members, commonly modified 113
PARMLIB, working with 110
Parmlibs 27, 55
Partitioned Data Sets 190
PASSWORD command 116, 119
password history 164
password interval 153
password, reset 119
paste and copy 219
pax command 275
PBX.DAEMON profile 162
PCOM configuration 334
PCOM's printer startup 348
PCOM, character sets 345
PDS compression 190
PDS editing, OS/2 346
PDSE (execution library) 275
permission bits and oedit 273
PERMIT (or PE) commands 125
PERMIT command 90, 91
PFK assignments 48, 78
ping 313
PLPA 28, 36, 56, 66
PLPA Page Data Set Full 115
PLPA, adding to 114
PREFIX parameter 199
prefix, TSO 212
prelink C and C++ 274
Prelinking C programs 272
primary disks 13
print a file in OS/390 360
print classes 194
PRINT01.SYS statement 351
printer performance, emulated 351
printer resolutions 359
PROFILE command 199
program control 168
program object 275

PROGxx 114
PROTECTALL function 172
PROTOCOL.INI 350
PSF/MVS 359
PTF status 241
PTFs (fixes) 239
PTFs in system 272
PUBLIC or STORAGE volumes 115
PUBLIC volume 83
PUT tapes 239

Q

qualifier, data set name 151

R

R/390 2
RACF Administration 152
RACF backup 157
RACF commands, who can issue 125
RACF data base 7
RACF database merging 163
RACF database name 166
RACF database, local 158
RACF execute permission 122
RACF FACILITY class 32, 61
RACF information display 153
RACF ISPF panels 153, 154
RACF parameters 42, 72
RACF passwords, rules 164
RACF profile, protected data sets 168
RACF read-only protection 275
RACF SEARCH command 154
RACF TSO segment 189
RACLIST 42, 72
read permission vs execute permission 122
read-only RACF protection 275
Redbooks Web site 369
 Contact us xii
region size 188, 190
reset a password 119
Resource Definitions 3
revoked user, dataset allocation 162
REVOKED userids 32, 61
REVOKED, in RACF 119
REXX clist 221
REXX script execution 218
rlogin 234
RMF automatic start 174
RVARY LIST 166

S

S, P, and F commands 265
S/390 expanded storage 348
S/390 Integrated Server 2
SAF 152
SAF interface 152
SCDS (Source Control Data Set) 141
SCEERUN placement 226

SCHEDxx 114
 SCHEDxx member 309
 scratch disks 134
 scratch volume 83
 scratch volume(s) 82
 SCSI attached tape drives 210
 SCSI tape drive 3
 SCSI3480.SYS 20
 SDLC 3174 282
 SDLC adapters 349
 SDSF LOG 183
 SDSF security 181, 184
 SE command 266
 security policy 152
 security, JES2 164
 ServerPac 5
 SET IOS=xx command 186
 SET SMF=00 173
 SETROPTS command 87
 SETSMS SAVEACDS 139
 SG24-4757 1
 shell script at every IPL 275
 SHUTDOWN scripts 108
 SIMETRID line in CLOCKxx 70
 size, loaded program 235
 SMF data 172
 SMF data sets 175
 SMF identifier 176
 SMF output, OMVS 235
 SMF recording 36, 66
 SMFPRM00 173
 SMFPRM00 member 172
 SMFPRMxx 115, 176
 SMP/E 239
 SMP/E data sets 37, 68
 SMP/E primary menu 272
 SMP/E region size 43, 73
 SMS disks 138
 SMS volumes, allocating to 150
 SNA monitor 287
 SPECIFY SYSTEM PARAMETERS 106, 112
 Specify System Parameters 104
 SPF profile data set, allocation 179
 spool data set 82
 spool space 174, 243
 SPOOLDEF parameters 174
 spooling data sets 66
 SQA and CSA sizes 274
 square brackets, displaying 345
 staging group 12
 STARTED class 42, 73
 STARTED class profile for OMVS 73
 STARTED class profiles 249
 STARTED task control 249
 started tasks 86
 stdout 221
 STGADMIN class 161
 STGADMIN controls 161
 STGADMIN.* 61
 sticky bit 237

STIDP instruction 340
 storage group 136
 STORAGE volume 20, 83
 STORAGEEGROUP parameter 139
 su and oedit 272
 su to root 163
 SUBMIT authority 171
 SUBMIT command 216
 submit job, surrogate 160
 SuperZAP 226
 Superzap 161
 SURROGAT class 161
 surrogate 161
 SVC dump 190, 268
 SVC dumps 247
 Sx37 ABENDS 222
 SYMBOLS, displaying 221
 SYS1.ACDS 149
 SYS1.ADCD10.PARMLIB 27, 55
 SYS1.CLIST 178, 218
 SYS1.DUMPnn 193
 SYS1.DUMPxx 248
 SYS1.IMAGELIB 353
 SYS1.IODF01 33, 63, 201
 SYS1.IODF99 33, 63
 SYS1.LOCAL.ISPFPNLS 180
 SYS1.LOCAL.VTAMLST 192
 SYS1.LOGREC 39, 175
 SYS1.MAN1 172
 SYS1.MIGLIB 226
 SYS1.MIGRATE.R10 8
 SYS1.P390.CNTL 49, 79
 SYS1.P390.LINKLIB 214
 SYS1.PARMLIB 27, 55
 SYS1.README 5
 SYS1.SAMPLIB 239
 SYS1.SCDS 149
 SYS1.UADS 121, 122
 SYS1.VTAMLST 192
 SYSDA devices 63
 SYSLEVEL command 348
 SYSLMOD DD statement 214, 275
 SYSMOD 240
 SYSNAME parameter 176
 SYSOUT 199
 SYSOUT classes 196
 system console 4, 247
 system definition file, EFS 103
 system IDs 176
 System Managed Storage 138
 system variables 221
 system-managed volume 136

T

tape device types for HCD 273
 tar file 275
 TCAS UNABLE TO ACCEPT LOGONS 107
 TCP/IP applications, compiling 314
 TCP/IP host name 176
 TCP/IP problems 311

- TCP/IP profile 296
- TCP/IP setup and customization 292
- TCP/IP setup for P/390 323
- TCP/IP, test version 320
- TCPIP procedure 295
- TCPIP.ETC.SERVICES 306
- TCPIP.HOSTS.LOCAL 306
- telnet sessions to USS 274
- TMOUT environmental variable 177
- TN3270 idle time parameter 177
- TN3270 ports 316
- TN3270E connections 316
- token ring adapters, how many 342
- token ring, disconnecting 346
- tracerte 313
- transfer data between OS/2 and OS/390 219
- TRUSTED attribute 73
- TSO account numbers and procedures 91
- TSO authorities 171
- TSO command from OE 234
- TSO command lines 211
- TSO definitions, VTAMLST 278
- TSO down 223
- TSO Edit 197
- TSO logon ID, finding 222
- TSO logon procedures 32, 62
- TSO prefix 212
- TSO region size 188, 189
- TSO sessions, number 192
- TSO timeout 176
- TSOAUTH 172, 183
- TSOAUTH profile 258
- TSOKEYxx 115, 192
- TSOPROC 91
- TSOPROC class 171

U

- UACC vs ID(*) 167
- UACC(READ), exposure 275
- UCBs 109
- UID, invalid 167
- UID=0 32, 61
- Unit Addresses 13
- UNIT=SYSDA devices 63
- Universal Character Set 353
- UNIX characters, displaying 345
- UNIX System Services file structure 36, 66
- Unix System Services terminology 229
- unload RACF 157
- unzip commands 9
- use attribute 134
- user catalog, adding 84
- user catalog, migration 134
- user catalogs 8, 84
- user catalogs, defining 132
- user groups 86
- user profile, modify 115
- userids 18, 30, 59
- userids, adding 88
- users, deleting 116

- USSTAB module 281

V

- VARY nnn, ONLINE and VARY nnn, OFFLINE commands 263
- VATLSTxx 115
- VATLSTxx member 83
- vi command, under USS 274
- VIO, using 213
- Virtual Memory, controlling 188
- volume serial number 135
- VSAM catalog 152
- VSAM data set components, removing 136
- VSAM data sets 82
- VTAM application definition 280
- VTAM application names 176
- VTAM logo 281
- VTAM name 176
- VTAM subarea number 176
- VTAM subsystem, startup 276
- VTAM, start and stop 279
- VTAMALL 58
- VTAMAPPL 58, 108
- VTAMAPPL program 113
- VTAMBB 58
- VTAMCI 58
- VTAMD2 58
- VTAMDB 58
- VTAMLST 44, 74, 192, 276
- VTOC index 135
- VTOC size, expanding 222
- VVDS 136

W

- WAC adapter 349
- WAN3172 device manager 283
- WAN3172 installation 284
- Web server 41, 71
- Web Server, bounce 237
- Web server, logon 43, 73
- Web site, AD support 4
- Webserver, CSA and paging 274
- WebSphere 73
- WebSphere, CSA and paging 274
- WorkLoad Manager 221

X

- XTAPE program 209

Z

- z/Architecture, testing for 275
- z/OS system, P/390 use 4
- Z1RES2 82
- zone, SMP/E 240



Redbooks

S/390 Partners in Development: OS/390 (and z/OS) New Users Cookbook

(0.5" spine)
0.475" x 0.873"
250 x 459 pages



S/390 Partners in Development: OS/390 (and z/OS) New Users Cookbook



Hints & tips for new users

Based on OS/390 R10 and Z/OS R1.1 AD CD-ROMs

Uses P/390, MP3000, and Netfinity Enabled for S/390

The IBM S/390 Partners in Development (PID) organization provides several prepackaged S/390 operating systems for its members. This IBM Redbook documents one of these, the OS/390 (or z/OS) Application Development CD-ROM system, commonly known as the AD CD-ROM system, or simply the AD system. (Equivalent systems have been produced for VM/ESA and VSE/ESA, but this IBM Redbook concerns only the OS/390 and z/OS version.) The AD system is a straightforward implementation of OS/390 or z/OS, plus a selection of other products. The CD-ROM portion of the name reflects the fact that these systems are distributed on CD-ROMs.

This document discusses the most recent AD releases, including basic planning for system administration. The bulk of the document consists of “how to” articles that may be useful to new users and owners. It does not attempt to provide comprehensive coverage of any topic, but it does address many common queries and situations encountered by new users. Users completely new to S/390 and OS/390 (or z900 and z/OS) will require initial education before using this document. The PID organization offers periodic classes for users completely new to S/390 and OS/390.

Most of the tips and techniques described in this document are not limited to the AD system, and should apply to any smaller OS/390 implementation. However, the specific examples included are based on the AD system.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-6204-00

ISBN 0738423505