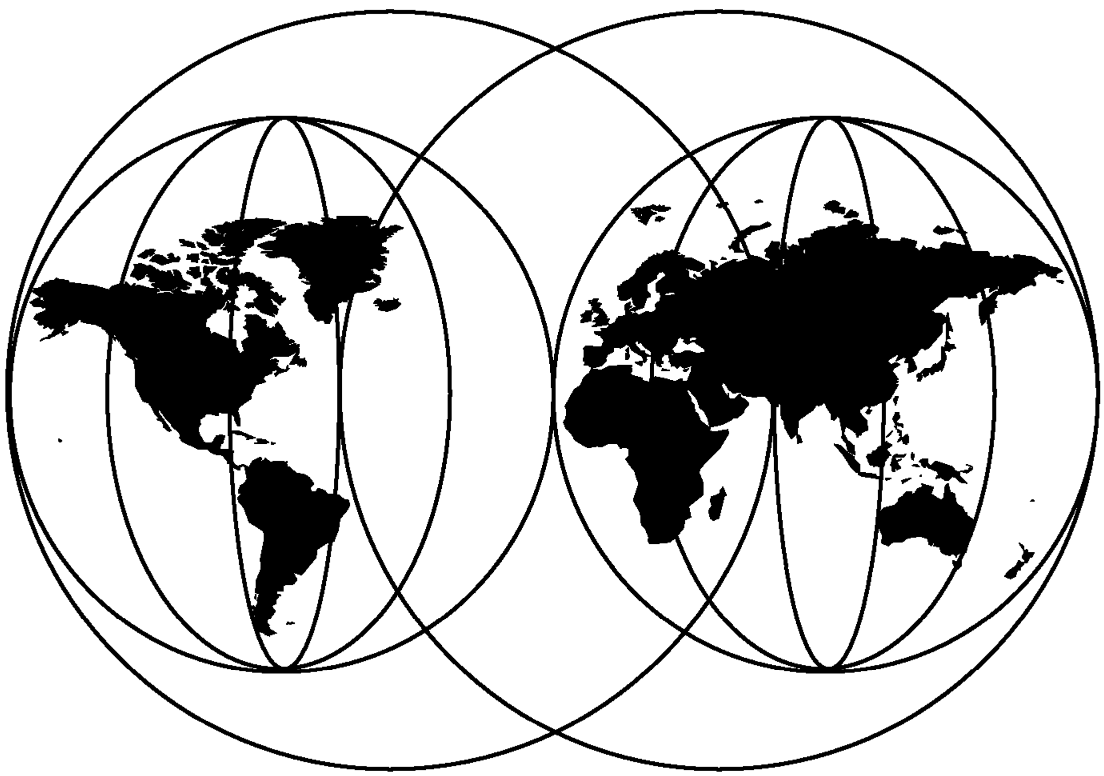
P/390, R/390, S/390 Integrated Server: OS/390 New User′s Cookbook

Bill Ogden



International Technical Support Organization

http://www.redbooks.ibm.com

This book was printed at 240 dpi (dots per inch). The final production redbook with the RED cover will be printed at 1200 dpi and will provide superior graphics resolution. Please see “How to Get ITSO Redbooks” at the back of this book for ordering instructions.

SG24-4757-01

International Technical Support Organization

SG24-4757-01

P/390, R/390, S/390 Integrated Server: OS/390 New User′s Cookbook

April 1999

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix A, “Special Notices” on page 321.

Second Edition (April 1999)

This edition applies to V2R6 of OS/390.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

Contents

Preface

IBM systems based on P/390 adapters provide entry-level platforms that are ideal for program developers. Many developers, new to MVS and OS/390, are using these to port or implement their program products onto OS/390. This has produced many queries about basic OS/390 usage. Basic operations, understood without thinking by long-time MVS users, can be confusing to new users.

The formal OS/390 documentation is not in the format to conveniently answer many of these simple usage queries. This Redbook is a collection of "How to..." articles that may be useful to new OS/390 users and owners. It does not attempt to be comprehensive -- no single document could do that -- but it does address many of the common situations encountered by new users. The OS/390 Application Development CD-ROM system is widely used by IBM′s S/390 Partners-In-Development, Business Partners, and others. The AD CD-ROM is a complete, ready-to-use OS/390 system, and contains a substantial number of products in addition to base OS/390.

Tips and techniques described in this document are not limited to the AD CD-ROM, but most examples and illustrations use the AD CD-ROM as the base system. It can be regarded as a concrete, specific expression of a standard OS/390 installation in a smaller environment.

The Author

Bill Ogden is a contractor at the International Technical Support Organization, Poughkeepsie Center. He was a Systems Engineer with IBM for 30 years, working primarily with MVS systems in a number of countries. He is now a principal with ITSS International, Inc., and works with several independent consultancy organizations.

Earlier Version

The following authors contributed to an earlier version of this document:

Martin Ceron is a manager with IBM Mexico′s Technical Support Services. He specializes outsourcing environments, with special interests in operations management, disaster recovery, and consolidation and migration planning.

Mark Worboys is a Senior Consultant with Independent Systems Integrators (ISI) of Sydney, Australia. ISI is an IBM Business Partner. Mark is a well-known MVS consultant in Australia with 23 years experience. He provides P/390 and Multiprise 2000 technical and marketing support for ISI.

Mikko Markkula is a Systems Engineer with IBM Finland. Mikko is one of the best-known printing specialists in Europe. He has helped produce many Redbooks in recent years.

Additional Contributions

We thank the following people for their contributions to this project:

- Chuck Berghorn - IBM P/390 Development (Poughkeepsie)
- Karen Miller - IBM Customized Solutions (Poughkeepsie)
- Rich Conway - International Technical Support Organization
- Lou Voerman - IBM P/390 Development (Poughkeepsie)
- Bob Haimowitz - International Technical Support Organization
- Kurt Meiser - ITSS International, Inc.
- Marty Ziskind - IBM P/390 OEM Support Manager (Poughkeepsie)
- Jay Hall - P/390 Development (Raleigh)
- Bill White - International Technical Support Organization
- Kathryn Arrell - International Technical Support Organization
- Reza Kaffashan - IBM Partners-In-Development (Poughkeepsie)
- Carmine Castaldo - R/390 Development (Poughkeepsie)
- Jim Nichols - IBM Partners-In-Development (Poughkeepsie)
- Doris Conti - IBM S/390 Division (Poughkeepsie)
- Peter Tannebaum - R/390 Development (Poughkeepsie)

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 331 to the fax number shown on the form.
- Use the online evaluation form found at `http://www.redbooks.ibm.com/`
- Send your comments in an Internet note to: `redbook@us.ibm.com`

Chapter 1. Introduction

This Redbook is intended for new owners of P/390, R/390, and IBM Integrated Servers, who are also relatively new users of OS/390.[1] "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.

This format of this document consists of many short "How to" articles, in no particular order. It is intended as a reference document, and is not intended to be read from beginning to end.

Many of the examples in this document refer to a specific packaging of OS/390. This is the Application Developers' system distributed on CD-ROMs, and is commonly referenced as the AD CD-ROM system. 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.

AD CD-ROM

The term *AD CD-ROM* refers to a particular packaging of OS/390 on several CD-ROMs. The full title is *Application Development System for OS/390*, and it is commonly known as the "AD" system. The package contains a straightforward implementation and customization of OS/390, a number of development products -- compilers, for example -- as well as CICS, IMS, DB2, and so forth. Many of the specific examples in this document refer to the AD CD-ROM.

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

This document uses the term MVS more frequently than OS/390, and the two are used interchangeably.

MVS

MVS is used instead of OS/390 throughout this document. Please be aware that the proper term in most cases is OS/390. In particular, you should not infer references to pre-OS/390 time frames when you encounter MVS references in this document.

[1] It also applies to special packaging of P/390 systems, such as the laptop model produced by Fieldworks, Inc. (1-888-FIELDWORKS or 1-612-974-7000 or www.field-works.com.)

Chapter 2. OS/390 AD CD-ROM Status

{ZB00} This chapter provides an overview of key elements of the OS/390 V2R6 Application Development (AD) CD-ROM system. This information is intended for planning and for orientation when moving from a previous release to this new release. Some of the material in this chapter was taken from the *README.MVS* file that is contained on the first CD of the V2R6 AD CD-ROM set. Members of the S/390 Developers' Association can order the AD CD-ROM system via IBM order number LCD4-0382-5 (for the V2R6 version).

2.1 Basic Installation Information

{ZB05} This system is based upon a ServerPac. All MVS products are installed at least through the end of the ServerPac generated jobs, and service from the service tape has been RECEIVEd (existing customers can RECEIVE this service using migration job MIGSVC).

All CICS and DBS products are installed through the end of the CBIPO Installation Guide steps. Note that most CICS and DB2 products have had some customization done to them and are ready for initial use. Please review SYS1.README(DBS) for details.

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.

If your P/390 or Integrated Server uses a RAID adapter to create a single virtual drive that is large enough for all the AD CD-ROM volumes, you can use a very simple installation process. In this case, to install the AD CD-ROM insert the OS/390 Release 6 CD-ROM Disk 1 in your CD drive, open an OS/2 window, MAXIMIZE the window, and enter:

```
x:\install
```

(Where x is the drive letter assigned to the CD-ROM drive.)

You will be prompted for the destination drive (d:) where you want OS/390 installed. To accept the default (d:) just press ENTER.[2] You will be prompted to load OS/390 CD-ROM Disk 2 after the first CD has been copied, and so forth. Installation on a R/390 is similar, allowing for the AIX differences.

If your D drive is not large enough for all the volumes, see 2.4, "Manually Loading MVS Volumes From CD-ROM" on page 9. The current AD release requires about 11 GB if all volumes are installed.

If you are installing this CD-ROM over an existing MVS Preconfigured or Application Development system, you have two options:

1. Completely replace the existing MVS system.

[2] You are not required to use the D drive, but this simple installation process assumes that all the OS/390 volumes will be restored to the *single* OS/2 drive you specify. Installation for the S/390 Integrated Server will go to the E drive, if using the disk partitions that are predefined for the Integrated Server.

- You should back up (on tape, probably) all your existing volumes, if appropriate. Remember that the backup process can be under OS/390 or under OS/2.
- You must manually erase all the OS/2 files corresponding to the OS/390 volumes that will be replaced.
- Restart the install program.

2. Replace only part of an existing OS/390 R6 Application Development system. This uses the *migration* process provided with the AD CD-ROM systems. *Be certain to back up your current system before starting.* Migration is designed to step from one release to the next. For example: If you are currently running OS390 Release 4 you must go through the migration to release 5 before you can migrate to release 6.

   - You should back up all your existing OS/390 volumes, so you can return to your existing state if necessary.
   - You must manually erase any OS/2 files that need to be erased to make room for new volumes. You may choose to optionally keep or erase SCPMV5.122 and/or P390DX.A83. If you keep your existing SCPMV5, you must follow the migration instructions located in the file MIGRATE.DOC on the CD-ROM.
   - Restart the install program.

   The install program will NOT overlay any existing volumes; you must manually erase or move any files with the same names as the new volumes. Remember that these are VERY LARGE files (1.2 - 3 GB) so you might not have enough room to keep backup copies on your PC Server hard disks.

2.2 Products Included

{ZB10} The OS/390 Version 2 Release 6 Application Developer's CD (North America version) components are:

- **OS/390 BASE RELEASE 6 COMPONENTS**
  - **System Services**: DFSMSdfp, EREP, ESCON Director Support, IBM High Level Assembler for MVS, ICKDSF, ISPF, JES2, MICR/OCR Support, MVS/Bulk Data Transfer (BDT Base), MVS/ESA BCP, TSO/E, 3270 PC File Transfer Program, FFST, TIOC
  - **Systems Management Services**: HCD, ICSF, SMP/E, SystemView for MVS Base
  - **Application Enablement Services**: DCE Application Support, Encina Toolkit Executive, GDDM/MVS (includes PCLK and OS/2 Link), Language Environment, OS/390 Application Enabling Technology, SOMobjects Runtime Library, VisualLift Runtime Library, C/C++ IBM Open Class Library
  - **UNIX System Services (X/Open UNIX 95 functions)**: OS/390 UNIX System Services Application Services (Shell, Utilities, and Debugger), OS/390 UNIX System Services (included in the BCP)
  - **Distributed Computing Services**: DCE Base Services (OSF DCE level 1.2), Distributed File Service (OSF DCE 1.2.2 level), Network File System
  - **eNetwork Communications Server**: IBM TCP/IP (with CICS Sockets and IMS Sockets), SNA/APPN Services (includes VTAM), Multiprotocol/HPR Services (includes Anynet)
  - **LAN Services**: LANRES, LAN Server, OSA Support Facility

- **Network Computing Services**: IBM BookManager BookServer for World Wide Web, WebSphere Application Server for OS/390 (includes NetQuestion)
- **Softcopy Services**: BookManager READ/MVS, Softcopy Print (includes Softcopy Print for DBCS Languages)

- **OS/390 OPTIONAL FEATURES**
  - **System Services**: JES3, MVS/BDT File-to-File, MVS/BDT JES3 SNA NJE
  - **Security Server**: Security Server (RACF, DCE Security Server at OSF DCE level 1.1, LDAP Server, and Firewall Technologies), LDAP Server - DES
  - **Systems Management Services**: DFSMS/MVS features (DFSMSdss, DFSMSrmm, DFSMShsm), RMF, SDSF
  - **Application Enablement Services**: DFSORT, GDDM-PGF, GDDM-REXX/MVS, C/C++ with Debug Tool IBM High Level Assembler Toolkit, SOMobjects Application Development Environment, VisualLift Application Development Environment for MVS/VSE/VM
  - **Distributed Computing Services**: DCE User Data Privacy (DES and CDMF) - OSF DCE 1.1 level, OS/390 Print Server (includes OS/390 Print Interface, Windows 95/NT client, IP PrintWay, NetSpool)
  - **eNetwork Communications Server**: IBM TCP/IP Kerberos (DES or CDMF), IBM TCP/IP Network Print Facility, IP Security DES/CDMF (includes SSL DES)
  - **Network Computing Services**: WebSphere Application Server for OS/390 (North America Secure Feature or World Trade Security Feature)
  - **Softcopy Services**: BookManager BUILD/MVS

- **Additional Program Products**
  - TME 10 Netview Ent LE/370 (1.02.00 5697-B82)
  - ADSM for MVS Server (3.01.00 5655-A30)
  - Java for OS/390 (1.01.00 5655-A46)
  - DITTO/ESA for MVS (1.02.00 5655-103)
  - IBM MQSERIES FOR MVS/ESA (1.02.00 5695-137)
  - IBM COBOL for OS/390 & VM V2 Full Function (2.01.00 5648-A25 **)
  - CODE/370 (1.02.00 5688-194)
  - Print Services Facility/MVS Version 2 (2.02.00 5695-040)
  - CCCA for OS/390 & MVS & VM (2.01.00 5648-B05)
  - OS PL/I Comp,Lib & Int Test Facility V2 (2.03.00 5668-909)
  - REXX/370 Compiler (1.03.00 5695-013)
  - REXX/370 Library (1.03.00 5695-014)
  - COBOL for MVS & VM Full Function (1.02.00 5688-197)
  - AD/Cycle PL/I Full Function (1.01.01 5688-235)
  - C/370 Unique Library (2.02.00 5688-188)
  - C/370 Common Library (2.01.00 5688-188)
  - CICS TS and DS V5 (5655-018)
  - IMS/ESA Transaction and DB Server V6 (5655-158)
  - Database 2 (DB2) V5 5655-DB2
  - DB2 Performance Monitor V5 5655-DB2
  - DB2 Net.data V5 5655-DB2
  - DB2 JDBC V5 5655-DB2
  - QMF V3R3 5706-254

** NOTE: COBOL for OS/390 was not included in the original system order because it is incompatible with CODE/370. However we have received that code on the system in a separate SMPE Global zone on the service volume OS39SV.

2.3 Volume and Space Planning

{ZB15} Volumes provided for the V2R6 system are:

N.A. VOLSER	W.T. VOLSER	Device	Adr	IPL	Cyls	GB	Free Tracks
OS39R6	WT39R6	3390-3	A80	Yes	3339	2.8 GB	13,800
OS3R6A	WT3R6A	3390	A82	Yes	2200	1.9 GB	2,000
OS39H6	WT39H6	3380	124	**	1500	1.0 GB	2,500
SCPMV5	SCPMV5	3380	122	Yes	1770	1.2 GB	7,500
OS39D6	WT39D6	3390-3	A81	No	3339	2.8 GB	
OS39SV	WT39SV	3390	A84	No	800	.7 GB	
HFSUS1	HFSUS1	3380	125	**	200	.15GB	700
P390DX	P390DX	3390-3	A83	No	3339	2.8 GB	3,600

** In theory, not required to IPL. In practice, probably required.

There are two versions of the AD CD-ROM: the North American version and the *World Trade* version. The difference is in the encryption products that are included. The North American version handles the higher levels of encryption (56-bit DES, TDES, longer public keys, etc), while the World Trade version uses CDMF (a compatible DES with 40 bits of security). For all practical purposes, the usage characteristics and interfaces of the two versions are the same. Different volsers are needed so that both versions can be built on the same driving system, at the same time.[3]

Using the North American volsers, these volumes contain:

- **OS39R6** is the IPL volume and contains the majority of the static OS/390 data sets. It uses a full 3390-3 volume. There is considerable free space on this volume. If you elect to use this space for your data sets, you may create a migration problem for your next release. (Do you have enough server disk space to have this volume plus all the next release volumes online?) SYS1.ADCD06.PARMLIB and the SMP/E target zone CSI are on this volume. For P/390, R/390, and Integrated Server systems, it restores as two OS/2 or AIX files: OS39R6_1.A80 (2.1 GB) and OS39R6_2.A80 (.8 GB), and is addressed as A80 in the distributed DEVMAPs.
- **OS3R6A** is an overflow volume for OS39R6, and contains more static OS/390 data sets. It has been trimmed to an "odd" size to conserve space on P/390 and similar systems. This is a required volume for any IPL. For P/390, R/390, and Integrated Server systems, it restores as a single file, OS3R6A.A84 (1.88 GB).
- **OS39H6** contains all the system-provided HFS files. Note that these are now in the root file system, in contrast to earlier releases where system-provided HFS contents were spread over multiple file systems. This volume has been trimmed to 1500 cylinders of 3380 space to conserve total disk space. Note that this file system will not normally contain /u files. Although it is possible to run this release with no HFS volume, it is not very practical; for practical purposes, this is a required volume for IPL. This is an SMS-managed volume. For P/390, R/390, and Integrated Server systems, it restores as a single file, OS39H6.124 (1.07 GB).

[3] This requirement to build two versions of every AD CD-ROM is one of the (many) reasons for the delay between a new release of OS/390 and the corresponding AD CD-ROM.

- **SCPMV5** contains most of the dynamic OS/390 data sets, such as spool, paging, catalogs, PARMLIB, and so forth. This volume is defined as a STORAGE volume and a reasonable amount of free space is needed for temporary data sets used during normal system operation. A double-density 3380 volume is used. *Please note that future releases may use a 3390 volume instead.* This volume is required for any IPL. Among other data sets, it contains the master catalog, SYS1.PARMLIB, SYS1.IPLPARM, SYS1.VTAMLST, SYS1.IODF00, the RACF data base, the JES2 spool file and checkpoint, and the SMP/E global zone CSI. There are about 7500 tracks free on this volume. This is the only STORAGE volume in the distributed volume set, and it tends to accumulate user data sets. You should monitor the available space on this volume or, better yet, allocate at least one local volume with the STORAGE attribute. (This attribute is set in a VATLST00 member in PARMLIB.) For P/390, R/390, and Integrated Server systems, it restores as a single file, SCPMV5.122 (1.26 GB).
- **OS39SV** contains a separate SMP/E global CSI, with an alternate COBOL compiler received in it. This volume also contains all the service shipped with the ServerPac that was used to build this AD CD-ROM. This volume is *not* required for IPL, and we expect many users may elect not to restore this volume if they are short on disk space. You may temporarily need it if you are forced to apply some of the PTFs it contains. For P/390, R/390, and Integrated Server systems, it restores as a single file, OS39SV.A84 (0.3 GB).
- **HFSUS1** is a new volume that contains an almost empty HFS file system. This file system is mounted as /u, and would probably be used for much of your local OE files. The size is arbitrary, at 200 cylinders. This volume is not required for an IPL, although you may need to remove its MOUNT command from the BPXPRM00 PARMLIB member if you do not want to use it. For P/390, R/390, and Integrated Server systems, it restores as a single file, HFSUS1.125 (0.14 GB).
- **OS39D6** contains the system DLIBs and the DLIB zone CSI. This volume is not required for IPL. You will need it if you apply PTFs. There is relatively little free space on this volume. This has the potential for problems if you install additional products or try to apply large volumes of fixes. For P/390, R/390, and Integrated Server systems, it restores as two files, OS39D6_1.A81 (2.14 GB), and OS39D6_2.A81 (0.7 GB).
- **P390DX** contains CICS, DB2, and IMS modules. It contains the DLIB and target SMP/E zone CSIs of the these products. This volume is not required to IPL. For P/390, R/390, and Integrated Server systems, it restores as two files, P390DX_1.A83 (2.14 GB), and P390DX_2.A83 (0.7 GB). (Earlier releases used a different address and device type for the P390DX volume.)

If you restore all volumes from the CD-ROMs, you will need about1 about 10.2 GB disk space on your server. If you use only OS39R6, OS3R6A, SCPMV5, and OS39H6 you will need about 7 GB disk space for a P/390 or R/390 system -- assuming that there is no loss due to hard disk boundaries.[4]

The following table provides a quick summary of the volumes included with the OS/390 R6 AD CD-ROM system.

[4] If all your space appears as a single large disk (typical RAID 5 or LVM setup) setup) this is the space required. If you are using discrete disks (not RAID 5 or LVM) then each volume must be completely contained on one disk.

Table 1. Volumes included with the AD CD-ROM systems							
OS/390 R6 volsers:	SCPMV5	OS39R6, OS3R6A	OS39H6	HFSUS1	OS39SV	P390DX	OS39D6
required to IPL	yes	yes	probably	probably	no	no	no
usually contains locally customized files	yes	no	yes	probably	no	Yes, if used	no
contains catalog	master	yes	yes	no	no	yes	yes
mounted as:	storage	private	private (SMS)	private (SMS)	private	private	private

Several of the IBM-provided volumes have considerable amounts of free space. Should you use this space? The considerations involved are:

1. Do you plan to apply maintenance (that is, PTFs) or install additional OS/390 products? If so, you will need some of the free space on the system volumes. Discussions with many AD CD-ROM users indicate that most do not routinely install PTFs or add additional OS/390 products. (This does not preclude the installation of a small number of PTFs that address a specific problem.) If you do not plan to install PTFs/products, we suggest that you keep a few cylinders, perhaps 10 - 20, free on these volumes[5], and use the remaining free space if appropriate.

2. All volumes, except SCPMV5 and HFSUS1, will be replaced by newer releases of the AD CD-ROM system. Any data sets you place on OS39R6, OS3R6A, OS39D6, OS39H6, OS39SV, or P390DX might be lost when you install a new release. We say *might* because you can easily retain these volumes[6] after installing a new release if you have sufficient disk space on your server. The typical P/390 system probably does not have sufficient disk space to hold an old release of OS/390, a new release, and local volumes.

3. If you use the migration process provided with the AD CD-ROM system[7], your SCPMV5 volume will be retained when you install a new release. Thus, any data sets you place there will be retained. However, please note that SCPMV5 is the only *STORAGE* or *PUBLIC* volume[8] provided with the AD CD-ROM system. (You can create new volumes and mount them as STORAGE or PUBLIC (via the VATLSTxx member in PARMLIB), but this is not done automatically.) You should ensure that there is *at least*, say, 100 cylinders free on SCPMV5 if there are no other STORAGE/PUBLIC volumes. If you do not use the migration process, or your own equivalent, then SCPMV5 will be replaced when you install a new release of OS/390 and any data left there will be lost.

4. We suggest that you do not place your MVS data sets on OS39H6 or HFSUS1. These are DFSMS-managed volumes, created to hold HFS data. You will need to learn to work with DFSMS controls and procedures if you use these volumes for non-HFS data sets. Unless you have a specific need to use

5 This space is most likely to be used in obtaining additional extents for existing system data sets if the data sets are expanded by PTFs or by simple edit/save activity. If your space is very tight, you might compress any PDSs that you update. A PTF might allocate a new data set, for which you would need space, but few PTFs do this.

6 You would need to assign new volume serials to OS39SV and P390DX if you install a new release that includes replacements for these volumes.

7 Informal surveys indicate that relatively few installations use this process.

8 These volumes are used for temporary data sets, such as compiler work areas, ISPF logs, and so forth.

DFSMS functions, we suggest that it is an additional complexity that you can avoid.

5. For HFS files, you can place your data on HFSUS1. Although this volume is provided on the AD CD-ROMs, it contains no files. It is intended for your files, provided your files are in the /u path. That is, the HFS on HFSUS1 is mounted at mount point /u in the root file system. You can change this by altering the BPXPRMxx member in PARMLIB. When the next release of the AD CD-ROM arrives, you would not install the HFSUS1 volume -- you would retain your current HFSUS1 volume.

6. We suggest you create your own emulated 3380 or 3390 volumes for your data, plus at least one volume for STORAGE/PUBLIC usage. Unless you are very constrained for server disk space, using the free space on the system volumes (instead of allocating your own additional volumes) is likely to lead to problems when you move to a new release. The author usually creates one or two 300-cylinder 3390 volumes for data sets, and retains these volumes across new releases of OS/390.

7. If you understand these considerations and still want to use the free space on the system volumes, then use the space. There are no hidden OS/390 functions that will be disturbed.

2.4 Manually Loading MVS Volumes From CD-ROM

{ZB20} Instead of using the INSTALL command to install all the volumes on all the CD-ROMs, you can manually install individual OS/390 volumes from the CD-ROMs. This can be useful if you need to reload a volume to recover from an error, or if you want to control the placement of the OS/390 volumes on several OS/2 drives.

1. Insert the OS/390 CD-ROM disk 1 and open an OS/2 window.

2. Examine the file x:\OS390R6.PAC (where x: is the drive letter of your CD-ROM drive). This will tell you the names and locations of the OS/2 files that contain each of the MVS volumes. For release 6, the CDs layout is (these files are always in \OS390 subdirectories on each CD):

   ```
   CD Number 1: OS39R6
   CD Number 2: OS3R6A, HFSUS1
   CD Number 3: P390DX, OS39H6
   CD Number 4: SCPMV5, OS39SV
   CD Number 5: OS39D6
   ```

3. Insert the CD-ROM disk containing the desired volume and enter the following command in the OS/2 window:

   ```
   UNZIP  x:\OS390\fname.ZIP  *.*  -d d:\

   where 'd:\' is the drive where you want to install the
   MVS volume and 'fname' is the MVS volume name.  x is the
   drive letter of the CD ROM drive.
   ```

4. If you place the file anywhere other than D:\OS390\ you will need to update the P/390 DEVMAP before the file can be used. In the DEVMAP, just tab over to the "FN/P" field for the DASD device you want to change, and overtype the actual drive/path/filename and press ENTER.

2.5 Initial P/390 Device Map

{ZB25} (Most of the comments in this chapter that mention the P/390 also apply to the R/390 or IBM Integrated Server, with minor differences for the R/390.) The P/390 device map (DEVMAP) distributed with the AD CD-ROM is copied to D:\OS390\DEVMAP.MVS if you use the INSTALL command (or manually copy the file). It is defined as follows:

```
ADR  DEV   MANAGER  PARMETERS              COMMENTS

00C  2540  AWS2540
00E  1403  AWS2821
00F  1403  AWS2821
122  3380  AWSCKD   D:\OS390\SCPMV5.122     Local data sets
125  3380  AWSCKD   D:\OS390\HFSUS1.125     For /u files
560  3480  SCSI3480                         SCSI tape (probably 4mm)
580  3420  AWSTAPE  D:\TAPE.580             tape emulated on disk
581  3420  AWSTAPE  D:\TAPE.581             tape emulated on disk
700  3278  AWS3274                          MVS Master Console
701  3278  AWS3274                          Local VTAM session
702  3278  AWS3274                          Local VTAM session
900  3278  AWS3274                          Local VTAM session
901  3278  AWS3274                          Local VTAM session
902* 3278  LAN3274                          LAN VTAM session
903* 3278  LAN3274                          LAN VTAM session
A80  3390  AWSCKD   D:\OS390\OS39R6_1.A80   IPL volume
A81  3390  AWSCKD   D:\OS390\OS39D6_1.A81   DLIBs
A82  3390  AWSCKD   D:\OS390\OS3R6A.A82     Extension of IPL volume
A83  3390  AWSCKD   D:\OS390\P390DX_1.A83   DB2, CICS, IMS
A84  3390  AWSCKD   D:\OS390\OS39SV         COBOL, PTFs, etc
E20* 3088  LCS3172                          For TCP/IP (even addr)
E21* 3088  LCS3172                          For TCP/IP (odd addr)
E40* 3088  LAN3172                          For SNA
```

```
* = Device is ″excluded″ in DEVMAP.
```

If you have a LAN adapter installed in your PC Server, you may want to ′activate′ the excluded entries that are LAN-related. Excluded devices are shown on the P/390 Configurator F2 Update System Devices menu in dark blue. The LAN devices contained in this DEVMAP are:

```
902-903 LAN3274: 3270 sessions over TCP/IP or CM/2
E20-E21 LCS3172: TCP/IP 3172 emulation
E40     AWS3172: VTAM 3172 emulation
```

To activate excluded devices when using the P/390 configurator, position the cursor on the first device and press ALT+F5. See the documentation or online help (F1) for additional details.

The following OS/2 ″pop-up″ message during IPL

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

means that you need to activate the SCSI3480.SYS device driver in CONFIG.SYS. Address 560 (in the DEVMAP shown above) 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 14.3, “How to Prepare a P/390 for TCP/IP” on page 257 or 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.

2.6 IPLing the AD CD-ROM System

{ZB30} If you copy the DEVMAP from the CDs (and this is done automatically if you use the INSTALL command), the system is set up to auto-IPL; you do not need to respond to any messages. It will IPL using the CS IPLPARM; this produces a “cold start,” meaning a CLPA and JES2 cold start. After the first IPL, you should set the IPLPARM to WS (Warm Start) by changing the parameter in the P/390 configurator (use PF4 in the primary menu of the configurator). The CS option takes considerably longer to IPL; it is very unlikely that you will want to use CS every time you IPL.

If you plan to use DB2, you should IPL with the DC parameter (again, in the P/390 configurator); this also performs a cold start but includes DB2 modules in the LPA list. After IPLing with DC, you would thereafter IPL with DB (warm start). In summary,

```
IPL once with CS and use WS thereafter, or
IPL once with DC and use DB thereafter.

You can use CS followed by WS followed by DC followed by DB.
You can use CS followed by DC followed by DB.

Do not use CS followed by DB.
Do not use DC followed by WS.
```

See 2.10, “IPL Parameters and Parmlibs” on page 14 for a more complete description of the various IPL parameters used with the AD system.

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.

When cold starting JES2, you will receive this message:

```
* 12.29.18       *$HASP436 CONFIRM COLD START ON
* CKPT1 - VOLSER=SCPMV5 DSN=SYS1.HASPCKPT
* CKPT2 - NOT IN USE
* SPOOL - PREFIX=SCPMV  DSN=SYS1.HASPACE
*12.29.18       *02 $HASP441 REPLY ′Y′ TO CONTINUE INITIALIZATION OR
* ′N′ TO TERMINATE IN RESPONSE TO MESSAGE HASP436
```

You should reply R 02,Y (using the correct reply number, which might not be 02).

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. Please 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). : You can log onto TSO when you see the following message:

```
IKT007I TCAS ACCEPTING LOGONS
```

2.7 Shutting Down the System

{ZB35} To stop your system, you can issue S SHUTDOWN from the Master Console. This may, or may not, produce a clean shutdown. You can edit the script in SYS1.ADCD06.PARMLIB(SHUTDOWN). You may want to, for example, reduce the 30 second delay after the message is sent to TSO users. (See script for details.)

The distributed R6 AD CD-ROM system has a minor error in the shutdown script. A few lines into the script it starts the Web Server instead of stopping it. You should edit SYS1.ADCD06.PARMLIB(SHUTDOWN) and change “S IMWEBSRV” to “P IMWEBSRV.”

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.

2.7.1.1 Device Addresses Defined

{ZB50} The following device addresses are defined in the AD system (that is, defined in the active IODF):

```
ADDRESSES       DEVICE

00C             2540
00E             1403
00F             3203
010             3277
063             3277
120-15F         3380        (64 devices)
240-25F         3380        (32 devices)
260-27F         3390        (32 devices)
560-57F         3480        (no compaction)
580-5AF         3400
700             3270
701-73F         3277        (63 devices)
900-91F         3277        (32 devices)
A80-ABF         3390        (64 devices)
E20-E27         CTC         ( 8 devices)
E40-F47         CTC         ( 8 devices)
```

There are large numbers of devices defined in OS/390. Remember that a P/390 or R/390 can have a maximum of 255 devices defined in its device map, so you can actually use only a subset of the OS/390 devices that are defined in the IODF provided with the AD system.

When possible, you should assign your devices (emulated disks, and so forth) to addresses in this list. If you cannot do this, you will need to use the HCD utility to alter the OS/390-defined devices. Note that none of the defined 3480 devices have the *compaction* feature. If you need this, you will need to use HCD to modify your IODF. The active IODF in the V2R6 AD CD-ROM is SYS1.IODF00.

Contrary to the README.MVS file on the AD CD-ROMs, there are no 3480 drives defined with the IDRC (or COMPACT) feature. If you need such drives, you will need to use HCD to create additional 3480 drives or modify existing 3480 definitions.

2.8 Catalogs

{ZB52} In the R6 AD system, the following catalogs are on the indicated volumes:

```
VOLSER   CATALOG NAME
SCPMV5   OS390.MASTER.CATALOG
OS3R6A   USERCAT.S390PROD
OS39D6   USERCAT.S390DLIB
OS39SV   USERCAT.S390CUST
P390DX   USERCAT.P390DX
```

In addition to many data set entries, the master catalog contains a large number of aliases pointing to other catalogs. The aliases are:

```
Target = USERCAT.S390DLIB
Aliases = DLIB

Target = USERCAT.S390CUST
Aliases = CUSTOM

Target = USERCAT.P390DX
Aliases = CICS, DGO, DSN510, IMS, IMSRLM61, OS39D6AD, QMF

Target = USERCAT.S390PROD
Aliases = ABJ, ADSM, ANF, AOP, API, APX, ASMA, ASMT, ASU, BFS, CBC,
          CEE, CPAC, CSF, DCE, DFS, DIT, ECN, EDC, EDF, EHW, EJS,
          EDY, EPH, EQAW, WUV, EWX, EZM, FFST, FIREWALL, GDDM,
          GIM, GLD, HFS, ICA, ICQ, IEL, IGY, IMW, IOA, IOE, ISP,
          JAVA, LDAP, MQM, MVSNFS, NETVIEW, NPM, OS39R6AD, PLI,
          PSF, REXX, SCRIPT, SOMMVS, SYSMVIEW, TCPIP, TME10GEM
```

Note that there are no aliases for the defined TSO userids. This means that any datasets cataloged under TSO will be cataloged in the master catalog. ISPF profile data sets do not exist for all the defined userids; they are created when the userid is first used. By default, they will be placed on SCPMV5, since it is the only STORAGE volume in the distributed AD CD-ROM system. If you plan to use aliases to place TSO user data sets in a user catalog, you should do this before using any of the userids (otherwise you will need to uncatalog the ISPF profile data sets before you can create the alias).

2.9 Paging Data Sets

{ZB54} The R6 AD CD-ROM has the following paging data sets defined:

```
PLPA    SYS1.PLPA1.PAGE      120 cylinders          (58% used)

COMMON  SYS1.COMMON.PAGE      50 cylinders          ( 6% used)

LOCAL   SYS1.LOCAL1.PAGE     300 cylinders          (20% used)
        VSAM data name is P390M5.LOCAL1.PAGE.DATA

LOCAL   SYS1.PLPA.PAGE        55 cylinders          (77% used)
        VSAM data name is P390M5.SYS1.PLPA.PAGE.DATA
```

The amount used is with a single TSO user logged onto the system. All four paging data sets are on SCPMV5. The data set names (and the data name, since these are VSAM data sets and VSAM data sets have multiple elements with names) are misleading for the two LOCAL paging data sets. These names were selected when the first release of the AD system was build. Changing the names now, to better reflect their use, would disrupt the migration process that accompanies the AD CD-ROM systems.

2.10 IPL Parameters and Parmlibs

{ZB56} This release provides six IPL parameters. (These are usually entered in the IPLPARM field in the F4 display of a P/390 or R/390 system. The complete field is normally 0122xx, where xx is one of the following parameters and 0122 is the address of the PARMLIB volume.) The parameters are:

- **CS** performs a CLPA function (without DB2 or IMS modules) and cold starts JES2.
- **WS** uses the existing PLPA and warm starts JES2.
- **DI** performs a CLPA function (including DB2 and IMS modules) and cold starts JES2.
- **DB** uses the existing PLPA (which should have been created by using a DI IPL), and starts DB2. JES2 is warm started.
- **DC** uses the existing PLPA (which should have been created by using a DI IPL) and starts DB2. JES2 is cold started.
- **00** is not normally used. It does not concatenate the second PARMLIB; it uses the existing PLPA and warm starts JES2. DB2 functions are not started.

Several other IPLPARM members exist (with suffix values 01, 03, 04, 05, 06, 10, NUCLIST00); you should ignore these. They are leftovers from the building and test process. Member SAMPLE contains useful comments.

Two parameter libraries are used: SYS1.ADCD06.PARMLIB and SYS1.PARMLIB. Except for the “00” IPLPARM option, SYS1.ADCD06.PARMLIB is always searched before SYS1.PARMLIB. If the target member is found in SYS1.ADCD06.PARMLIB, then SYS1.PARMLIB is not searched for the member.

The current arrangement uses “00”-suffix PARMLIB members for almost all functions. Where 00-suffix PARMLIB members are not used, the suffix corresponds to one of the IPLPARM values (usually DB) and the member is taken from SYS1.ADCD06.PARMLIB.[9]

For an IPL using the IPLPARM selection CS, parmlib members were taken from the following sources, in the order they were used:

```
           +---------Taken from SYS1.ADCD06.PARMLIB
           |
           |  +------Taken from SYS1.PARMLIB
           |  |
           |  |  +---Various members used, depending on
           |  |  |   the IPLPARM

MEMBER     A  S  V   Purpose.............................

IEASYM00   A         Establish system symbols
IEASYS00   A         System parameters
IEASYSCS   A     V   Override the IEASYS00 parameters
IEASVC00   A         Define local SVCs
PROG00     A     V   Authorized libs and LNKLST
LPALST00   A     V   List data sets in link pack area
IEAPAK00      S      LPA pack list
IEAFIX00      S      Fixed LPA list
IEALPA00      S      Modified LPA list
DIAG00        S      Storage tracking
IEAOPT00      S      SRM controls
IEAICS00   A         More SRM controls
IEAIPS00      S      Yet more SRM controls
IEAABD00   A         SYSABEND dump controls
```

[9] The only exceptions are CTIJESxx members that are taken from SYS1.PARMLIB.

```
IEADMP00     S      SYSUDUMP dump controls
IEADMR00     S      SYSMDUMP dump controls
CLOCK00   A         Clock offset from GMT and Sysplex clock
COUPLE00  A      V  Sysplex coupling controls
CTIXCF00     S      Component trace for XCF
CTIXES00     S      Component trace for XES
CTIGRS00     S      Component trace for GRS
IGDSMS00     S      SMS definitions
CTISMS00     S      Component trace for SMS
IFAPRD00  A         Licensed product control
CONSOL00     S      MVS console definitions
CTIOPS00     S      Component trace
IEACMD00  A         Commands for IPCS and SLIP
COMMNDCS  A      V  Automatic commands at IPL time
MSTJCL00  A         Source of master JCL
SCHED00   A         PPT (Program Properties Table)
IECIOS00     S      Missing interrupt handler controls
VATLST00  A         Volume usage attributes
BPXPRM00  A         OMVS parameters
IKJTSO00  A         TCAS and TSO parameters
PFKTAB00     S      Master console PFK setup
SMFPRM00     S      SMF recording controls
IEFSSN00  A      V  Subsystem startup
IEASLP00  A         SLIP automatic commands
ADYSET00     S      Dump suppression (DAE) controls
MPFLST00  A         Message Suppression
CTIPBX00  A         Component trace for OMVS
CTILOG00  A         Component trace for Logger
CTIJES01     S      Component trace for JES2
CTIJES02     S      Component trace for JES2
CTIJES03     S      Component trace for JES2
JES2PARM  A         JES2 parameters
CTIJES04     S      Component trace for JES2
IVTPRM00  A         Communication storage mgr (VTAM)
COFVLF00  A         VLF Parameters
CTIEZB00  A         Component trace for TCP/IP
```

The "V" flag indicates that different suffixes are used, depending on the IPLPARM (which selects different IEASYSxx members, which have different suffix values for the indicated members). If you need to change PARMLIB values, we suggest you use the above table to determine which member to change, in which library.

The various IPLPARM values affect these PARMLIB members:

```
IPLPARM     Uses members (other than 00 members)............

  CS        IEASYSCS COMMNDCS
  DB        IEASYSDB COMMNDDB COUPLEDB LPALSTDB PROGDB SCHEDDB IEFSSNDB IEASVCDB
  DI        IEASYSDI COMMNDDI LPALSTDB PROGDB SCHEDDB IEFSSNDB IEASVCDB
  DC        IEASYSDC COMMNDDC COUPLEDB LPALSTDB PROGDB SCHEDDB IEFSSNDB IEASVCDB
  WS        IEASYS00
```

Member names not shown here are the suffix "00" members. That is, IPLPARM xx (which results in the use of an SYS1.IPLPARM(LOADxx) member, which specifies an IEASYSzz member) will use many PARMLIB members with suffix 00. Only the non-00 members are listed here.

An important result of selecting different IPLPARM values is the use of different COMMNDxx members. It is the COMMNDxx contents that, to a large extent, produces the different results of the different IPLPARM specifications. The COMMNDCS member has these commands:

```
S JES2,PARM='COLD,NOREQ'
S VTAM,,,(LIST=00)
SET MPF=00
S IRRDPTAB
S VTAMAPPL
S EZAZSSI,P=P390
DUMPDS ADD,VOL=(SCPMV5)
DUMPDS (name specifications)
DUMPDS ALLOC=ACTIVE
```

And SYS1.ADCD06.PARMLIB(VTAMAPPL) contains (ignoring comments and time delay commands):

```
S VLF,SUB=MSTR
S LLA,SUB=MSTR
S TCAS
S TCPIP
D T
```

Other COMMNDxx members are the same as COMMNDCS, with these changes:

```
COMMNDDB and COMMND00 warm start JES2 (instead of a cold start)

COMMNDDC is the same as COMMNDCS

COMMNDDI adds or changes these lines:
         'S VTAMIMS,,,(LIST=IM)'
         'S IRLM'
         'S JRLM'
```

2.11 RACF Parameters and Profiles

{ZB58} The following userids are defined:

```
USERID   PASSWORD   GROUP   UID  DAYS   Attributes.....

IBMUSER  IBMUSER    SYS1     0     30   SPECIAL OPERATIONS
P390     P390       SYS1     0     30   SPECIAL OPERATIONS AUDITOR

OPEN1    SYS1       SYS1     0    180   none
OPEN2    SYS1       SYS1     0    180   none
OPEN3    SYS1       SYS1     0    180   none

P390A    TEST       TEST     -     30   SPECIAL OPERATIONS GRPACC
P390B    TEST       TEST     -     30   none
...
P390Z    TEST       TEST     -     30   none

Where GROUP is the default group, UID is the OMVS assigned
UID (and - indicates no OMVS definition for the user), DAYS
is the password interval, and Attributes lists the RACF
special attributes for the user.
```

The early system we examined had 96 userids defined, including those listed above. Many of the additional userids are intended to be used for started tasks. (You can list all the defined userids by using the RACF panels, and performing a SEARCH with no arguments. You must have the SPECIAL attribute to do this.)

The same system had 102 groups defined. We suggest you do not use any of these groups (except SYS1 and TEST) unless you have a specific need to do so.

There was no default UID/GID defined in our early system. That is, if a user (such as P390A) attempted to use OMVS, he would fail. You would need to either define an OMVS segment for P390A, or define a default OpenEdition user. See 7.17, “How to Define A Default OpenEdition Userid” on page 102 for defining a default OpenEdition user.

Several RACF General Resource Classes are critical for many functions. In the following lists, access levels are indicated by N=NONE, R=READ, U=UPDATE, and A=ALTER. The FACILITY class profiles are:

```
PROFILE               UACC Access List.................

BPX.DAEMON              N  IBMUSER(A) OMVSKERN(R) WEBSRV(R) FTPD(R)
PBX.SERVER              N  P390(A) WEBSRV(U)
PBX.SUPERUSER           N  IBMUSER(A)
CBD.CPC.IOCDS           N  P390(A)
CBD.CPC.IPLPRM          N  P390(A)
DCEKERN.START.REQUEST   N  IBMUSER(U) DCEKERN(U)
DFSKERN.START.REQUEST   N  IBMUSER(U) DCEKERN(U)
IRRDPI00                R  P390(A)
DITTO.* (G)             R  IBMUSER(A) SYS1(A)
STGADMIN.ADR.* (G)      N  P390(A) SYS1(A)
STGADMIN.IDC.* (G)      N  P390(A) SYS1(A)
STGADMIN.IGD.* (G)      N  P390(A) SYS1(A)
STGADMIN.IGG.* (G)      N  P390(A) SYS1(A)
STGADMIN.* (G)          N  P390(A) SYS1(A)
```

The TSOAUTH class profiles are:

```
PROFILE               UACC Access List.................

ACCT                   N   IBMUSER(A) P390(R)
JCL                    N   all
MOUNT                  N   IBMUSER(A)
OPER                   N   IBMUSER(A) P390A(R) P390(R)
RECOVER                N   all
```

The ACCTNUM class profiles are:

```
PROFILE               UACC Access List.................

ACCT#                  N   all
```

The TSOPROC class profiles are:

```
PROFILE               UACC Access List.................

DBSPROC                N   P390(A) DSN1SPAS(R) SYSADM(R) SYSOPR(R)
IKJACCNT               N   IBMUSER(A)
ISPFPRCB               N   IBMUSER(R) P390(R)
ISPFPROC               N   all
OMVSPROC               N   IBMUSER(A) OPEN1(R) OPEN2(R) OPEN3(R)
```

There are no STARTED class profiles defined, although the STARTED class is active. Please note that this situation -- no STARTED class profiles -- may not exist in future releases.

Currently, there is a userid defined[10] for every started task provided with the AD CD-ROM system. These userids include:

```
DUMPSRV   OMVS      XCFAS     IXGLOGER  SMF       WLM       CATALOG
VTAMAPPL  IRRDPTAB  VTAM      VTAMIMS   BLSUPRMI  EZAZSSI   JES2
RACF      IOSAS     JESXCF    ANTMAIN   DFSCM     SMS       ANTAS00
INIT      NFSC      VMCF      BPXOINIT  BPXAS     VLF       TCAS
LLA       TCPIP     IMWEBSRV  NFSS      FTPD      CICS
and others
```

Do not delete these userids, unless you convert to RACF′s STARTED class controls. As provided, these userids depend on a RACF exit (ICHRIN03) to provide trusted system privileges when they are started. The same default ICHRIN03 entry is used for all started tasks. The console message indicating that a STARTED class entry was not found and that the ICHRIN03 parameters are used has been suppressed (by placing the message number in the MPF list in PARMLIB). The messages appear in the full system log.

If you add your own started tasks, you can use the same process --that is, add a userid corresponding to your started task name.

Userids associated with OpenEdition jobs (which may include indirect use of OpenEdition facilities) should have OMVS segments containing UIDs. These userids should also be associated with groupids that have OMVS segments with GIDs. We found the following groupids have the indicated GIDs:

```
DCEGRP     GID=2          EMPLOYEE   GID=500         IMWEB    GID=205
DFSGRP     GID=2          EXTERNAL   GID=999         SPECIAL  GID=255
OMVSGRP    GID=1          SYS1       GID=0           TTY      GID=0
```

There are no special meanings for GID=0. Note that group TEST (which is the default group for most of the userids supplied with the system) does not have an OMVS segment or GID.

2.12 DFSMS Definitions and Planning

{ZB60} The DFSMS control data sets are:

```
SCDS = SYS1.SCDS
ACDS = SYS1.ACDS
COMMDS = SYS1.COMMDS

System Name = P390
```

The following DFSMS classes and ACS routines have been added to the AD CD-ROM system:

```
   TYPE                     NAME

Management Class         DEFAULT
Data Class               --none--
Storage Class            DEFAULT
Storage Group            DEFAULT
```

The corresponding ACS routines are in SYS1.P390.CNTL with member names starting ACSxxx. Note that the name DEFAULT was chosen; it does not imply a

[10] These userids are defined without a TSO segment, providing an additional measure of assurance that they will not be misused.

default action. The ACS routines do nothing except assign the name DEFAULT to their appropriate DFSMS variable.

DFSMS function are appropriate for large installations, with hundreds or thousands of DASD volumes. P/390 and R/390 users do not need these functions and normally ignore DFSMS to the extent possible. One area that cannot be ignored is that volumes containing HFS files *must* reside on DFSMS-managed volumes. Two such volumes are provided with the current system: OS39H6 and HFSUS1.

If you want to create your own volumes for HFS files, you must make them DFSMS-managed. The most basic way to do this involves:

1. Indicating that they are managed when you initialize them with ICKDSF, and
2. Adding the new volsers to a DFSMS Storage Group list.

Unfortunately, changing a storage group list requires a number of steps to modify a source control file and then create a new active control file for SMS. One way to bypass these details is to use a volser that is already defined in the DEFAULT Storage Group. Fortunately, the definitions in the Storage Group have not been cleaned up, to remove old volsers from previous AD CD-ROM lists. Our system had volsers OS39H1, OS39H2, OS39H3, and so forth defined. You can simply create a new volume (3380 or 3390, remembering to use the "managed" option when you initialize the volume) using one of these volsers and you will not need to change anything in DFSMS.

You should verify what volsers are defined in the DEFAULT Storage Group for your system. To do this, use the following ISPF steps:

```
- ISPF primary menu
M Select the More option
  2 Select ISMF
    0 ISMF Profile
      0 User Mode Selection
        Set User Mode = 2  (Storage Administrator)
        PF3
      PF3 to exit from ISMF
  2 Start ISMF again
    6 Storage Group
      CDS Name = 'ACTIVE'
      Storage Group Name = 'DEFAULT' or '*'
      Enter
        In the Storage Group List, in the Line Operation field
        of the DEFAULT group, enter 'LISTVOL'  (without the quotes).
        Press Enter.
        You should obtain a list of all volsers that are already
        members of this Storage Group.
```

In our R6 system, the LISTVOL command displayed volsers OS39H1, OS39H2, OS39H3, and OS39H4. It did not display OS39H6 and HFSUS1, which are the volsers of the active HFS volumes for the R6 AD system. However, OS39H6 and HFSUS1 function correctly. We cannot explain this situation.

2.13 Initial MVS Console PFK Assignments

{ZB62} The MVS console PFK assignments distributed with the AD CD-ROM are:

```
  ACTION                                        PFK:
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).

2.14 ISPF Usage

{ZB64} The second panel of the primary ISPF menu, accessed through the "M" (More) option on the first panel, contains:

```
 1 SMP/E
 2 ISMF
 3 RACF
 4 HCD
 5 SDSF
 6 IPCS
 7 DITTO
 8 RMF
 9 DFSORT
10 OMVS
```

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.

The ISPFPROC logon procedure for TSO, as supplied with the AD system, contains a large number of allocations. Many of these are needed only for the systems programmer functions accessed through ISPF, such as RACF, RMF, ISMF, and so forth. Creating a new logon procedure, with many fewer allocations, substantially decreases TSO logon time.[11] A sample shortened procedure was given in the *Cookbook*. It appears that the same example in the Cookbook should work for the V2R6 release. The ISPFPROC included in V2R6 has a few changes from previous releases -- mostly in the area of HLQs -- but these changes are to DD statements that are removed for the faster logon.

[11] The tradeoff, of course, is that you cannot use the ISPF panels corresponding to the removed DD statements in your shorter logon procedure. If you sometimes need these functions, simply log on using the full ISPFPROC procedure.

2.15 Supplied Jobs

{ZB66} 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. Jobs (corresponding to member names in the data set) include:

- **ADDID**, **ADDIDS**, and **ADDWEDID** all add TSO userids. The last one also defines many of the group ids and system data set protection profiles. All of these jobs also add users to SYS1.UADS; you probably do not want to do this. The example of adding a userid that appears in the *Cookbook* may be more useful.
- **DMPFULL** provides a standard job to backup a full DASD volume to tape.
- **DSFINIT** provides a job to initialize an SMS-managed volume, such as you would use for HFS. (Remove the STORAGEGROUP operand to initialize a non-SMS managed volume.)
- **HFSALLOC** is a job to create a new HFS. It assumes the indicated volser has been initialized as SMS-managed.
- **HFSCONV** converts a non-SMS managed volume to an SMS managed volume without completely initializing the volume.
- **HFSDSF** has a misleading name. It initializes a volume WITHOUT the SMS-managed element that would be needed for an HFS volume.
- **HFSDUMP** dumps an HFS (which is a single data set, as far as MVS is concerned). This is a dump-by-dataset and not a dump-by-volume (or physical dump) job. **HFSREST** is the matching restore job.
- **HFSNEW** is a batch job that executes a TSO MOUNT command to dynamically mount a new HFS. (An authorized user could issue the same command from TSO.)
- **LISTCAT** lists the complete master catalog. You need to correct the name of the master catalog in the JCL. The current name is OS390.MASTER.CATALOG.
- **NFS** and **NFSRACF** are the jobs used to initialize NFS data sets for the V2R6 release. You will not need to run this job (again), but it provides an overview of the NFS setup.

2.16 SMP/E Data Sets

{ZB68} The key SMP/E data sets are:

```
VOLSER   NAME      Dataset.....................   TYPE

SCPMV5   GLOBAL    MVS5.GLOBAL.CSI                GLOBAL
OS39D6   MVSD100   OS39D6AD.MVSD100.CSI           DLIB
OS39D6   MVSD101   OS39D6AD.MVSD100.CSI           DLIB
OS39R6   MVST100   OS39R6AD.MVST100.CSI           TARGET
OS39R6   MVST101   OS39R6AD.MVST100.CSI           TARGET
```

The LTS, MTS, SCDS, and other SMP/E secondary data sets are on volume OS3R6A. Note that the MVSD101 and MVST101 CSIs contain only JES3 materials. All of the base system is in the MVSD100 and MVST100 CSIs.

Separate Global CSIs are included for CICS, IMS, and DB2. The easiest way to locate these is to use ISPF option 3.4 and search for data sets with names ***.*.CSI**. While there is no rule against it, the only data sets in the AD CD-ROM system with the LLQ of CSI are the SMP/E data sets.

2.17 Minor Adjustments

{ZB70} Using the P390 userid (or any other userid with UID=0), you need to go to OMVS and:

```
cd /
chmod 755 u      (or whatever permissions you want)
```

If this is not done, OMVS usage by non-UID=0 users will fail. You might first see this as a failure when such userids telnet to the system.

If you try to create SYS1.IODF00.WORK, HCD will inform you that it already exists. If you try to use it, you will receive error messages from IDCAMS. You should use HCD to delete SYS1.IODF00.WORK (because it does not really exist). You can then create a new SYS1.IODF00.WORK if you need it. Always place IODF data sets on SCPMV5.

The COBOL environment has changed from earlier releases of the AD system. Earlier releases used COBOL for OS/390 and VM, which is the most current release of COBOL. In response to many requests, the R6 AD system uses an older version of COBOL. Unfortunately, while making these changes, the COBOL compiler library was dropped from LNKLST, the COBOL procedure library was dropped from the JES2 concatenation of procedure libraries, and the wrong runtime routines appear first in LNKLST.

You can correct this several ways. The easiest way, without changing any of the system setup, is to add several JCL statements to a COBOL compile (or CLG) job:

```
//MYJOB  JOB .........
//   JCLLIB ORDER=(IGY.V1R2M0.SIGYPROC)
//JOBLIB DD DISP=SHR,DSN=IGY.V1R2M0.SIGYCOMP
//       DD DISP=SHR,DSN=SYS1.SCEERUN
//MYCLG  EXEC  IGYWCLG
.
```

The JCLLIB[12] statement lets the JOB find the JCL library. The first JOBLIB data set contains the compiler and the second JOBLIB points to the correct runtime libraries. You could place equivalent STEPLIBs in the compile and go steps of the procedures, of course. You could add the procedure library to the JES2 started procedure. (Be careful! If you create an invalid JES2 procedure you will not be able to start the system to correct it. Be certain to have an alternate procedure to start JES2.

To run COBOL programs, you need a runtime STEPLIB pointing to SYS1.SCEERUN (or a JOBLIB that accomplishes the same thing). The LNKLST has another library before SCEERUN that has the same runtime module names; this library is for CICS. If you change the order of LNKLST, to place SCEERUN

[12] This is a JES2 control statement, not “real” JCL. See the *JCL Reference* for details.

earlier, you may have problems with COBOL programs in CICS. Using a STEPLIB for batch COBOL programs is the easiest way around this problem.

The R6 AD system, as originally shipped, has a problem with some of the HFS directories. The problem exists only in the North American version of the CDs. (If your IPL volume is OS39R6, you have the North American version.) In particular, if you attempt to use /usr/lpp/tcpip/lib/nls/msg/C or any lower-level directory, you will encounter 0F4 ABENDS. A diskette is available to replace the corrupted portion of the file system. If you are a user of the AD system and plan to use OE, be certain to request this update (via your Partners-in-Development interface).

2.18 TCP/IP

{ZB70} TCP/IP implementation is discussed at length in Chapter 13, "TCP/IP" on page 223; this section briefly addresses minor topics for the V2R6 AD system.

By default, both an NFS server and NFS client are started at IPL. You can alter this by editing SYS1.ADCD06.PARMLIB(VTAMAPPL).

The TCPIP.PROFILE.TCPIP and TCPIP.TCPIP.DATA data sets are enqueued when TCP/IP is active. You cannot edit these unless you stop TCP/IP. In addition to stopping TCPIP, you may need to cancel INETDx (see the output from a D A, command). You will probably need to cancel NFSC. Notice that NFSC does not appear in a D A,L listing, but it will appear in a D A,ALL listing.

When NFSC ends, it leaves a WTOR on the operator console. Replying "R" appears to restart it.

In summary, to edit PROFILE or DATA, you need to:

```
D A,L           (check if an INETDx address space is present)
P TCPIP
C INTEDx        (if needed)
C NFSC

   edit PROFILE and/or DATA

S TCPIP
S NFSS
r nn,R          (reply R to NFSC message; or S NFSC)
```

The commands to start NFSC and NFSS (the network file system client and server) appear to be unnecessary. They are both subsystems and are apparently started via other initialization.

2.19 Migration Planning for Next CD System

{ZB80} The following guidelines will make it easier for you to replace this level of system software with new levels built the same way, using the migration jobs provided with the AD CD-ROM systems. (Additional details are provided in SYS1.README(PRODDOC)):

- Only IBM-supplied system software should reside on OS39R6, OS3R6A and OS39D6. These volumes are intended to be replaced. You should try to use alternatives to updating these volumes when possible.

All changes to these volumes will be lost when they are replaced, so you should keep a record of any changes made to them. And back-up your system on a regular scheduled basis.

- When you get new levels of the system software, you may choose whether or not to replace SCPMV5. We expect that most people will choose not to replace this volume. Sample jobs are provided for updating SCPMV5 to reflect the new system in dataset SYS1.MIGRATE.R6 on OS39R6. This migration process is described in MIGRATE.DOC on this CD-ROM and in SYS1.MIGRATE.R6. **Always back up your systems before starting any migration jobs.**
- You will also be given the option of replacing P390DX when you receive new levels of system software. A minimum of customization data should reside on this volume. Keeping CICS, IMS, DB2 and QMF customization data elsewhere will make it easier to rebuild this volume after replacement.

Chapter 3. System Planning

{ZC00} There may be a considerable step from an IPLable OS/390 system to one that is ready for useful work. One requirement is to ensure that your work (data, customization, administration) can be carried over to new releases of OS/390. New releases of OS/390 appear every six months and new versions of the AD CD-ROM are available somewhat later. Some installations will install every new OS/390 release -- especially software vendors needing to test their products with every OS/390 release. However, many installations will not install every new OS/390 release; in some cases, one new release per year is considered the best compromise between a current system and the hassle of installing a new release.

3.1 New Release Considerations

{ZC10} 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 SCPMV5 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.[13]
- 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. Suggestions for doing this are given in the next several sections.

3.2 Planning Additional Disk Volumes

{ZC20} You will probably need to add several emulated disk volumes to your P/390 system. There are several reasons for doing this:

- You should use dedicated scratch volume(s). The distributed AD system uses SCPMV5 for scratch space. This is not good because (a) there are too many important data sets on this volume, and (b) there is only a limited amount of space remaining on the volume.
- 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

[13] 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.

system will place new data sets you create on SCPMV5. 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 quite limited. We strongly recommend allocating a separate volume for this -- the volume can be a smaller, nonstandard size.
- 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 Open Edition, you will certainly want additional hierarchical file systems. HFS file systems (which appear as a single PDSE data set to OS/390) *must* be placed on SMS-managed volumes.
- Unless you have experience working with SMS-managed volumes, we strongly recommend that you limit your SMS usage to the HFS volumes.

As a starting point, you might consider adding the following volumes to your OS/390 configuration:

```
volser  unit  cyls   usage...........................

USER01  3380  885    TSO and user files
PAGE01  3380  400    Additional paging  (see comments below)
HFS001  3380  500    Hiercharcial files systems for Open Edition
SPOOL1  3380  500    Larger JES2 spool and checkpoint data sets
WORK01  3380  500    Scratch space (storage volume)
WORK02  3380  500    Scratch space (storage volume)
RACCAT  3380  200    Local user catalog and RACF database
```

The volsers and sizes listed here are arbitrary. You can use either 3380 or 3390 device types[14] It is good practice to select volser names that reflect the usage of the volume, and the names shown here are reasonable. A 3380 with 500 cylinders holds about 350 MB, and this may be a reasonable amount of storage for the purposes indicated. Some of the rationale behind this list is explained in the following paragraphs.

The suggested USER01 volume has a standard 3380 size, 885 cylinders, because some user-supplied programs *might* depend on standard sizes. It is intended as a primary storage volume for your local files.

P/390 systems are not efficient paging systems, and are usually operated with very low paging rates. The nature of the CKD emulation design is such that random I/O with small blocks (such as for paging) is a worst-case situation. Nevertheless, some applications may require large amounts of virtual storage, with data paged into this storage, and will require more paging data sets than are provided on the SCPMV5 volume. If this is your case, you may want to establish an additional paging volume as shown in the list above.

The suggested volume named RACCAT is intended to contain your RACF database and your primary local user catalog. These do not require a separate volume; they could be placed on USER01, for example. If performance is not an issue, a good location might be the suggested SPOOL1 volume or the PAGE01

[14] There may be a small theoretical performance advantage to using emulated 3380 volumes, but we have been unable to measure the differences.

volume (if used).[15] However, your user catalog and your RACF database are likely to be very important items for your extended use of OS/390 and are critical items that need to be carried forward to future releases of the OS/390 CD-ROMs. Using a dedicated volume helps prevent accidents, and isolates two data sets that are critical for migration.

If you have enough information to do so, please plan your own list of additional volumes you will need. If you do not have enough information, you might use the above list as a starting point. If you have a mixture of RAID and non-RAID disks in your OS/2 system, you might take this into account and place appropriate volumes on the non-RAID drives. These would include scratch volumes, possibly the new spool and paging volumes, and the base system volumes.[16] Reasonable planning is required. The P/390 can manage a maximum of 255 devices (as listed in the P/390 Configurator function), including locally-defined terminals (local 3270s) and disks. A reasonable system should not approach this limit.

Step-by-step instructions for adding additional emulated disk volumes are given in 6.1, "How to Add Additional Disk Volumes" on page 64.

If you can reasonably do so, we suggest you allocate all your additional volumes at the same time. Of course, you can create additional volumes at a later time. There is a certain amount of hassle involved in creating and initializing volumes, and it is generally easier to do it all at once.

3.3 Managing Your Own Migration

{ZC30} Moving to a new OS/390 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:

1. 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 the USER01 volume (as described previously).
2. 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 entires in the master catalog for any locally-defined data sets. Defining a user catalog is described in 6.3, "How to Inspect the Master Catalog" on page 67 and the use of ALIAS pointers is described in 5.2, "How

[15] Performance tuning of a P/390 system, as related to emulated disk usage, is very different than a "real" mainframe, and the separation of spooling, paging, catalog, and RACF volumes is not as important. Nevertheless, our suggested list of additional volumes follows principles that would be used on a mainframe system.

[16] It may seem odd to consider the OS/390 IPL volume as "non critical." If you have followed our suggestions, you have not customized or changed the system volumes; you could restore it at any time from CD-ROM.

to Add New Users" on page 48. **Establishing a user catalog and ALIAS definitions should be done before defining any local TSO userids;** otherwise, these new userids will already generate entries in the master catalog and you will not be able to define an ALIAS for them.

3. Keep a list of the ALIAS names you have added to the master catalog.
4. 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.
5. Establish and use your own RACF database, so that the RACF database on SCPMV5 can be replaced (by the new release) without losing your RACF definitions. The steps for defining and using your own RACF database are given in 7.4, "How to Use a Local RACF Database" on page 94.

If you have established the disciplines just described, then moving to a new OS/390 release is fairly simple:

1. You would delete the old release volumes[17] volumes, probably by using OS/2 DEL commands. (We assume you would take backups first.)[18]
2. You would install your new release from the CD-ROMs and IPL it.
3. You log on as userid P390 and take the following steps:
   a. IMPORT your user catalog. (See 6.5, "How to Migrate Your User Catalog" on page 71.)
   b. Add your list of ALIASs to the new master catalog. (See 6.3, "How to Inspect the Master Catalog" on page 67.)
   c. 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, reIPL 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.
   d. Change the RACF data set name table, so that *your* RACF database will be used at the next IPL. (See 7.4, "How to Use a Local RACF Database" on page 94.)
   e. Update your ACDS, or copy the SCDS from your previous system.
4. 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

[17] Depending on what you have done on the P390DX volume, you might want to rename the volume, give it a new address, and retain it for some time after moving to the new OS/390 release. You could then merge the older changes to the new P390DX volume at your leisure.

[18] If you have sufficient OS/2 disk space, there are many options here. If you have sufficient disk space, you could keep all your old OS/390 volumes. You may need to rename some of them (using OS/2 commands) to avoid conflicts with the new OS/390 CD-ROMs you are preparing to install. If your current IPL volume is OS39R2 and your new system will have OS39R3, there is no conflict. However, your new system will probably have a new SCPMV5 volume, so you will need to rename your existing file if you want to keep it. You cannot have two volumes online to OS/390 with the same volsers, so you will need to "clip" the old SCPMV5 name to something different if you want to have it online with the new release. The same is true for the P390DX volume. Instructions for clipping are listed in 6.7, "How to Change a Volume Serial Name" on page 73. You will also need to juggle the addresses in the P/390 configurator if you want to have your old system volumes and new system volumes online to OS/390 at the same time. We strongly suggest changing the addresses for the old volumes and using the expected addresses for your new volumes -- for example, the new IPL volume would have address A80, and so forth.

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.

3.4 Planning Your User Catalog(s)

{ZC40} The preceding descriptions of additional emulated S/390 volumes you may need and migration considerations involved with new OS/390 releases was a bit lengthy. It can be important for your OS/390 use, and much of it implies one particular step you should take shortly after your P/390 OS/390 system arrives --- and before you add any local TSO or OMVS userids. **You should add a user catalog, on a locally-defined volume, and add ALIAS statements to the master catalog.** Once you have added local userids, you will find it difficult to move all their catalog activity to a user catalog. If you do not have your own user catalog for all locally-defined data sets, you may find that moving to a new release of OS/390 (if you cannot use the *migration jobs*) is difficult.

3.5 Other Planning

{ZC50} In addition to the longer-range planning for new releases and the importance of defining one or more user catalogs shortly after installation, there are a number of other areas that need planning attention. These are not urgent -- you can IPL OS/390 and familiarize yourself with initial usage before considering these areas -- but these areas should be considered before you add many users to your OS/390 system. For your initial system use, we suggest you use the userid P390 (whose initial password is P390). This userid is predefined in the AD CD-ROM systems.

3.5.1.1 Backup/Restore Strategy

Before you begin creating your own files or changing the system configuration (changing system files) you should determine your Backup/Restore strategy. This will guard against both data loss (disk or array failure) as well as user error (incorrectly changing the system configuration resulting in an OS/390 system that will not IPL). A number of options are available and are discussed in 8.28, "How to Backup and Restore OS/390" on page 150.

Backup is quite important, and your plans for doing it may affect how you decide to manage disk and tape usage. Therefore, *planning* your backup and recovery strategy should be done before you get into heavy use of the P/390.

3.6 Data Set Naming Conventions

{ZC45} Basic OS/390 will accept almost any name for disk data sets, provided it is in the right format. (See 6.10, "How to Select Data Set Names" on page 87 for a brief discussion of name formats.) This may be acceptable for a very small, unsecured system, but it can rapidly lead to trouble in a larger installation --- even for a P/390-size system. The core requirement is to plan and control the use of High Level Qualifiers (HLQ). This is the first part of a data set name, such as OGDEN for the data set OGDEN.LIB.CNTL, or SYS1 for many of the OS/390 system data sets.

The HLQ is usually a key element in RACF controls, and can be important in many other situations. In general, you (and your users) should use only planned HLQs. HLQs usually correspond to one of these:

- System-defined names, such as SYS1, TCPIP, DLIB (for the AD CD-ROM systems) and so forth,
- TSO userids,
- Major application names, which are defined as GROUP names to RACF. For example, PAYROLL, ORACLE, and so forth.
- A few other planned names, such as HFS or OMVS as the HLQ for all HFS allocations.

3.7 OS/390 Open Edition Considerations

{ZC47} You need some planning before using OMVS for productive purposes. These plans should include:

1. 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. Unlike AIX (for example), OE does not automatically supply the next sequential UID number when you define a new user.
2. The OE publications recommend defining a separate HFS for each OE user, and using the automount function to mount the appropriate HFS when a user logs onto OMVS.
3. Whether or not you use separate HFSs for each user, you should define at least one volume for HFS and define one or more user HFSs on the volume. You can use the HFSUS1 volume for this, if you are using the AD R6 system; this volume need not be replaced by a new release. However, the HFSUS1 volume provided is quite small.
4. 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*.

It is important to plan for OMVS use early; once your users start working with it, it is difficult to change file systems and controls.

3.8 OS/390 Documentation

{ZC75} Sooner or later you will require access to some part of the large library of technical manuals covering OS/390, its components and associated products. Printed manuals are still available for many topics, but the bulk of the documentation is delivered on CD-ROM. At the time this was written, this was a a set of eight CD-ROMs for OS/390 documentation, and the order number was SK27-6700.

We strongly suggest you order these CD-ROMs and install the viewer on the OS/2 side of your P/390 system. Many of the documents you might need (such as the OS/390 Messages manuals) are on one CD-ROM and this can often be left “resident” in the CD-ROM drive, ready for instant use from an OS/2 window.

The material delivered with the CD-ROMs contains instructions for installing the viewing programs (which are also contained on the CD-ROMs), and there is no reason to repeat these instructions here.

The README.MVS file found in the root directory of CD-ROM 1 of the AD system is useful. It can be accessed using the OS/2 System Editor or any editor capable of handling standard ASCII files. You may wish to print a copy of this file so you can refer to it during the loading and initial IPL of the system. A summary of the information found in this file follows:

- A complete list of products delivered with the system
- An overview of the system structure (OS/390 disk volumes)
- Guidelines for managing the system
- OS/390 build information
- Instructions for installing the system from CD-ROM
- Instructions for loading individual OS/390 volumes from CD-ROM
- Instructions for performing an IPL of OS/390
- Description of userids/passwords provided
- Description of the P/390 Devmap provided
- Details of standalone programs provided (ICKDSF and DFSMSdss)

3.9 How to Use the OS/390 CD-ROM Documentation

{ZC77} The AD system includes the latest edition of the *Online Library Omnibus Edition: MVS Collection* SK2T-0710. This CD-ROM collection (6 CD-ROMs at the time of writing) contains softcopy of all technical manuals for OS/390 and the other IBM program products distributed with the system.

You should ensure you have access to this information. The books and technical manuals in the Collection can be viewed by using any of the IBM licensed products BookManager READ/2 (for OS/2), READ for Windows or READ/DOS. If you do not have access to any of these licensed products the MVS Collection contains special editions of these products that limit you to reading only IBM books and any other books that are specifically ″For use with the Library Readers.″

The BookManager READ/2 special edition can be installed on the P/390 and run under OS/2. Full installation instructions are available in book *Online Library: Getting Started and Managing the Library* GC23-0545. If hard copy is not available, a soft copy will be available after you perform the following summarized installation instructions:

1. On the P/390 system, open an OS/2 window
2. Insert Disc 1 of the MVS collection in the CD-ROM drive
3. Change to the drive letter of the CD-ROM drive (usually E:)

   ```
   e:
   ```

4. Change to the directory on the CD-ROM that contains the library reader install procedure

   ```
   cd \ez2inst
   ```

5. Start the install process

   ```
   install
   ```

6. Follow the on-screen prompts and installation flow to complete the installation. Online help is provided during this process.
7. Restart OS/2 to finalize the installation

BookManager READ/2 is now available. If you did not have access to the book referenced above (*Online Library: Getting Started and Managing the Library* GC23-0545), it is now available in the *Online Library Reference Bookshelf*. This book provides valuable information required to get the most out of the MVS Collection.

You should spend some time becoming familiar with these facilities, becoming proficient with the use of the Library Reader product and understanding what information is contained in the MVS Collection and how to access it. Effective management of OS/390 will depend upon your ability to find the right information when you need it.

3.9.1.1 Using Library Reader and the MVS Collection

The *MVS Collection* CD-ROM set organizes its books into logically related groups of books called bookshelves. There is a bookshelf containing books about OS/390, one for the DFSMS series of products (formerly DFP), one for security, one for networking,and so on. There is also a bookshelf for each CD-ROM Disc containing a list of all books on that Disc.

Of special interest is a bookshelf called Booklet for MVS Collection containing a single book *MVS Collection Booklet*. This bookshelf/book combination can be found on each CD-ROM Disc in the collection. By opening this book and using the READ programs search facility you can determine the Disc containing the required technical document. This search can be conducted using the document title, part of the document title or the document number.

Chapter 4. IPL and Startup

{ZD00} This chapter contains topics relating to IPLing MVS. Key elements are the handling of PARMLIBs and the use of an IODF. See 2.6, "IPLing the AD CD-ROM System" on page 11 for specific information about IPLing the R6 AD system.

4.1 How to Start the P/390 Subsystem

{ZD02} With a P/390, 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 OS/390 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 reIPL. This will cause all the P/390 support programs to end, and IPL P/390 icon will then start fresh copies.

4.2 How to IPL OS/390

{ZD04} Specific information for IPLing the R6 AD system is presented in 2.6, "IPLing the AD CD-ROM System" on page 11

In general, smaller MVS systems (such as might be used for P/390 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. 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.[19]

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. The setup for automatic IPL is done in the LOADxx and COMMNDxx members of the PARMLIBs (and possibly SYS1.IPLPARM).

Versions with an automatic IPL should provide an alternate SYS1.PARMLIB(LOADxx) member that leads to a normal OS/390 IPL, during which you must respond to several messages. You could use this alternate process by:

1. Before IPLing, use the P/390 configurator to go to the Update System Environment panel (F4 in the Configurator options list).

[19] Note that the terms *automatic IPL* and *manual IPL* are not standard terms.

2. Find the line similar to:

```
System Load Parameter    > 0122CS   (for OS/390 R6 CD-ROM)
```

3. Change the 0122CS parameter to 0122xx, where xx is a documented alternate LOADxx number. It is probably 00, so the full parameter would be 012200. The meaning of the System Load Parameter is explained in 4.3, "Understanding IPL Parameters" on page 37.
4. Press ENTER, F10, F6 to exit from the configurator.
5. IPL OS/390 by clicking on the IPL P/390 icon.

The remainder of this section assumes you are performing a normal OS/390 IPL and responding to the normal messages. The first message you normally see is:

```
IEA101A SPECIFY SYSTEM PARAMETERS
```

and you have several options for replying:

1. Be certain your make the 3270 window with this message your active window. You can do this by moving the mouse pointer to the window and clicking *once*. If you double click, the 3270 cursor may move to the mouse pointer location, and this is probably not where you want it. If this happens, press TAB to move the cursor to the input field.
2. Simply press ENTER on the console[20] Or,
3. Enter

   ```
   reply 00,'U'
   ```

   which has exactly the same effect (only for a very few startup messages) as pressing ENTER. Or,
4. Enter

   ```
   reply 00,'CMD=02,MLPA=04,........'    (or whatever is needed)
   ```

   where you are overriding parameters in SYS1.PARMLIB(IEASYS00). This is normally done only when recovering (or testing) error situations.
5. Enter

   ```
   reply 00,CLPA
   ```

   if you have changed any of the LPA data sets and need to "do a CLPA the next time you IPL."

You may see a highlighted (in red) message:

```
ILR005E PLPA PAGE DATA SET FULL, OVERFLOWING TO COMMON
```

This can be ignored.[21]

The next response you may need[22] during a normal IPL is for the message:

```
01 $HASP426 SPECIFY OPTIONS - JES2 OS ---
```

and the normal response to this message is:

```
1,NOREQ                  (you can enter in lower case)
```

[20] This is not always so simple, because the ENTER key may have moved. Most 3270 emulators place the ENTER function with the right-hand Ctrl key. The large key marked Enter becomes a NewLine key. However, some of the 3270 emulator customizations provided with P/390 (and R/390) systems use the PC Enter key as the 3270 ENTER key -- which is logical for PC users but disconcerting for 3270 users.

[21] Certain MVS folklore asserts that paging efficiency may slightly improve if PLPA overflows into COMMON; this may or may not be true. 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.

[22] You can bypass this message by specifying a NOREQ parameter in the COMMND00 member that starts JES2.

in which the "1" indicates that you are replying to message number 1. (If there were unusual interactive messages earlier in the IPL sequence, of if it takes you several tries to enter an acceptable response, the number will be greater than 1. See 11.2, "How to Reply to Console Messages" on page 196 for more information about formats for console replies.) The NOREQ means "no requests," and is the most typical JES2 startup parameter. The only alternate parameter you might want to enter is COLD, which means to cold start JES2 (that is, erase all the old contents of the spool area). If you specify COLD, you will receive messages asking if you really mean to do this and you must reply to these messages. The most common reply is U.[23] After the JES2 cold start reformats the spool area, *you must enter the command "$S" (two characters, no leading number) to tell JES2 to start its normal processing.*

The last IPL response that is often needed is for the message:

```
02 IKT003D TCAS UNABLE TO ACCEPT LOGONS
   REASON CODE=0092  REPLY 'RETRY' or 'TERM'
```

This message means that TSO is trying to start, but cannot find VTAM. 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 message again. Again, the message number may not be 2 if you had more IPL response messages than listed here.

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.

At this point, you can select another 3270 session and log onto TSO.

4.3 Understanding IPL Parameters

{ZD20} OS/390 provides great flexibility in determining the configuration of the resources managed. Each new version of the operating system has added to the controls and methods available to Systems Programmers to achieve this flexibility. The following text provides a brief summary of the components used by the AD system during IPL to achieve this environment.

1. System Load Parm. Specified in the Update System Environment panel of the P/390 Configuration function, the supplied value is:

   ```
   0122CS        (for OS/390 R6; other systems may differ)
   ```

   This indicates the current IODEF VSAM data set is located on device address 0122 (SCPMV5) and to use a LOADxx member named LOADCS. The blank following 0122CS indicates the default is being used for the IMSI (Initialization Message Suppression Indicator) function. The default is to

23 The single character "U" is the most common reply to many OS/390 messages. It generally means to continue with whatever is happening.

suppress many of the informational messages previously observed during an OS/390 IPL.

2. The AD R6 system contains a SYS1.IPLPARM data set, so the LOADCS member referenced above is found there.

```
IODF     00 SYS1     CBIPO    00
NUCLEUS  1
SYSCAT   SCPVM5113COS390.MASTER.CATALOG
SYSPARM  CS
NUCLST   00
IEASYM   00
PARMLIB  SYS1.ADCD06.PARMLIB
```

3. The LOADCS member is processed with the following results:

   - the IODF statement names the current IODF data set as SYS1.IODF00, the OS Configuration ID as CBIPO and the EDT identifier as 00.
   - the NUCLEUS statement indicates member IEANUC01 from SYS1.NUCLEUS is used to build the nucleus.
   - the SYSCAT statement defines the master catalog as an ICF catalog named OS390.MASTER.CATALOG on volume SCPMV5.
   - The SYSPARM statement specifies CS. This means member IEASYSCS (in PARMLIB) is used as the starting point for PARMLIB processing.
   - SYS1.ADCD06.PARMLIB will be concatenated before SYS1.PARMLIB. (Such PARMLIB concatenations are a critical part of the LOADxx member, and can greatly alter the startup parameters of your system.)

4. The master catalog (OS390.MASTER.CATALOG) has catalog entries for SYS1.ADCD06.PARMLIB and SYS1.PARMLIB; these are searched (in order) for IEASYSCS.

5. IEASYSCS is then used to obtain the remaining system parameters including references to many other SYS1.PARMLIB members containing configuration information for components and sub-systems of OS/390. See 2.10, "IPL Parameters and Parmlibs" on page 14 for a list of the PARMLIB members used.

Much greater detail can be found in the following manuals (available on Disc 1 of the OS/390 Collection):

OS390 Initialization and Tuning Guide
OS390 Initialization and Tuning Reference
OS390 System Commands

You should also review 4.5, "How to Work With PARMLIB" for more details on how SYS1.PARMLIB is used in OS/390 and how to safely manage change to this critical data set.

4.4 How to Specify an IODF

{ZD30} The MVS/ESA 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). In a standard S/390, 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[24] A P/390 (or R/390) does not use IOCDS functions; consequently the Dynamic Reconfiguration function provided through IODFs is not available for these systems. You must reIPL a P/390 (or R/390) 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 IOCDSs. With current versions, configurations are changed with the HCD function, which produces IODF files. An IODF file contains, among other things, the information needed to build MVS UCBs. HCD can also produce IOCDS files, but these are not used with P/390s or R/390s. 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: a *production* IODF and a *work* IODF. You can have many production IODF data sets; one of these 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 P/390 OS/390 environment. The IODF supplied with the MVS 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 only have a production IODF and you need to modify it, you must do the following :
  1. Create a work IODF from the production IODF.
  2. Alter it as needed, using HCD functions.
  3. Create a new production IODF from the work IODF.
  4. IPL using the new production IODF.[25]
- 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 must create a new production IODF for every change.
- You can write a new production IODF over an existing production IODF, even the *active* 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.

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 use the standard naming convention: if your production IODF (that you are viewing with HCD option 1) is SYS1.IODFnn, you should name the work IODF created from it SYS1.IODFnn.WORK.[26] *Place all IODF data sets on the same volume.*

---

[24] The IOCDS is a file used by the S/390 “hardware” to define and manage channel, control unit, and device mapping.

[25] There is also an option to dynamically switch to the new IODF; this does not work with P/390 and OS/390.

[26] You might select a new nn value such that it will match the new production IODF you intend to create. For example, if you start with production SYS1.IODF00 and try to modify it, HCD will ask you for a name for a new work IODF. It would be logical to select SYS1.IODF01.WORK. You should then modify SYS1.IODF01.WORK to add new tapes, and then create a new production IODF file. You could name this SYS1.IODF01. However, the most common convention appears to be IODF00 ---> create work ---> IODF00.WORK ---> create new production ---> IODF01, and so forth.

Once a production IODF is produced, it can be used. Here are some of the key points about using an IODF:

- An IPL parameter is the starting point. The IPL parameter 012200 is used with the AD CD-ROM system. This value is entered in the appropriate field of the P/390 environment configuration panel (which is obtained by pressing PF4 while in the main menu of the P/390 configuration program). Do not confuse the *IPL address* (sometimes called the *IPL volume address* or *SYSRES address*) with the *IPL parameter*. The IPL address is the channel address of a disk drive that contains the IPL program. The IPL parameter is a data string that is placed in registers before starting the IPL process.

  Our IPL parameter tells the MVS IPL program to search the disk volume at address 0122 for SYSn.IPLPARM or SYS1.PARMLIB.[27] Our system used SYS1.PARMLIB, on the volume at address 122.
- SYS1.IPLPARM (or SYS1.PARMLIB, etc) is a PDS. Once found, the IPL program opens member LOADnn, where nn is the fifth and sixth digits of the IPL parameter (and is 00 in the AD CD-ROM system).
- In our case, SYS1.IPLPARM(LOADCS) (on the 3380 volume at address 0122) contains a line like this:

  ```
  IODF     00 SYS1     CBIPO      00
  ```

  IODF starts in column 1. The 01 begins in column 10, the SYS1 begins in column 13, and the CBIPO begins in column 22. These fields are sensitive to particular columns. Two of these fields are used to specify the name of the production IODF data set to be used during OS/390 operation. This data set must be on the same volume as the SYS1.PARMLIB data set (or the SYSn.IPLPARM data set, if it is used instead of SYS1.PARMLIB). (It appears that the CBIPO parameter can be omitted, if there is only one choice.)

  In this example, the data set name will be SYS1.IODF00. The SYS1 portion of the name is taken from the field beginning in column 13 of the IODF statement. The IODF portion of the name is fixed. The 00 portion of the name is from column 10 of the IODF statement. A given IODF data set can contain multiple MVS definitions. In our example, the definition CBIPO (from column 22) is used for the OS/390 being started. The 00 parameter in the IODF statement is the number of an EDT table in the IODF definition.[28]

You can verify your currently active IODF name with the OS/390 command:

```
D IOS,CONFIG(ALL).
```

This will produce (on the OS/390 console) a response like this:

```
ACTIVE IODF DATASET = SYS1.IODF00
CONFIGURATION ID = CBIPO   EDT = 00
HARDWARE SYSTEM AREA DATA COULD NOT BE OBTAINED
```

If you need to change your OS/390 configuration you should proceed in a way that provides a fallback if your new IODF is not usable. That is, you want to leave your old, working IODF intact and usable. There are a number of ways to do this. We created SYS1.IODF01 for our first new IODF and added member

[27] The system will search first for SYSn.IPLPARM (where n is 0 to 9) on the volume indicated by the IPL parameter, then it will search for SYS1.PARMLIB on this volume, then it will search for SYS1.PARMLIB on the IPL volume. It expects to find member LOADxx in the first of these data sets it finds.

[28] The EDT is the eligible device table used to resolve esoteric JCL device names.

LOAD01 to the SYS1.PARMLIB data set. (Remember to keep all your IODF data sets on the same volume.)

Our initial AD CD-ROM contained SYS1.IPLPARM(LOADCS):

```
IODF     00 SYS1     CBIPO    00
NUCLEUS  1
SYSCAT   SCPVM5113CMVSV5.MASTER.CATALOG
SYSPARM  CS
NUCLST   00
IEASYM   00
PARMLIB  SYS1.ADCD06.PARMLIB
```

We created member SYS1.IPLPARM(LOAD01) like this:

```
IODF     01 SYS1     CBIPO    00
NUCLEUS  1
SYSCAT   SCPVM5113COS390.MASTER.CATALOG
SYSPARM  CS
NUCLST   00
IEASYM   00
PARMLIB  SYS1.ADCD06.PARMLIB
```

The new member is exactly the same as the original member except for the two digits (column 10 in the IODF statement) used as the suffix for the IODF data set name. The new member will use SYS1.IODF01 as the production IODF. By changing the IPL parameter from 012200 to 012201 we can IPL with our new IODF. If this fails for some reason, we can IPL with parameter 012200 and use the original IODF.

Note that this example of LOADCS specifies an alternate PARMLIB that is concatenated before SYS1.PARMLIB. The AD system uses this, although your system may not.

4.5 How to Work With PARMLIB

The partitioned data set SYS1.PARMLIB (PARaMeter LIBrary) is the primary source of information used when configuring and customizing OS/390 and other software products that execute in the OS/390 environment. It 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 configuring process can be at IPL time, when a component or program is started, or may be driven by an operator command from an OS/390 console.

This has been complicated by the introduction of multiple PARMLIBs. These are libraries that are specified in the LOADxx members (see 4.3, "Understanding IPL Parameters" on page 37) and that are concatenated before SYS1.PARMLIB. 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 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 8.55, "How to Use ISRDDN" on page 171 for one technique for working with concatenated PARMLIBs.

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. Unless you override the system (or unless your LOADxx member pointed to by the IPL parameter overrides it), IEASYS00 is used. An override would use IEASYSxx (a suffix other than "00," that you select.) In our examples here, we assume you use IEASYS00. IEASYS00 contains two types of parameters:

- Controls for some part of the OS/390 configuration
- References to other members of SYS1.PARMLIB

See Figure 1 on page 43 for the contents of IEASYS00 as supplied in SYS1.PARMLIB with the AD system.

The names of members of SYS1.PARMLIB have a leading fixed part followed by an installation-defined part of 2 characters. The fixed section often indicates the function of the member, for example:

```
IEASYS    The control member
IEAFIX    List of modules fixed in storage
CONSOL    Definition for the OS/390 Consoles
LNKLST    List of libraries used for OS/390 standard program fetch
LOAD      Parameters used during early OS/390 IPL
```

The installation defined suffix can be any two alphanumeric characters. The AD uses two zeros for all supported SYS1.PARMLIB members; for example, the members referenced above are named:

```
IEASYS00   The control member
IEAFIX00   List of modules fixed in storage
CONSOL00   Definition for the OS/390 Consoles
LNKLST00   List of libraries used for OS/390 standard program fetch
LOAD00     Parameters used during early OS/390 IPL
```

The supplied SYS1.PARMLIB also contains many versions of the various members (that is, they have suffixes such as 01, 99, MI, AB, etc). These are not used during normal operation but illustrate different configurations that could be specified. Which version of a SYS1.PARMLIB member used during IPL or component startup can be determined several ways:

- A corresponding entry in IEASYSxx. For example the line:

  ```
  LNK=01,
  ```

  will cause member LNKLST01 to be used to determine the system linklist contents. (Note the *comma*. All the lines in IEASYS00, except the last data line, should have commas.)

- The system operator can reply at the master console to message IEA101I. For example,

  ```
  R 00,LNK=01
  ```

  to message IEA101I will cause LNKLST01 to be used.

- 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 fuctions (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 IEA101I will overide a specification in the controlling member IEASYSxx. This principle is used to provide fallback when making changes to SYS1.PARMLIB.

```
CLOCK=00,                    ===> PLEASE READ NOTES AT END
CMB=(UNITR,COMM,GRAPH,CHRDR), ===> TO MAKING CHANGES
CMD=00,                      COMMND00
CON=00,                      CONSOL00 (DEFAULT)
CSA=(3000,8000),             MVS/ESA CSA RANGE
DUMP=DASD,                   PLACE SVC DUMPS ON DASD
FIX=00,                      IEAFIX00
GRS=NONE,                    NO INTERSYSTEM SERIALIZATION
ICS=00,                      IEAICS00 (DEFAULT)
IPS=00,                      IEAIPS00 (DEFAULT)
LNK=00,                      LNKLST00 (DEFAULT)
LNKAUTH=LNKLST,              APF-AUTHORIZE LNKLSTXX LIBS
LOGCLS=L,                    WILL NOT BE PRINTED BY DEFAULT
LOGLMT=999999,               MAX WTL MSGS QUEUED (6 DIGITS)
LOGREC=SYS1.LOGREC,          LOGREC DATASET NAME
LPA=00,                      LPALST00
MAXUSER=250,                 MAX ADDR SPACES (STC+INITS+TSO)
MLPA=00,                     IEALPA00
MSTRJCL=00,                  MSTJCL00 (DEFAULT)
PAGTOTL=(5,2),               CAN ADD 3 PAGE, 2 SWAP D/S
PAK=00,                      IEAPAK00
OPI=YES,                     OPERATOR MAY OVERRIDE  PARMS
OPT=00,                      IEAOPT00
PAGE=(SYS1.PLPA.PAGE,        PAGE
      SYS1.COMMON.PAGE,          DATA SET
      SYS1.LOCAL1.PAGE,                   NAMES
        L),                  LIST PAGE DATA SET NAMES AT IPL
PROG=00,                     PROG00 - DYNAMIC APF LIST
RSU=0,                       NO RECONFIG STORAGE UNITS
RSVSTRT=5,                   RESERVED ASVT ENTRIES
RSVNONR=5,                   RESERVED ASVT ENTRIES
SCH=00,                      SCHED00 (DEFAULT)
SMF=00,                      SMFPRM00 (DEFAULT)
SQA=(8,8),                   MVS/ESA SQA APPROX 1024K
SSN=00,                      IEFSSN00
SVC=00,                      IEASVC00 (DEFAULT)
SYSNAME=P390,                SAME AS SID IN SMFPRM00
VAL=00,                      VATLST00 (DEFAULT)
VIODSN=SYS1.STGINDEX,        VIO JOURNALING DATA SET
VRREGN=64                    DEFAULT REAL REGION SIZE
```

Figure 1. SYS1.PARMLIB Member IEASYS00 Supplied with the AD System. Comments at the bottom have been removed.

Making changes to SYS1.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. In this case, you are not able to use TSO to correct your mistake. You can minimize this danger by implementing the change in a controlled manner with a guaranteed back out (or fallback). Most installations achieve this by controlled usage of the suffix part of the SYS1.PARMLIB member names. A simple, but effective technique is to use the following naming standard:

```
PARMLIB member names ending with 00 - Member used for normal operation
                                      and when installing a change.
PARMLIB member names ending with 01 - Backout version, that is the
                                      unchanged version of the member
                                      when installing a change.
```

The principle is that the 01 version of a member is a "good" copy and will always function correctly. Note that the 01 version does not need to be the latest good version of the member; it simply must be a version that works. **You should always maintain a version of each modified PARMLIB member that will guarantee a successful IPL.**

The steps involved, using member LNKLST as an example, are:

1. Copy the present version of the member (LNKLST00) to the backout version (LNKLST01). At this point both versions are the same.
2. Make the changes to the normal version (LNKLST00). In this case add a library name to the link list.
3. Test your change by IPLing the system, thereby invoking the change (using LNKLST00).
4. If the test is successful the change is implemented. The backout version of the member (LNKLST01) contains a record of the previous system configuration.
5. If the test fails for any reason, the change can be backed out by restarting the system and using the backout version (LNKLST01) rather than the normal operation version (LNKLST00). You can then correct the problem in the normal operation member (LNKLST00) and retry the test or completely remove the change by copying the backout version (LNKLST01) over the normal operation member (LNKLST00).
6. To restart the system using LNKLST01, you would reply to message IEA101I[29] during IPL:

   ```
   R 00,LNK=01
   ```

   The format of the reply to message IEA101I will depend on which member or members you wish to overide from the normal operation flow. The *OS/390 Initialization and Tuning Reference* provides details for the syntax required for IEASYSxx as well as the format of the replies to message IEA101I.

There are many variations possible for the names and content of the SYS1.PARMLIB members. While the method outlined above provides a simple method of changing SYS1.PARMLIB with full backout, you can use more complex naming standards and structures to achieve your goals. Just remember that if you get it wrong recovery can be difficult.

Every MVS systems administrator has, at one time or another, made his system unworkable due to an error in SYS1.PARMLIB. Typical recovery approaches are:

- IPL a different system and use ISPF from that system to edit and correct your SYS1.PARMLIB. This is practical if you have enough disk space to keep a spare OS/390 system. This recovery system is often referenced as a *one-pack* system, meaning that it is small enough to fit on a single, smaller disk volume.
- Use standalone DFDSS to restore the volume containing SYS1.PARMLIB. This assumes you have a backup of that volume available and that performing the restore will not introduce other problems such as regressing catalog entries (the master catalog is on the same volume in the AD system).

[29] This assumes you are performing a "manual IPL" in which you have an opportunity to respond to this message. You may need to use a different IPLPARM (and different LOADxx member) to do this.

- You may wish to keep a copy of this critical volume (SCPMV5) on your OS/2 system (assuming you have enough disk space). Then you can recover your system by using the P/390 configurator to point to the old backup version of SCPMV5.
- If all else fails you will have to reload the system from CD-ROM and start again. With this scenario you may be able to rename or backup your production OS/390 volumes using OS/2 software and later use the reloaded system as a driver to correct your production system volumes. This will depend upon the availability of sufficient disk, time and patience.

All members of SYS1.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 IEA101I:

```
R 00,SYSP=xx
```

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, the parameter itself can be specified in reply to message IEA101I.
- IEASYSxx, the parameter will be used if found in IEASYSxx when specified by operator reply, that is R 00,SYSP=xx.
- IEASYS00, the parameter has not been specified by the operator explicitly or in an alternative IEASYSxx member.
- System defaults are used if the parameter is not explicitly specified using any of the above methods.

4.5.1 Commonly Modified PARMLIB Members

The following are the most common members that you are likely to change and the reason you would need to change them:

Table 2 (Page 1 of 2). Commonly modified PARMLIB members

COMMNDxx	MVS operator commands executed at IPL time. You can add your own commands, for example automatically start your databases or submit a job to perform disk space cleanup. You may also wish to remove some of the functions provided with the system such as RMF or the Open Edition address spaces (APPC, ASCH and OMVS).
CONSOLxx	This member contains the definitions for the OS/390 system consoles. You are not likely to change this except for adding sub-system consoles (consoles available from other address spaces such as TSO or Netview).
IEAICSxx	Definitions for classifying work in the OS/390 system. You would change this member if you wish to tune your OS/390 system.
IEAIPSxx	Definitions for performance management in the OS/390 System. You would change this member if you wish to tune your OS/390 system.
IEASVCxx	Definitions for user SuperVisor Calls (SVCs). Installing database software and some non-IBM software products may require user SVC definitions.
IEASYSxx	The control member for SYS1.PARMLIB use during IPL. This chapter and the Initialization and Tuning manuals provide information about this member and the reasons that may require you to change it.
IEFSSNxx	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.

Table 2 (Page 2 of 2). Commonly modified PARMLIB members	
IGDDFPKG	Control member for DFSMS product usage. Change instructions are provided with the DFSMS documentation. Note this member does not use suffixing.
IKJTSOxx	Control member for TSO/E. You will generally not need to change this member, but you may have to increase the USERMAX parameter if TSO users are prevented from logging onto the system because of the ′maximum users logged on, try later′ message.
LNKLSTxx	This member 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 library is a very common occurrence in most installations.
LOADxx	Parameters used very early in the IPL process. You may need to change this to use a different IODF, master catalog or nucleus during IPL. Several other parameters are available to change the way the IPL sequence occurs.
LPALSTxx	This member provides a list of libraries whose members are used when constructing the OS/390 link pack area. Some products will include optional steps for adding libraries to this member. This should be considered in the P/390 environment because of the performance benefits that apply to programs included in the link pack area.
MPFLSTxx	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.
MSTJCLxx	This member contains the JCL used by the OS/390 master address space. Generally you will not need to change this member.
PROGxx	This member contains a list of libraries that are ′authorized′ under OS/390 to use restricted facilities. Many IBM and non-IBM products will require you to add libraries to this member. Auditors may question the need for you to have libraries in the list.
SCHEDxx	This member 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 bypass data set integrity).
SMFPRMxx	The System Management Facilities (SMF) configuration is controlled by this member. You may wish to change this member particularly to start disk recording, to add data sets for recording and to change the system wait time (to prevent TSO users from abending with S522 abends).
TSOKEYxx	This member controls various special facilities within the TSO environment. Some program products may require TSO commands to be added to the tables in TSOKEYxx.
VATLSTxx	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.

All the above members, plus many other more specific members, are delivered in SYS1.PARMLIB. All are initially suffixed with 00. The following technical manuals, available on the *MVS Collection* CD-ROMs, provide a great detail of technical information about the available members of SYS1.PARMLIB, what they contain and how and when they are used:

MVS/ESA SP V5 Initialization and Tuning Guide
MVS/ESA SP V5 Initialization and Tuning Reference
MVS/ESA SP V5 System Commands

Chapter 5. User Administration

{ZE00} The AD CD-ROM system is already provided with these users:

Table 3. Userids provided with AD CD-ROM system

Userid	Password	Comments
IBMUSER	SYS1	Full security access (RACF SPECIAL); very limited logon procedure; intended for an “emergency” userid only. Uses SYS1.UADS.
P390	P390	Full security access (RACF SPECIAL); member of SYS1 group; has ALTER access to SYS1 data sets. NO RACF TSO segment, uses SYS1.UADS.
P390A	TEST	Has SPECIAL and OPERATIONS.
P390B thru P390Z	TEST	Normal TSO userids
TESTER	TEST	Normal TSO userid
OPEN1 thru OPEN3	SYS1	Open Edition user with UID=0

We suggest the following:

1. Leave the IBMUSER userid unchanged. Do not use it, except perhaps to try it. It may be usable when no other userid works because it uses a much more limited logon procedure and can operate without RACF. This userid has full RACF (and SYS1.UADS) authority and could be a security exposure. You should probably change the password (in both RACF and SYS1.UADS).[30]
2. Use the P390 userid for routine administration. It has RACF SPECIAL authority (and SYS1.UADS ACCOUNT authority) and is completely suitable for OS/390 administration. P390 also has OMVS access and has UID=0. You should change the password from P390 to a local password as soon as possible.
3. Ignore the P390A - P390Z and TESTER userids unless you are in an educational environment where you need predefined userids for students. You might use these userids as targets for practicing deleting users. If you do not delete them, you might want to change their passwords (one at a time) if you are concerned about logon security.
4. The OPEN1 - OPEN3 userids have no special RACF authority, but they are all defined with UID=0 and are dangerous from an OMVS viewpoint. The P390 userid also has UID=0, so the OPEN1 - OPEN3 userids are not needed for OMVS administration. You might want to delete these userids, or at least change their passwords as soon as possible.

The TSO logon procedure included with the AD CD-ROM system is quite large, and is consequently rather slow. The primary ISPF menu included with the system contains options for SMP/E, ISMF, RACF, HCD, IPCS, DITTO, RMF, and DFSORT. These functions are typically used by few users, and the logon procedure will be much faster if the data sets for these functions are not allocated by the TSO logon procedure. A suggestion for defining new logon procedures are given in 8.9, “How to Make TSO Logon Faster” on page 118.

30 You may also want to violate normal security recommendations and write down the new password. In normal usage, IMBUSER is very seldom used and you may not remember the password when you need it.

Removing these allocations from the TSO logon procedure has an obvious consequence -- these associated ISPF menu items will fail. This can be overcome in several ways. The two common approaches are:

1. Define a CLIST script that allocates the data sets for SMP/E, ISMF, and so forth. In this case, the TSO logon procedure would be shortened for all users, and anyone needing to use one of the advanced ISPF applications would need to execute the CLIST first. This is a convenient solution, and is described in 8.26, "How to Set Up an ISPF Administrative Environment" on page 144.
2. Define two TSO logon procedures: a shortened version, and the full version. This is less convenient because a typical user would need to log on again if he unexpectedly needed an advanced application.

5.1 How to Start Organizing User and RACF Administration

{ZE05} 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:

1. Use a batch job to define new users,
2. Use TSO RACF commands to alter the RACF profiles of existing users,
3. Use RACF ISPF panels to work with RACF general resource profiles and other more advanced RACF functions.
4. 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.[31]

5.2 How to Add New Users

{ZE15} 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, MARTIN is the new userid, and MARTIN will have access to TSO and OMVS.

[31] 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.

```
//P390T    JOB  1,P390,MSGCLASS=X
//RACF     EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//SYSUADS  DD   DSN=SYS1.UADS,DISP=SHR
//SYSLBC   DD   DSN=SYS1.BRODCAST,DISP=SHR
//SYSTSIN  DD   *
  AU MARTIN  NAME('MARTIN CERON H') PASSWORD(MERL1N)          +
    OWNER(SYS1) DFLTGRP(SYS1) UACC(READ)                      +
    TSO(ACCTNUM(ACCT#) PROC(ISPFPROC) JOBCLASS(A) MSGCLASS(X)  +
      HOLDCLASS(X) SYSOUTCLASS(X) SIZE(4048) MAXSIZE(0))      +
    OMVS(HOME(/u/martin)      PROGRAM(/bin/sh)  UID(201))

  CONNECT MARTIN GROUP(SYS1)
  PERMIT ACCT#    CLASS(ACCTNUM) ID(MARTIN)
  PERMIT ISPFPROC CLASS(TSOPROC) ID(MARTIN)
  PERMIT JCL      CLASS(TSOAUTH) ID(MARTIN)
  PERMIT OPER     CLASS(TSOAUTH) ID(MARTIN)
  AD 'MARTIN.*'     OWNER(MARTIN) UACC(READ) GENERIC
  SETROPTS REFRESH RACLIST(TSOPROC ACCTNUM TSOAUTH)
  LU MARTIN
  LD DA('MARTIN.*') ALL
/*
//*************************************************************//
//* Add an ALIAS to the user catalog  (Not used with the AD    *//
//* system unless you added a user catalog!)                   *//
//*************************************************************//
//ALIAS    EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 DEFINE ALIAS (NAME(MARTIN) RELATE(CATALOG.RACCAT))
//*************************************************************//
//* Create an ISPF profile data set. (Not needed with AD, but  *//
//* this allows you to select volser for ISPF profiles.)       *//
//*************************************************************//
//ISPF     EXEC PGM=IEFBR14
//DD1      DD   DSN=MARTIN.ISPF.ISPPROF,DISP=(,CATLG),
//    VOL=SER=WORK01,UNIT=3380,DCB=(LRECL=80,RECFM=FB,BLKSIZE=0),
//    SPACE=(TRK,(3,3,10))
//*************************************************************//
//* Remember to update the ALIAS job (if you have one)         *//
//*************************************************************//
```

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 CD-ROM 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, but (in the AD system) has a very limited logon procedure that is not appropriate for routine use. 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 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.

- Ten RACF commands are executed: AU (ADDUSER), CONNECT, several PERMITS, AD (ADDSD or add dataset 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, 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:
  1. The userid (which is the operand just after the AU command),
  2. The NAME field (which can be listed in RACF reports, but which has no direct processing function, should reflect the user′s name),
  3. The UID value (in the OMVS segment),
  4. The HOME directory (in the OMVS segment),
  5. The userid in the CONNECT command,
  6. The userids in the PERMIT commands,
  7. The userid in the AD command,
  8. The userids in the LD and LU commands,
  9. The NAME field in the DEFINE ALIAS command,
  10. The HLQ in the DSN parameter in the ISPF job step (which should be set to the new user′s userid).

Note that everything except the UID value and NAME field could be changed by a single global *change* command using the ISPF editor.

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, XRFSOFF, 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 P/390 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 CONNECT command connects this user to an existing group named SYS1. The SYS1 group exists in practically all RACF systems (including the AD system) and is usually considered the base group to which all users belong.
- The UACC value of READ (in the AD statement) will permit any other user or job to read Martin′s data sets. You could set the UACC value to NONE, 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 ISPFPROC statements allow this user access to the specified account number (ACCT#) and the standard logon procedure (ISPFPROC). See 7.25, "How to Use TSO Authorities Profiles" on page 107 for more discussion about the TSOAUTH and similar profiles.
- The PERMIT JCL and OPER commands are used primarily to control SDSF security, and this is addressed in 8.11, "Understanding Basic SDSF Security" on page 121. You may want to remove the PERMIT OPER command.
- After a new user is added, using this job, you can use the RACF command ALTUSER 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 TSO command ALTUSER 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 ISPFPROC in the PROC parameter is the name of the full-function TSO logon procedure supplied with the AD CD-ROM system. If you create a smaller logon procedure for normal users (as recommended in 8.9, "How to Make TSO Logon Faster" on page 118), you would use the name of your new procedure in the PROC field.
- The example here does not contain RACF commands to CONNECT the new user to RACF groups. In the example, new users are part of the SYS1 group, but are not connected to any other groups. We cannot provide a reasonable example for CONNECTS, since this depends completely on your group environment.[32]
- 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 MARTIN.** (with two asterisks). By default, EGN is not enabled in the AD CD-ROM systems.

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

[32] If you are not interested in creating RACF groups, you can simply leave users as part of the SYS1 group and never create any additional groups. Group definitions are quite important for larger installations, but a small development organization using a P/390 might ignore the whole issue.

5.3 How to Modify a RACF User Profile

{ZE20} A most 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 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 NOOMVS
ALU ogden TSO(ACCTNUM(ax) PROC(ispfproc) UNIT(SYSDA))
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 P/390 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.

5.4 How to Delete Users

{ZE25} 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 usually 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 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[33] will not take the time to fully examine and dispose of all the data sets 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 it 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:

```
ALTUSER 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 this command. The REVOKE function does not delete any RACF entries or user data sets; it merely prevents the user from using the system.

[33] This means you, the system owner/administrator, or your advisors, or a department manager, or anyone else involved.

The second step is to identify the user's data sets; in the most normal case, this means to identify all data sets with the HLQ equal to the userid being deleted. There are two cases here:

1. 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.
2. Data sets that are not cataloged. There is no easy way to find these. The normal 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. (Note that data sets on SMS-managed volumes must be cataloged.)

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 relatively new RACF tool, IRRRID00, can be used to locate all uses of a given userid in RACF.[34] 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=3380,SPACE=(CYL,(10)),DCB=(LRECL=4096,RECFM=VB),
```

34 This tool can also be used for other clean-up purposes. See the *Security Server (RACF) Security Administrator's Guide* for more details.

```
//          VOL=SER=USER01
//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=3380,
//          SPACE=(TRK,(2,2)),VOL=SER=USER01
//          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.BRODCAST,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 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 addition 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(VSAMDSET)          /* REMOVING FROM GROUP VSAMDSET */
RE MARKW GRO(SYSCTLG)           /* REMOVING FROM GROUP SYSCTLG  */
DD 'MARKW.**'                   /* DELETING GENERIC SC12104.**  */
DU MARKW                        /* DELETING USER ID SC12104     */
```

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, DELETE USER (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. These will usually include (using MARKW as the target userid):

```
DELDSD 'markw.*'                           (delete data set profile)
DD     'markw.**'                          (DD is the same as DELDSD)
PERMIT 'payroll.input' ID(markw) DEL GEN   (remove from access list)
DELUSER markw                              (delete userid)
RDELETE facility markwx    (general resource profile associated
                            with markw.  No easy way to find these.)
```

5.5 How to Reset a Password

{ZE30} A system administrator is often asked to reset a user's password. There are two common reasons for this:

1. The user forgot his password (or made too many errors when attempting to change it), or
2. 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 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.

```
                        RACF - CHANGE USER MARTIN
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 bl

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

Figure 2. RACF Change User Menu

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 martin

USER=MARTIN   NAME=Martin C.             OWNER=SYS1
 DEFAULT-GROUP=SYS1      PASSDATE=96.137  PASS-INTERVAL= 30
 ATTRIBUTES=REVOKED                              <== note this line
 REVOKE DATE=96.138 RESUME DATE=NONE
 LAST-ACCESS=96.138/10:03:05
 CLASS AUTHORIZATIONS=NONE
 INSTALLATION-DATA A214 12104 12104
 NO-MODEL-NAME
 LOGON ALLOWED   (DAYS)          (TIME)
```

In this case we would probably use this command:

```
ALU martin RESUME PASS(newpwd)      <== if REVOKED
ALU martin PASS(newpwd)             <== if not REVOKED
ALU martin PASS(newpwd) NOEXPIRED   <== if not REVOKED

PASSWORD NOINTERVAL USER(martin)    <== if you want this    )
```

and you would need to tell Martin 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.[35] 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(oldpw newpw)
```

[35] Using this is very poor security, but may be appropriate in smaller, closed systems.

in that PASSWORD must be entered twice.[36] 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 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 in Figure 2 on page 57, 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.

5.6 How to Provide Emergency Userids

{ZE35} 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.)
- The userid is functional without RACF.
- It has a very minimal logon procedure associated with it. (IBMUSER id defined this way, although it is authorized to use several logon procedures.)
- It does not automatically start ISPF, and does not allocate any ISPF data sets (including an ISPF profile data set). (IBMUSER is defined this way.)
- It should have RACF SPECIAL and OPERATIONS privileges, and TSO ACCOUNT, JCL, and OPER authorities. (IBMUSER is defined this way.)
- 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,

[36] Once as the command name, and once as a keyword parameter.

are made only to RACF. To change the SYS1.UADS password, you must use the ACCOUNT command. (See 5.7, "How to Work with SYS1.UADS" on page 59 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.

The OS/390 R6 AD CD-ROM contains a SYS1.UADS entry only for IBMUSER. Earlier versions defined a number of other users in SYS1.UADS.

5.7 How to Work with SYS1.UADS

{ZE36} 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 be 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 the ACCOUNT command, 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 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. 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 all 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 PW88 1 ISPFPROC)   add user OGDEN, with password=PW88,
                      account number = 1, logon procedure = ISPFPROC
ADD (OGDEN PW88 1 ISPFPROC) JCL OPER    add user with attributes
CHANGE (OGDEN PW88 1 ISPFPROC) DATA (NEWPROC) change logon procedure
CHANGE (OGDEN) JCL OPER ACCT     change attributes
CHANGE (OGDEN PW88) DATA(NEWPW)  change password to NEWPW
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 P/390 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.

Unfortunately, default SDSF controls (when not using PARMLIB or RACF for SDSF security controls) 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. (For the AD CD-ROM R6, OPER appears to provide the SDSF authority to inspect other users' output, view the LOG, and view other users and jobs with the DA command.)

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.

Chapter 6. Disk Administration

{ZF00} The available CD-ROM systems[37] are tightly-designed OS/390 systems, built for distribution and use with a minimal number of volumes. You will almost certainly want to add new volumes to meet your particular needs. This will involve creating new (emulated) disk volumes, as discussed in 6.1, "How to Add Additional Disk Volumes" on page 64.

"Real" disks, on mainframes, have specific sizes. For example, a real 3390-1 volume contains 1113 cylinders. The P/390 support programs 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 volume sizes:

Table 4. Basic DASD Geometry

Disk Model	Cylinders	Tracks/Cyl	Bytes/Track	Bytes/Cyl	Bytes/Vol
3380-AJ4	885	15	47476	712140	630MB
3380-AE4	1770	15	47476	712140	1.26GB
3380-AK4	2655	15	47476	712140	1.89GB
3390-1	1113	15	56664	849960	946MB
3390-2	2226	15	56664	849960	1.89GB
3390-3	3339	15	56664	849960	2.83GB
9345-1	1440	15	46456	696840	1.0GB
9345-2	2156	15	46456	696840	1.5GB
emulated 3380	1 - 2655	15	47476	712140	up to 1.89GB
emulated 3390	1 - 3339	15	56664	849960	up to 2.83GB
emulated 9345	1 - 2156	15	46456	696840	

If you have a *dump tape* from a mainframe disk, and you plan to *restore* the volume on your P/390 OS/390 system, 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.[38]

If you want to create more disk volumes for your P/390 OS/390 system, and are not concerned about moving volumes to or from a mainframe, you can create emulated volumes with a wide range of sizes. Note that an OS/390 address is required for each volume. The AD CD-ROM OS/390 systems have an ample

[37] This refers to both the *preconfigured system* and the *Application Development system*.

[38] 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.

number of extra addresses defined. These are listed in 2.7.1.1, "Device Addresses Defined" on page 13.

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 are unreasonable for a small, P/390-size system. A reasonable number of emulated disk volumes might be in the 10 to 20 range. A P/390 (and R/390) system has an absolute maximum of 255 emulated devices, where each "device" is defined by a line in the P/390 configurator file.

This limitation is on the P/390 side. OS/390 can have many more devices defined than are emulated by the P/390, or might be present on a S/390 mainframe.

6.1 How to Add Additional Disk Volumes

{ZF10} 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.[39] 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, so we will restrict our comments to 3380 and 3390 devices.

Follow these steps:

1. 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 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 2.7.1.1, "Device Addresses Defined" on page 13.

2. Examine your OS/2 disks[40] to determine if sufficient disk space is available. Use Table 4 on page 63 to estimate the amount of server disk space required. In OS/2 terms, a volume will be about 2% larger than the logical size indicated in the table.

3. Assign volume serial numbers (*volsers*) for the new volumes. These are six-character names that must be unique within your system. Typical volsers to be added to a P/390 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.

4. Decide on an OS/2 name for the OS/2 file used for each emulated volume. For example:

   ```
   volser    OS/2 file name

   TSOPAK    D:\MVS\TSOPAK.130
   WORK01    D:\MVS\WORK01.131
   SPOOL1    E:\SPOOL1.128
   ```

   You can use any OS/2 names you like, but we strongly recommend a naming convention similar to that shown here.

39 You can install emulated 3375, 3350 and 3330 volumes, but OS/390 does not support these older devices.

40 This might be a single large disk created by your RAID adapter.

5. While OS/390 is running, use the ISPF editor to add your new volsers 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 6.2, "How to Use OS/390 Volume Use Attributes" on page 66 for more information about mount attributes.

6. Using the P/390 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 128:

```
>128 >3380    >SPOOL1                 >440C  >2 >E:\SPOOL1.128
 131  3380     WORK01                  440C   2  D:\MVS\WORK02.131
 130  3380     TSOPAK                  885C   2  D:\MVS\TSOPAK.130
```

These are large files and the formatting process can take many minutes.

IMPORTANT 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 Chapter 6, "Disk Administration" on page 63 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 in Chapter 6, "Disk Administration"; note that 3390-9 emulated is not provided.

7. At this point, the P/390 subsystem must be restarted in order to include the new volumes.

8. After restarting the P/390 subsystem and after OS/390 is IPLed, you should place the new volume offline, using an OS/390 command like V 128,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 them.

9. 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(128) NOVERIFY VOLID(SPOOL1) OWNERID(P390) -
      VTOC(0,1,14)
/*
```

(If you are a purist, you might want to position the VTOC for the scratch volume in the center of the volume.) 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.

10. After the job completes, you can vary the volume online and issue an appropriate MOUNT command, 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 DASDI" a volume. These terms are not completely logical, but they are widely used.

Which emulated disk type should you use? For practical purposes, they all have equal performance. If you need a particular device type, then use it. Otherwise, 3390 is probably the best choice (or the most modern choice). 9345 drives have not been as widely used as 3380s and 3390s. Old-time MVS users tend to be more comfortable with 3380s, but for no logical reason. In terms of emulated functions, there is no difference; 3380s, 3390s, and 9345s are all emulated with ECKD support.

These steps do not create a DFSMS-managed volume. See 6.9, "How to Create and Use SMS Disks" on page 75 for a discussion of DFSMS and HFS volumes.

6.2 How to Use OS/390 Volume Use Attributes

{ZF15} *The following discussion does not apply to DFSMS-managed volumes.* Remember that any volume containing HFS data sets must be managed by DFSMS.

In principle, OS/390 *use attributes* apply to disk and tape volumes. In practice, the use mount attribute 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

123456789112345678921 2           <-- indicates columns
         0         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.

6.3 How to Inspect the Master Catalog

{ZF20} You can use the batch job described in 8.23, "How to Make a Trivial IDCAMS Job" to submit this IDCAMS command:

```
LISTCAT ALL CAT(OS390.MASTER.CATALOG)            (in batch)

LISTCAT ALL CAT('OS390.MASTER.CATALOG')          (in TSO)
```

to list everything in the master catalog.[41] Our system produced about 11,000 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

[41] The name shown here, OS390.MASTER.CATALOG, is the name of the master catalog in the AD CD-ROM OS/390 R6 system. There is nothing magic about this name, and other OS/390 or MVS systems may have a different name for the master catalog.

you probably do not want. The information you probably want is structured something like this:

```
NONVSAM ----- SYS1.SCLDBLD
   HISTORY
       DATASET OWNER -- (NULL)     CREATION ---- 1997.030
   VOLUMES
       VOLSER ---------  OS39RB    DEVTYPE -------X'xxxxxxxx'
   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.[42] 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's 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(OS390.MASTER.CATALOG)     (use batch, not 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 the relevant material from the AD OS/390 R2 system was:

```
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
```

42 In some ways, this use of a catalog alias for a data set name is similar to the UNIX 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 MVS and OS/390* , and is a relatively recent feature of the catalogs. We strongly suggest you do not use it often -- traditional MVS customers will find it disconcerting. It is used in the AD CD-ROM systems to provide a transition from a nonstandard naming convention used by earlier AD systems to the use of standard names for program product libraries.

```
        VOLSER ---- OS39H2
      ASSOCIATIONS
        ALIAS  ---- HFS
    USERCATALOG --- USERCAT.OS39R1   (catalog on previous IPL volume)
      VOLUMES
        VOLSER ---- OS39R1
      ASSOCIATIONS
        ALIAS  ---- OS39R1AD
        ALIAS  ---- OS390R1
        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
        VOLSER ---- P390DX
      ASSOCIATIONS
        ALIAS  ---- CICSTZN
        ALIAS  ---- CICTZN
        ALIAS  ---- CICSDZN
        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, SCPMV5, or almost anywhere else. A user catalog on a non-SMS- managed volume can catalog SMS-managed data sets, but this situation is usually avoided.
- The ALIAS for a user catalog 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 automatic 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 the OS/390 R2 version of the AD system. The implications is that you should use release-independent user catalogs for your data sets.

6.4 How to Define User Catalogs

{ZF25} 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 3.4, "Planning Your User Catalog(s)" on page 31 and 6.5, "How to Migrate Your User Catalog" on page 71 for more discussion about the need for 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 ( USERCAT.RACCAT )    +
         ICFCATALOG                 +
         VOLUME ( RACCAT )          +
         CYLINDERS ( 15 5 ) )
```

This JCL defines a User Catalog named USERCAT.RACCAT on volume RACCAT 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 (15 cylinders 3380 space) for a P/390, and should be ample for most cases.

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 SYS1.PARMLIB in the CLASS NAME(IGGCAS) section; that is, add after other EMAJ(xxxxx) statements but before the MAXVIRT(256) line

```
          EMAJ(USERCAT.RACCAT)
```

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(USERCAT.RACCAT) )
  DEFINE ALIAS ( NAME(USERID2) RELATE(USERCAT.RACCAT) )
  DEFINE ALIAS ( NAME(PAYROLL) RELATE(USERCAT.RACCAT) )
  DEFINE ALIAS ( NAME(JOE)     RELATE(USERCAT.RACCAT) )
/*
```

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 USERCAT.RACCAT and not in the master catalog. You would change USERCAT.RACCAT 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:

1. Defining initial ALIAS entries when a new user catalog is created,

2. 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 5.2, "How to Add New Users" on page 48.

You can add an ALIAS using TSO commands, although we do not recommend this because the potential for mistyping a parameter 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:       'usercat.raccat'
```

and the new alias is defined and ready for use.

The following job will list full details of your user catalog:

```
//P390C JOB 1,P390,MSGCLASS=X
//LIST   EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
  LISTCAT ENT(usercat.raccat) ALL CAT(usercat.raccat)
/*
```

6.5 How to Migrate Your User Catalog

{ZF27} If you migrate to a new release of OS/390 by completely loading the new system 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 migration 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(usercat.raccat) ALL CAT(OS390.master.catalog)
/*
```

You can use the output from this job to ensure that your ALIAS job (see 6.3, "How to Inspect the Master Catalog") is complete. OS390.MASTER.CATALOG is the name of the master catalog in the AD system; your system might have a different name, of course.

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 (( usercat.raccat    +
         DEVICETYPE(3380) VOLUMES(raccat) ) )
/*
```

The second step is to re-establish the ALIAS pointers to your reconnected user catalogs. The job shown in 6.3, "How to Inspect the Master Catalog" on page 67, using the DEFINE ALIAS statements you have maintained as you added aliases to the system, will complete the catalog migration.

6.6 How to Erase Scratch Disks

{ZF29} 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:

- Reinitialize the whole volume, using an ICKDSF job such as the one shown in 6.1, "How to Add Additional Disk Volumes" on page 64. 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 SYS96180.T120135.RA000.OGDEN.R0100029. The first qualifier contains the date. If the date is older than the current[43] date, the data set is almost certainly unwanted. Enter "D" at the beginning of the line (in the ISPF display) and press ENTER. The data set will be deleted. You can quickly delete

[43] The OS/390 command "D T" can be used to display the current date in YYDDD format.

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.

6.7 How to Change a Volume Serial Name

{ZF30} 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(oldser) VOLID(newser)
/*
//* 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 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.

6.8 How to Remove VSAM Data Set Components

{ZF31} The normal way do 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* (SC26-4906). 6.9, "How to Create and Use SMS Disks" on page 75 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 HFS volumes for the AD CD-ROM system.

The AD system contains a storage group named DEFAULT.[44] The DEFAULT group contains volser names such as OS39H1, OS39H2, and so forth.[45]

To add volsers, follow these steps:

1. Select ISMF from the ISPF menu. (It is in the "M" group.)
2. If option 6 is not shown in the primary ISMF menu, do this:
    a. Select option 0, ISMF Profile
    b. Select option 0, User Mode
    c. Select option 2, Storage Administrator
    d. Use PF3 repeatedly and completely exit from ISMF.
    e. Start ISMF again and you should see option 6.
3. Select option 6, Storage Groups. Enter CDS Name = 'SYS1.SCDS' and Storage Group Name = DEFAULT.
4. Select option 1, List, from the options presented in the middle of the display. This should produce a STORAGE GROUP LIST.
5. 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.
6. Press PF3 twice to return to the STORAGE GROUP APPLICATION SELECTION display. Select option 4, Volume, on this screen and press enter.

[44] 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"

[45] It should contain OS39H6 and HFSUS1, but an error in the building process omitted these. Nevertheless, volsers OS39H6 and HFSUS1 work as HFS volumes. We do not know why. If you add volsers to this group, we suggest you also add OS39H6 and HFSUS1 if they are not already there.

7. 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.
8. This displays an SMS VOLUME STATUS DEFINE screen. Press PF3. You should receive the message ALL VOLUMES DEFINED.
9. Press PF3 twice to return to the primary ISMF screen.
10. Select option 8, Control Data Set.
11. Select option 5, Activate, and enter Data Set Name = ′SYS1.SCDS′. Press enter.
12. Enter / (a slash character) in the appropriate field on the verification screen that follows.
13. 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.
14. You need to initialize any new HFS volumes with the STORAGEGROUP attribute. See 6.9, "How to Create and Use SMS Disks" for a sample job.

6.9 How to Create and Use SMS Disks

{ZF35} See 2.12, "DFSMS Definitions and Planning" on page 19 for a shortcut to define additional HFS volumes for the AD system. Also note that this description does not exactly parallel the AD R6 system (or later); these releases provide an additional HFS volume, HFSUS1, that is MOUNTed as /u.

System Managed Storage (SMS or DFSMS) is intended automate and ease the management of large numbers of disk volumes by making the particular volume (volser) used transparent to the user. SMS routines analyze the characteristics of a data set (usually by inspecting the associated JCL) 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 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 a 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 P/390 environment, and new P/390 OS/390 owners probably do not want another complex function that they must administer. Some areas of OS/390 usage *require* the use of SMS, and OpenEdition is one of these areas. In particular, Hierarchical File Systems (HFS) data sets *must* reside on SMS-managed volumes.

You may have one of these situations:

1. You want to fully implement DFSMS functions. In this case, we assume you are familiar with DFSMS administration and the steps outlined in this section are not relevant.

```
                          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 3. ISMF Primary Option Panel (for Administrators). This is the starting point for adding new HFS volumes. Select the Storage Class option.

2. You will not use DFSMS at all; this implies little or no usage of OpenEdition.[46]

```
                     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 ===> hfs            (For Storage Class List, fully or
                                          partially specified or * for all)

SELECT ONE OF THE FOLLOWING OPTIONS  ===> 3

 1   LIST    - Generate a list of Storage Classes
 2   DISPLAY - Display a Storage Class
 3   DEFINE  - Define a Storage Class
 4   ALTER   - Alter a Storage Class

IF OPTION 1 CHOSEN ABOVE,
   RESPECIFY VIEW CRITERIA           ===> N  (Y or N)
   RESPECIFY SORT CRITERIA           ===> N  (Y or N)

USE ENTER TO PERFORM SELECTION; USE END COMMAND TO EXIT.
```

Figure 4. ISMF Storage Class Panel

[46] Since the AD system already has one SMS-managed volume defined (OS39H2 or OS39H3), and a number of HFS data sets already on it, you can use OE with this volume without doing any additional SMS administration. This is acceptable for trivial

```
                             STORAGE CLASS DEFINE
COMMAND ===> _

SCDS NAME:          SYS1.SCDS.DATA
STORAGE CLASS NAME: HFS

TO DEFINE STORAGE CLASS, SPECIFY:
  DESCRIPTION ===> storage class for hfs files
              ===>
  PERFORMANCE OBJECTIVES
   DIRECT MILLISECOND RESPONSE     ===>              (1 to 999 or blank)
   DIRECT BIAS                     ===>              (R, W or blank)
   SEQUENTIAL MILLISECOND RESPONSE ===>              (1 to 999 or blank)
   SEQUENTIAL BIAS                 ===>              (R, W or blank)
   INITIAL ACCESS RESPONSE SECONDS ===>              (0 to 9999 or blank)
   SUSTAINED DATA RATE (MB/SEC)    ===>              (0 to 999 or blank)
  AVAILABILITY                     ===> N            (C, P, S or N)
  ACCESSIBILITY                    ===> N            (C, P, S or N)
  GUARANTEED SPACE                 ===> y            (Y or N)
  GUARANTEED SYNCHRONOUS WRITE     ===> N            (Y or N)

USE ENTER TO PERFORM VERIFICATION; USE END TO SAVE AND EXIT;
```

Figure 5. ISMF Storage Class Definition Panel

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

TO PERFORM STORAGE GROUP OPERATIONS, SPECIFY:

  CDS NAME           ===> 'SYS1.SCDS.DATA'
                              (1 to 44 character data set name or 'ACTIVE')
  STORAGE GROUP NAME ===> hfs               (for Storage Group List, fully or
                                             partially specified or * for all)
  STORAGE GROUP TYPE ===> pool              (VIO, POOL, DUMMY, OBJECT,
                                             OBJECT BACKUP, or TAPE)
SELECT ONE OF THE FOLLOWING OPTIONS ===> 2

  1  LIST    - Generate a list of Storage Groups
  2  DEFINE  - Define a Storage Group
  3  ALTER   - Alter a Storage Group
  4  VOLUME  - Display, Define, Alter or Delete Volume Information

IF OPTION 1 CHOSEN ABOVE,
  RESPECIFY VIEW CRITERIA            ===> N   (Y or N)
  RESPECIFY SORT CRITERIA            ===> N   (Y or N)

USE ENTER TO PERFORM SELECTION; USE END COMMAND TO EXIT.
```

Figure 6. ISMF Storage Group Application Panel

OE usage, but is probably not acceptable for significant use because all your HFS files will be on a volume that is replaced by new releases of the AD CD-ROM.

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

SCDS NAME:          SYS1.SCDS.DATA
STORAGE GROUP NAME: HFS

TO DEFINE STORAGE GROUP, SPECIFY:
 DESCRIPTION  ===> storage pool for hfs files
              ===>
 AUTO MIGRATE ===> n  (Y, N, I or P)  MIGRATE SYS/SYS GROUP NAME ===>
 AUTO BACKUP  ===> n  (Y or N)        BACKUP SYS/SYS GROUP NAME  ===>
 AUTO DUMP    ===> N  (Y or N)        DUMP SYS/SYS GROUP NAME    ===>

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

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

 DEFINE SMS STORAGE GROUP STATUS      ===> N   (Y or N)

USE ENTER TO PERFORM VERIFICATION AND SELECTION; END TO SAVE AND EXIT.
```

Figure 7. ISMF Pool Storage Group Definition Panel

3. You want to do the minimum SMS administrative work necessary to add one or more additional volumes for OE, and allocate HFS data sets on these volumes. The remainder of this section addresses this process.

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

CDS NAME:           SYS1.SCDS.DATA
STORAGE GROUP NAME: HFS
STORAGE GROUP TYPE: POOL

SELECT ONE OF THE FOLLOWING OPTIONS   ===> 2

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

SPECIFY A SINGLE VOLUME (in PREFIX), OR RANGE OF VOLUMES:
       PREFIX   FROM     TO    SUFFIX  HEX
       ______  ______  ______  _____    _
 ===>  hfs001                                  ('X' in HEX field allows
 ===>                                           FROM - TO range to include
 ===>                                           hex values A through F.)
 ===>

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

Figure 8. ISMF Storage Group Volume Selection Panel

The AD system already has one SMS-managed volume containing the HFS files delivered with OS/390. This volume is in storage group DEFAULT. The following instructions will define another SMS storage group named HFS, and add new volume HFS001 to this group. You can easily add additional volumes, such as HFS002 and so forth, to this group. You will have two storage group classes, DEFAULT and HFS, and can direct new HFS file systems (new PDSE/HFS allocation) to one of these two groups.

1. Define a new volume to your system. This must be undertaken with the configurator, see 6.1, "How to Add Additional Disk Volumes" on page 64 for details. OS/390 must be IPLed (or reIPLed) to gain access to the volume.

2. Vary the device offline to OS/390 using the following system command (assuming the new volume is at address 126):

   ```
   V 126,OFFLINE
   ```

3. Initialise 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(0126) NOVERIFY  VTOC(0,1,29)  -
       INDEX(2,0,10) OWNERID('P390') VOLID(HFS001) -
       STORAGEGROUP
   /*
   ```

   Note the use of the STORAGEGROUP parameter. SMS volumes also require an index for the VTOC (INDEX parameter).

4. Make the volume available for OS/390 with the following system commands:

   ```
   V 126,ONLINE
   M 126,VOL=(SL,HFS001),USE=PRIVATE
   ```

5. 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 ) )
   /*
   ```

   Do not make the ACDS too small; it may provoke obscure problems. We suggest 15 tracks. Now initialize the ACDS backup with the following system command:

   ```
   SETSMS SAVEACDS(sys1.acds.backup)
   ```

6. After backing up the active SMS environment, you must use the online interface Integrated Storage Management Facility (ISMF) to make changes and additions to SMS. Use ISPF to access the ISMF primary panel shown in Figure 3 on page 76. (To use ISMF, you must have various ISMF libraries, such as its ISPF panel library, allocated in your TSO session. If you use a shortened logon procedure, such as described in 8.9, “How to Make TSO Logon Faster” on page 118, you will need to allocate the required additional data sets, as outlined in 8.26, “How to Set Up an ISPF Administrative Environment” on page 144. The standard 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 but you must 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.

7. First you should define a storage class to allow you to direct allocations of SMS data sets to your SMS disks. Select option 5 from the ISMF primary panel to display the Storage Class Application panel shown in Figure 4 on page 76.

   You should enter the name of the SCDS (Source Control Data Set). 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 4 on page 76, SYS1.SCDS, is the data set used in the AD system.

   Enter the name of the Storage Class you wish to add; this is HFS in our example. Enter 3 in the option field (define). Press Enter to display the Storage Class Define panel shown in Figure 5 on page 77.

   You may wish to experiment with the performance objectives but they hold little meaning for a R/390 or P/390 system. You should enter a meaningful description in the field provided. If you wish to be able to explicitly control the volume when allocating a 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.

8. 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 6 on page 77.

   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 7 on page 78.

   You should turn off AUTO MIGRATE and AUTO BACKUP (change to a ′n′) and ignore the other management options. You should enter a meaningful Description in the field provided. Pressing the END key will complete your Storage Group definition and take you back to the Storage Group options panel shown in Figure 6 on page 77. Now you must add the SMS volume to the group so enter 4 in the options field to display the Storage Group Volume Selection panel shown in Figure 8 on page 78.

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

SCDS NAME:               SYS1.SCDS.DATA
STORAGE GROUP NAME:      HFS
VOLUME SERIAL NUMBERS:   HFS001


TO DEFINE SMS VOLUME STATUS, SPECIFY:

   SYSTEM/SYS GROUP      SMS VOL
   NAME                  STATUS
   ----------------      -------
   P390             ===> ENABLE        ( Possible SMS VOL STATUS values
                    ===>                 for each system or system group:
                    ===>                 NOTCON, ENABLE, DISALL, DISNEW,
                    ===>                 QUIALL, or QUINEW )
                    ===>
                    ===>                 * SYS GROUP = sysplex minus systems
                    ===>                   in the sysplex explicitly defined
                    ===>                   in the SCDS

USE ENTER TO PERFORM VERIFICATION; END TO SAVE AND EXIT.
```

Figure 9. ISMF SMS Volume Status Define Panel

You must enter 2 in the option field and enter the name of the SMS volume in the first Prefix field (HFS001 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 HFS 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 9.

You should accept the default of ENABLE for your volume. 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.

```
                          ACS APPLICATION SELECTION
COMMAND ===> _

SELECT ONE OF THE FOLLOWING OPTIONS ===> 1

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

IF OPTION 5 CHOSEN ABOVE, SPECIFY:

  CDS NAME  ===> 'SYS1.SCDS.DATA'
                              (1 to 44 character data set name or 'ACTIVE')

USE ENTER TO PERFORM SELECTION; USE END COMMAND TO EXIT.
```

Figure 10. ISMF ACS Application Selection Panel

```
  Menu  RefList  RefMode  Utilities  LMF  Workstation  Help
------------------------------------------------------------------------------
                               Edit Entry Panel
Command ===> _

ISPF Library:
   Project . . . SYS1
   Group . . . . P390     . . .          . . .          . . .
   Type  . . . . CNTL
   Member  . . . acssgrp         (Blank or pattern for member selection list)

Other Partitioned or Sequential Data Set:
   Data Set Name . . .
   Volume Serial . . .           (If not cataloged)

Workstation File:
   File Name . . . . .

Initial Macro  . . . .                 Options
Profile Name . . . . .                    Confirm Cancel/Move/Replace
Format Name  . . . . .                    Mixed Mode
Data Set Password  . .                    Edit on Workstation
```

Figure 11. ISMF ACS Edit Entry Panel

9. 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 10.

   Select Option 1 to begin editing the ACS Storage Group ACS routine supplied with the AD system. The Edit Entry panel shown in Figure 11 is displayed.

   Enter the name of the AD system ACS storage group source code in this standard ISPF editor panel. In the AD system, this is SYS1.P390.CNTL(ACSSGRP). Press Enter to edit the existing source code. Make the changes shown in Figure 12.

```
  File  Edit  Confirm  Menu  Utilities  Compilers  Test  Help
-------------------------------------------------------------------------------
EDIT       SYS1.P390.CNTL(ACSSGRP) - 01.05                 Columns 00001 00072
Command ===>                                                  Scroll ===> CSR
             _
****** ***************************** Top of Data ******************************
000100         PROC STORGRP
000110         IF &STORCLAS = 'HFS' THEN
000200           SET &STORGRP='HFS'
000201         ELSE
000210           SET &STORGRP='DEFAULT'
000300         END
****** **************************** Bottom of Data ****************************
```

Figure 12. ISMF ACS Edit Source Panel

```
                             TRANSLATE ACS ROUTINES
 COMMAND ===> _

 TO PERFORM ACS TRANSLATION, SPECIFY:

    SCDS NAME           ===> 'SYS1.SCDS.DATA'
                                      (1 to 44 character data set name)

    ACS SOURCE DATA SET ===> 'SYS1.P390.CNTL'
                                      (1 to 44 character data set name)

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


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


 USE ENTER TO PERFORM ACS TRANSLATION; USE END COMMAND TO EXIT.
```

Figure 13. ISMF Translate ACS Routines Panel

These changes allow you to direct the allocation of data sets to the new SMS volume(s) by specifying the "HFS" storage class.[47] When you have finished the changes press END until you go back to the ACS Application Selection panel shown in Figure 10 on page 81. From that panel select Option 2 Translate to display the panel shown in Figure 13.

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

```
   Menu  Utilities  Compilers  Help
 -------------------------------------------------------------------------------
 BROWSE    P390.LIST                                          ACS OBJECT SAVED
 Command ===> _                                                Scroll ===> CSR

********************************* Top of Data **********************************
ACS TRANSLATOR ***** TIME 09:36:13 DATE 06/13/1996 PAGE 0001 *****


SCDS NAME:                SYS1.SCDS.DATA
ACS SOURCE DATA SET:      SYS1.P390.CNTL
ACS SOURCE MEMBER:        ACSSGRP

0001              PROC STORGRP
0002              IF &STORCLAS = 'HFS' THEN
0003                SET &STORGRP='HFS'
0004              ELSE
0005                SET &STORGRP='DEFAULT'
0006              END

TRANSLATION RETURN CODE:  0000
```

Figure 14. ISMF ACS Translator Result Panel

[47] Perhaps our example was not well chosen. In this case, the name HFS is simply an arbitrary name for a storage class. We could have used the name XYAZQW. The use of HFS here should not be confused with the specification of HFS as a type of data set. We intend to use our HFS storage class to hold HFS data sets, but the two uses of the name HFS are independent.

```
                          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 15. ISMF Output Listing Disposition Panel

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 14.

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 15.

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.

10. 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 16.

    Select Option 4 to begin the validation process. The Validate ACS Routines or Entire SCDS panel shown in Figure 17 on page 85 is displayed.

```
                          CDS APPLICATION SELECTION
COMMAND ===> _

TO PERFORM CONTROL DATA SET OPERATIONS, SPECIFY:

  CDS NAME ===> 'SYS1.SCDS.DATA'
                         (1 to 44 character data set name or 'ACTIVE')

SELECT ONE OF THE FOLLOWING OPTIONS ===> 4

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

USE ENTER TO PERFORM SELECTION; USE END COMMAND TO EXIT.
```

Figure 16. ISMF CDS Application Selection Panel

```
                       VALIDATE ACS ROUTINES OR ENTIRE SCDS
COMMAND ===> _

TO PERFORM VALIDATION, SPECIFY:

  SCDS NAME       ===> 'SYS1.SCDS.DATA'
                                  (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 ===> LIST
                                  (1 to 44 character data set name)


USE ENTER TO PERFORM VALIDATION; USE END COMMAND TO EXIT.
```

Figure 17. ISMF Validate ACS Routines or Entire SCDS 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 revalidate before proceeding. When you press END, you will see another Output Listing Disposition panel similar to Figure 15 on page 84.

When you have a successful validation press END until you go back to the CDS Application Selection panel shown in Figure 16 on page 84 and select Option 5. This will display the Confirm Activate Request panel shown in Figure 18.

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 a short time later you will receive a TSO line mode message:

```
                             CONFIRM ACTIVATE REQUEST
COMMAND ===> _

  TO CONFIRM ACTIVATION ON THE FOLLOWING CONTROL DATA SET:

      CDS:  SYS1.SCDS.DATA

  SPECIFY THE FOLLOWING:
      Enter "/" to select option     /  Perform Activation

USE ENTER TO PERFORM OPERATION; USE END COMMAND TO EXIT.
```

Figure 18. ISMF CDS Confirm Activate Request Panel

```
 Menu  RefList  Utilities  Help
                          Allocate New Data Set

Command ===> _

                                                                More:     +
Data Set Name . . . . : P390.SMS.ALLOC

Management class. . . .                 (Blank for default management class)
Storage class . . . . . hfs             (Blank for default storage class)
  Volume serial . . . .                 (Blank for authorized default volume)
Data class  . . . . . .                 (Blank for default data class)
  Space units . . . . . MEGABYTE        (BLKS, TRKS, CYLS, KB, MB or BYTES)
  Primary quantity. . . 50              (In above units)
  Secondary quantity. . 5               (In above units)
  Directory blocks. . . 5               (Zero for sequential data set) *
  Record format . . . . FB
  Record length . . . . 80
  Block size  . . . . . 23440
  Data Set Name type. . HFS             (LIBRARY, HFS, PDS, or blank)  *
  Expiration date . . .                 (YY/MM/DD, YYYY/MM/DD
                                         YY.DDD, YYYY.DDD in Julian form
 Enter "/" to select option              DDDD for retention period in days
```

Figure 19. ISPF 3.2.M Enhanced Data Set Allocation Panel

```
IGD008I New Configuration activated from SCDS SYS1.SCDS
```

6.9.1 Allocating To SMS Volumes

To allocate a data set (which might be a whole HFS) on an SMS-managed volume, you must use the Storage Class parameter. The values to use are DEFAULT for the OS39Hx volume (in the AD system) or HFS 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 = DEFAULT or HFS

You can allocate data sets on SMS volumes by using the ISPF 3.2 Data Set Utility panel. You must use the option M (Enhanced data set allocation) not the older option A (Allocate new data set). Using the ISPF 3.2.M option will display the panel shown in Figure 19 [48]

You must use a value in the Storage Class field of DEFAULT or HFS if you followed the example in the previous section. SMS will select the volume for allocation and ignore any value in the Volume Serial field. If you do not put a value in the Storage Class Field and enter the volume serial number of an SMS volume in the Volume Serial field, SMS will fail the allocation with a message indicating that non-SMS data sets are not allowed on SMS volumes.

[48] If you are using MVS 5.2.2 and you use the ISPF 3.2.M Enhanced data set allocation panel to allocate HFS files by entering the value HFS in the Data Set Name Type field it will fail with a message indicating the value is not valid. The panel as distributed in the MVS 5.2.2 release of the AD has a defect. Either research and apply the correcting PTF for APAR OW15847 or edit member ISRUAASE in data set SYS1.SISPPENU and change: IF (&ZDFSMS12 = 'Y') to IF (&ZDFSMS12 = 'YES')

6.10 How to Select Data Set Names

{ZF40} 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 P/390 installation will have no established rules for data set names.

The basic OS/390 rules are:

- A 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.

On top of 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.[49]
- 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
  - Contains assembler source: ASM
  - Contains object modules: OBJ, OBJECT, OBJLIB
  - Contains load modules: LOAD, LINKLIB, LOADLIB
  - Contains JCL procedures: PROC, PROCLIB
  - Contains C: C
  - Contains text (documentation): TEXT, DOC
  - Contains DCF documents (source): SCRIPT

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.
- 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.

[49] 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.

- If a data set is a PDS, use the letters LIB somewhere in the name. There is no fixed way to recognize a PDS (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)
```

6.11 How to Recognize a VSAM Catalog

{ZF42} Data set names such as Z9999994.VSAMDSPC.etc.. usually are VSAM catalogs. There have been three versions of catalogs during the life of MVS: CVOLs, VSAM, and ICF. Current MVS (and OS/390) 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.

Chapter 7. RACF Administration

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.[50].

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 subjects in this chapter 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.

7.1 Basic Security Decisions

{ZG01} 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 alternation 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:

Table 5 (Page 1 of 2). Security considerations

	closed development environment	closed production environment	open production environment
Password interval?	NO	MAYBE	YES
Length of interval?		180 days	60 days
Automatically protect user data sets?	probably NO	MAYBE	YES
Level of automatic or standard protection? (NONE = no access by other users)	READ	READ	NONE
Protect SYS1.PARMLIB, etc from changes by normal users?	YES	YES	YES
Limit RACF authority to an administrator?	MAYBE	YES	YES

50 In particular, OpenEdition requires a number of functions from the security product.

Table 5 (Page 2 of 2). Security considerations			
Permit anyone to update authorized libraries?	PROBABLY	NO	NO
Permit users to browse (SDSF) output of other users?	YES	PROBABLY	NO
Provide many users with UID=0 for Open Edition?	MAYBE	NO	NO
Require operator logon for using additional OS/390 consoles?	NO	NO	YES

7.2 How to Display RACF Data

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. They 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 for a RACF entry in order to display it.* A normal TSO user can display only the RACF data relevant to himself. A user with SPECIAL authority can display almost anything.[51]

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
```

[51] 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.

```
                        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

  99  EXIT
```

Figure 20. RACF Services Option Menu

```
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:

```
//P390S  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.BRODCAST,DISP=SHR
//SYSTSIN  DD   *
LD DA('MARTIN.*') ALL
LU MARTIN
/*
```

```
                   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 21. RACF Data Set Profile Services Panel

```
                    RACF - DATA SET PROFILE SERVICES - DISPLAY
COMMAND ===>

ENTER THE FOLLOWING INFORMATION:

   PROFILE NAME         ===> 'P390.open1.*'

   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 22. RACF Data Set Profile Name Example

An alternate way to display RACF data is to use the RACF ISPF panels.[52] 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.)

```
                         RACF - DISPLAY DATA SET PROFILE
COMMAND ===>

   PROFILE: 'P390.OPEN1.*'

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 23. RACF Display Data Set Profile Panel

[52] This assumes that the appropriate data sets are allocated to your TSO session. The full-function logon procedure provided with the AD CD-ROM does this. A smaller logon procedure, as recommended in 8.9, "How to Make TSO Logon Faster" on page 118 would probably not allocate the RACF panel libraries. You may need to execute a CLIST to do this, as described in 8.26, "How to Set Up an ISPF Administrative Environment" on page 144.

```
 BROWSE - RACF COMMAND OUTPUT------------------------ LINE
 COMMAND ===>
********************************* Top of Data ************
INFORMATION FOR DATASET P390.OPEN1.* (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 24. A Standard RACF Command Output Display

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 20 on page 91 If we select option 1 (data set profiles), we should see the display shown in Figure 21 on page 91. If we then select choice "8 or D Display" in this panel we will see the panel shown in Figure 22 on page 92.

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 ** operands will not work.[53] If you supply an operand and press ENTER, you should see the panel shown in Figure 23 on page 92. 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 24. It contains all the required information, but is perhaps not as compactly formatted as it might be for viewing on a terminal.

[53] 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.

7.3 How to Back Up and Unload RACF

{ZG10} The following job will allow you to create a back up 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). If you do not want to verify the structure of the data base, you could use the IEBGENER utility to copy the database.

```
//P390K JOB 1,P390,MSGCLASS=X
//STEP      EXEC   PGM=IRRUT200
//SYSRACF   DD     DSN=SYS1.RACF,DISP=SHR
//SYSUT1    DD     DSN=SYS1.RACF.BKUP,DISP=(NEW,CATLG,DELETE),
//          UNIT=3380,SPACE=(CYL,(10)),DCB=(LRECL=4096,RECFM=F)
//SYSUT2    DD     SYSOUT=*
//SYSPRINT  DD     SYSOUT=*
//SYSIN     DD     *
    INDEX
    MAP
    END
/*
```

The following 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 a backup RACF database for input, but you could also use the active 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, a backup copy is typically used.

```
//P390E JOB 1,P390,MSGCLASS=X
//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=3380,SPACE=(CYL,(10)),
//          DCB=(LRECL=4096,RECFM=VB)
//SYSPRINT  DD     SYSOUT=A
```

7.4 How to Use a Local RACF Database

{ZG12} 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[54], 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.

There are a number of approaches to handling RACF database updates, but these may assume more RACF expertise than can be found in a new OS/390 P/390 installation. Described here is a simplistic approach to this problem. It does not solve all the difficulties, but it is a starting point.

While you are using your “old” OS/390 system (which might be the first one you installed on your P/390 system), copy the RACF database to a local volume --

[54] Or any other OS/390 packaging that is a complete, IPLable system with RACF.

that is, a disk volume that will not be replaced by a new release of OS/390. In this example, we use an emulated 3380 volume named RACCAT. 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.RACF,DISP=SHR
//SYSUT1   DD  DSN=SYS1.RACFNEW,DISP=(NEW,CATLG,DELETE),UNIT=3380,
//             VOL=SER=RACCAT,SPACE=(CYL,(7,1))
//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 90 tracks (but you should verify this on your system); your copy should be this size or larger. 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 your new RACF database to SYS1.RACF, which is the name used by the AD CD-ROM systems and most other OS/390 systems. The steps involved are:

1. 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.
2. Using ISPF, rename SYS1.RACF to SYS1.RACFOLD.
3. Using ISPF, rename SYS1.RACFNEW to SYS1.RACF.
4. Re-IPL.

You will now be using your RACF database, on your RACCAT volume.

When you install a new OS/390 CD-ROM release, you will initially be using the SYS1.RACF that was provided with that release.[55] At some point, after whatever initial work you need with your new OS/390, you would:

1. Work in a single-user environment.
2. Rename SYS1.RACF to SYS1.RACFXXX
3. Catalog the SYS1.RACF on RACCAT (in your new master catalog)
4. Re-IPL.

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 7.2, “How to

[55] Your new release will have a master catalog entry that points to SYS1.RACF on a volume that was part of the new release. This is volume SCPMV5, in the case of the AD CD-ROM systems. Your SYS1.RACF, on another volume, is not cataloged in the master catalog of your new release.

Display RACF Data" on page 90). You might print an unloaded, flat-file listing of the new distributed RACF database and look for new entries (see 7.3, "How to Back Up and Unload RACF" on page 94). Or, you might ignore the problem.

Please note that there are other ways to switch RACF databases. Also, the use of a backup RACF database would slightly complicate the method described above --- although the backup database could be copied and renamed the same way. 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.

7.5 How to Simplify RACF Control of Started Tasks

{ZG15} 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 INIT, ASCH, JES2, and so forth. RACF can control started tasks through its STARTED class. If this class is not activated, there is no problem. The AD CD-ROM system is provided with the RACF STARTED class activated, but with no profiles in it.

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. There can be several dozen of these messages during IPL. The R6 AD system suppresses these messages by listing message number IRR813I in the MPFLST00 member of SYS1.ADCD06.PARMLIB.

The messages do no harm, but you might want to eliminate them to produce a cleaner IPL message stream. The proper way, of course, is to define proper controls for all your started tasks, using various RACF commands. If you are not quite ready for this, you might consider the following:[56]

```
SETR GENERIC(STARTED)
AU startu DEFLTGRP(SYS1) OWNER(SYS1) PASSWORD(obscure) -
  OMVS(UID(0) HOME(/))
RDEF STARTED ** STDATA(USER(startu))
SETR RACLIST(STARTED) REF
```

You must be logged onto TSO with a userid that has SPECIAL privileges to issue these commands. The commands define a new user to RACF, named *startu*. (You can use any name.) The password can be a random string (that cannot be easily guessed) because this userid will never be used for logons. The second statement defines a generic profile in the STARTED class that matches all names, and assigns the userid *startu* to all started tasks. This satisfies RACF to the extent that the console messages are no longer issued.

This solution uses a single userid, startu, for all started tasks. This userid needs to have the necessary access authority for all data sets used by all the started tasks.

[56] The userid defined here is very simple. It has no TSO segment, and is never intended for use by a *real* user. You should never grant additional RACF authorities to this userid. Its sole purpose is to associate a valid userid with started tasks.

If system security is a substantial concern in your installation, you might not want to use the simplistic solution outlined here. A better solution is to define several userids for started tasks, each with a logical grouping of access authorities. You can add additional profiles to the STARTED class with more specific names than the "**" shown above; the more specific profiles will be used instead of the generic ** profile. For example,

```
RDEF STARTED SOCK* STDATA(USER(p390))
```

would cause a started task with the name SOCKET2 to be run with the RACF userid p390 and have whatever access authorities are associated with this userid.

In order to start tasks that use OMVS, the userid must have an OMVS segment. As shown, the OMVS UID is zero; that is, this is a "superuser" or "root" user, and any started task that uses this profile will have root authority.

7.6 How to Submit Jobs Under Another User′s ID

{ZG20} 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 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 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).

7.7 How to Grant RACF Access to Manipulate DASD

{ZG21} 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 ususual 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 daa 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 R6 AD system, 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. (You may need to specify the ADMINISTRATOR keyword in the ADRDSSU job).
- Any job (usually with Superzap) that attempts to maniuplate a VTOC will request permission from the MVS operator.

We suggest that these controls are sufficient for most P/390 development 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
```

7.8 How to Allocate Data Sets With HLQ of Revoked User

{ZG22} 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).

7.9 How to Understand RACF and Daemons

{ZG23} 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 PBX.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 created with:

```
RDEFINE FACILITY BPX.DAEMON UACC(NONE)

PERMIT BPX.DAEMON CLASS(FACILITY) ID(daemonuserid) ACC(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

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.

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 7.23, "How to Program Control Libraries and Files" on page 105 for instructions to make a library or file program controlled.

7.10 How to Merge RACF Databases

{ZG24} 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.[57] 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:

[57] Doing this can be quite complex if the two profiles contain conflicting specifications.

1. 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.)

2. 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 P/390-class installations. Note that DBSYNC is not an IBM-supported program. It is an informal tool provided by Walt Farrell, a well-known consultant from IBM's RACF team.

7.11 How to Allow a User to su to root

{ZG25} 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[58] user must:

1. Know the target userid, and
2. Be permitted to the RACF profile BPX.SUPERUSER.

The BPX.SUPERUSER profile is almost certain to already exist in any current OS/390 system. 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.

7.12 How to See Detailed JES2 Security Messages

{ZG26} 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
```

7.13 How to Delete User with Digital Certificate

{ZG27} 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 downlevel. 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.

[58] Please note that the correct name is now OS/390 UNIX System Services. Informal documentation and discussions tend to use OpenEdition, OE, OMVS, and UNIX Systems Services as synonyms.

7.14 How to Set Rules For RACF Passwords

{ZG28} 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.

7.15 How to Use RACF Global Access Control

{ZG29} 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 R6 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 entrys 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.[59]

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 appropraite.

7.16 How to Find Name of Current RACF Database

{ZG31} 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.

7.17 How to Define A Default OpenEdition Userid

{ZG33} 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.

[59] SYS1.RACF is a good example because you are very unlikely to want global READ access to it.

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('odefu/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.)

7.18 What is a RACF User?

{ZG35} 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.

7.19 Understanding the Difference Between UACC and ID(*)

{ZG34} 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.

They 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.

7.20 How to Locate HFS Files With Invalid UID

{ZG36} 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:

1. Directory lists , with the `ls -al` command for example, list the numeric UID and/or GID instead of a name, and
2. 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.

7.21 How to Find Data Set Names Protected By RACF Profile

{ZG37} 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:

1. Only cataloged data sets are listed. Furthermore, only currently active catalogs are searched.
2. 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 R6, have not activated EGN.)
3. 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.

7.22 Modifying RACF Control in OpenEdition Processes

{ZG38} When a tradtional 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 whole discussion generally assumes you are working with authorized programs and are directly modifying MVS control blocks.)

7.23 How to Program Control Libraries and Files

{ZG39} 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.

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.[60]

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 module 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 9.2, "How to Set and Display Extended Attributes" on page 177.

What does program control do?

---

60 The R5 and R6 AD systems do not have the WHEN(PROGRAM) activated. (We assume this was an oversight.)
If you use the command SETROPTS LIST, the first few lines show the status of WHEN(PROGRAM); in these systems it is set to NOWHEN(PROGRAM). These systems have a number of program controlled libraries defined for TCP/IP, the web server, and so forth. These programs work as expected. From this we conclude that, if WHEN(PROGRAM) is not active, then an address space is never marked as dirty.

In the basic case we are describing here, it indicates that all the controlled modules are considered acceptable[61] by 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:

1. When an address space is created, it is "clean." ("Clean" and "dirty" are key words in the discussion of program control.)
2. 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.
3. 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.
4. 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.

7.24 How to Permit Users To View OMVS Segments

{ZG40} The OMVS segment in RACF contains a user′s UID, his home directory path, and his initial program when he uses an OpenEdition shell. 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 data.

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
```

[61] 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."

1. A simple LU command lists basic RACF (but not OMVS) data for the current user.
2. 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.
3. 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.
4. 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.
5. 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.

7.25 How to Use TSO Authorities Profiles

{ZG42} 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.[62] 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:

62 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.

```
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 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.[63]

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 P/390 development installations, we suggest READ is an appropriate level. The AD system uses NONE, for reasons we were unable to discover.

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.

There are a number 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. We changed our AD system to make some of these profiles have UACC(READ). After doing this, we did not need to issue PERMIT commands whenever we added a new TSO user. We used 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
```

[63] Most of these profiles are binary decisions. Any access level other than NONE is treated as permitting access.

```
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 R6 AD system, it will allow
all users to view all SDSF output:

RALT TSOAUTH OPER UACC(READ)
SETROPTS RACLIST(TSOAUTH) REFRESH
```

You can read more about TSO authorization controls in *Security Server (RACF) Security Administrator's Guide* (SC28-1915).

7.26 Accessing You Own Data Sets

{ZG43} 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 has ALTER access to all data sets that have his userid as the high level qualifier, whether there is a profile for these data sets or not.

Chapter 8. Miscellaneous Administration and Techniques

This chapter contains a collection of various Hints and usage Techniques. It not in any order and is not intended to be read sequentially. Many of the tips and techniques apply to any small OS/390 system, but most specific details are for the AD system.

8.1 How to Collect SMF Data

{ZH02} 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 CD-ROM systems do have an SMF output data set defined, SYS1.MAN1, with 20 cylinders of 3380 space. If you want to collect a small amount of SMF data, for a limited time, this may be sufficient. If you want to routinely operate with SMF recording, you will need to allocate at least one additional SMF data set. *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)
```

Additional 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.MAN2),VOLUME(xxxxxx),   +
                  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 SYS1.PARMLIB(SMFPRM00)), by issuing the OS/390 command:

```
SET SMF=00
```

Please note that IPLing can be difficult if you have an invalid SMFPRM00 parameter. We suggest that you copy the original SMFPRM00 member into SMFPRM01 (replacing an irrelevant member there). If there is an IPL problem, you can specify that the alternate SMF member be used.[64]

[64] You will need to be able to perform a "manual IPL" to do this.

SMF can create considerable overhead, and fill the 20 cylinders of the original SYS1.MAN1 data set 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

8.2 How to Remove RMF Automatic Start

{ZH04} Earlier AD CD-ROM system automatically started RMF during the IPL process. These versions also produced 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 SYS1.PARMLIB member COMMND00 (or member VTAMAPPL, depending on your release level) 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 SYS1.PARMLIB, see 4.5, "How to Work With PARMLIB" on page 41 for details on updating SYS1.PARMLIB and how to prevent the situation where your change prevents OS/390 from starting. More information about using RMF is in 8.22, "How to Use RMF (Resource Management Facility)" on page 138.

The current AD systems do not automatically start RMF.

8.3 How to Increase SPOOL Space

{ZH06} The AD CD-ROM systems contain a limited amount of SPOOL space. It is sufficient for basic system 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 SPOOL space you need. We selected 500 cylinders (3380), but that was an arbitrary number. You can place your new SPOOL data set on any volume, if there is space, but we elected to create a volume just for SPOOL. See 6.1, "How to Add Additional Disk Volumes" on page 64 for information about creating new emulated disk volumes. We decided to use the new volume for the JES2 checkpoint data set as well as SPOOL data; the new volume was given the volser SPOOL1.

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 a P/390 OS/390 system. The checkpoint data set can be placed on a separate volume, for performance, but we believe this is unnecessary for typical P/390 systems.[65]

After selecting (or creating) a volume for SPOOL, and deciding how much space to allocate, follow these steps:

1. 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 (450 cylinders, same attributes), both on SPOOL1.

2. Copy SYS1.PARMLIB(JES2PARM) to SYS1.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.
   ```

3. Change the SPOOLDEF parameters of JES2PARM (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.[66]

4. Change the CKPTDEF parameters of JES2PARM (change DSN=SYS2.HASPCKPT and VOLSER=SPOOL1). (These names, SYS2..., are arbitrary but we recommend using easily recognized names such as the ones shown.)

5. Stop MVS and re-IPL. 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.

6. When formatting is complete, enter a “$S” 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 $S function.)

8.4 How to Clear & Print SMF Data Sets

{ZH08} 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=23000,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

[65] These considerations would be very different for a P/390 OS/390 system devoted completely to large-scale printing, but this would be a special case.

[66] 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 volser.

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, or later).

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 should accept spanned records. The BLKSIZE is not important; half-track blocking on whatever emulated disk you use is suggested.

8.5 How to Clear SYS1.LOGREC

{ZH10} OS/390 automatically records hardware (and some software) error 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 the 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=IFCEREP1,PARM='CARD'
//SERLOG  DD DSN=SYS1.LOGREC,DISP=OLD
//ACCDEV  DD DUMMY
//EREPPT  DD SYSOUT=*
//TOURIST DD SYSOUT=*
//SYSIN   DD *
   SYSUM
   ACC=Y
   ZERO=Y
/*
```

8.6 How to Change the System ID

{ZH12} 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 SYS1.PARMLIB(SMFPRM00).[67] 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(P390).
- The identifier in SYS1.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.

[67] Or SMFPRMxx if you are using an alternate member.

- The JES2 member name. You can change this by editing the MEMBER NAME=ccccc field in SYS1.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 SYS1.VTAMLST. In the AD CD-ROM system it is the name of the APPL statement for JES2 in SYS1.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.[68]
- 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 (see 12.7, "How to Set Up NJE" on page 218), 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.

8.7 How to Alter TSO Timeout Logoffs

{ZH14} After a period of inactivity, TSO will automatically log off a user. This period is set in SYS1.PARMLIB(SMFPRM00), by the parameter JWT(hhmm). 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(2400), which is 24 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=00
```

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.

Note that the SMF job wait time may not apply to users within the OMVS shell, although this condition may change with future releases.

[68] Shared DASD for emulated 3380/3390 drives is not supported by the P/390 support programs at the time this was written. Likewise, DASD attachment via parallel or ESCON channels was not supported at the time this was written. In this isolated environment, the SMS configuration name is not very relevant.

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. 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 TMOUT environmental variable (placed in /etc/profile, for example) to provide a timeout function for idle shell users.

8.8 How to Make a Logon CLIST

```
 P/390 OS/390 R2                                 Mountain Widget Co.
                                     /'\         Help Desk 555-1234
                                    /%%%%\
                         /)       /%%  %/%\_
             /)        /% |     %/%    /%%%  \_
           /% |      /%  %%\__%%/%%   /%     /%\
         /%   %\_ %/%     |%  %/     /      /%  %\_
       /% /   |%/%/%  %    % /  %   /%%     \      \_
     /% /  %  /   /%%   %       %       %X      %    \ % _
   /% /   %X   %%     %% %      %%     % %     %   \     \
 ----------------------------------------------------------------
    1. Enter WHSE or CUST to start appropriate application.
    2. NOTES training available Sept 8 - Sept 10.  Call 4321.
    3. WORK03 will be scratched in TWO days.  Move your files!
    4.
    5. Complete backups are made every Friday evening.
READY
```

Figure 25. Display Produced by CLIST

{ZH16} 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) for these users, and (2) automatically executing a CLIST from this procedure. A logon procedure typically allocates a number of files so that they will be immediately available to the TSO user without further action. This is addressed in more detail in 5.3, “How to Modify a RACF User Profile” on page 52 and 8.9, “How to Make TSO Logon Faster” on page 118. 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
```

The CLIST you automatically execute may contain anything you want. The example here contains:

1. A small logo.
2. A place for important announcements. (You would need to edit the CLIST to change the announcements.)
3. 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. 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:

1. 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.

2. 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 P390 OS/390 R2                                Mountain Widget Co.
WRITE                                     /'\       Help Desk 555-1234
WRITE                                    /%%%%\
WRITE                         /)       /%%  %/%\_
WRITE             /)        /% |     %/%    /%%%  \_
WRITE           /% |      /%  %%\__%%/%%   /%     /%\
WRITE         /%   %\_ %/%     |%  %/     /      /%  %\_
WRITE       /% /   |%/%/%   %    % /  %  /%%     \      \_
WRITE     /% /  %  /   /%%   %       %       %X      %    \ % _
WRITE   /% /   %X   %%     %% %      %%     % %     %   \     \
WRITE ------------------------------------------------------------------
WRITE 1. Enter WHSE or CUST to start appropriate application.
WRITE 2. NOTES training available Sept 8 - Sept 10.  Call 4321
WRITE 3. WORK03 will be scratched in TWO days.  Move your files!
WRITE 4.
WRITE 5. Complete backups are made every Friday evening.
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
```

3. Test your CLIST, using the TSO command:

   ```
   EX 'SYS1.CLIST(LP390)' CLIST
   ```

4. 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.)
5. 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 overtype this field (and the change is saved in RACF) and name your new logon procedure.

8.9 How to Make TSO Logon Faster

{ZH18} We use the following TSO logon procedure for most users of the AD system. It is a stripped-down version of the full-function logon procedure that is included (in SYS1.TSOPROC) in the AD CD-ROM. Using this procedure reduced the elapsed logon time, for a single user on an idle system, to less than half the time of the full logon procedure.

We created this procedure in SYS1.TSOPROC(USERPROC). It is much easier to start with a copy of SYS1.TSOPROC(ISPFPROC) and remove lines than to enter the whole procedure from a keyboard.

```
//*************************************************************
//*    LOGON PROCEDURE FOR NORMAL TSO USERS   OS/390 R3 AD
//*************************************************************
//USERPROC EXEC PGM=IKJEFT01,REGION=0M,DYNAMNBR=175,
//              PARM='%ISPFCL'
//SYSLBC   DD  DISP=SHR,DSN=SYS1.BRODCAST
//SYSPROC  DD  DISP=SHR,DSN=ISP.SISPCLIB
//         DD  DISP=SHR,DSN=SYS1.CLIST
//         DD  DISP=SHR,DSN=SYS1.SBPXEXEC
//SYSEXEC  DD  DISP=SHR,DSN=SYS1.SBPXEXEC
//SYSHELP  DD  DISP=SHR,DSN=SYS1.HELP
//         DD  DISP=SHR,DSN=ISP.SISPHELP
//ISPMLIB  DD  DISP=SHR,DSN=ISP.SISPMENU
//         DD  DISP=SHR,DSN=SYS1.SBPXMENU
//         DD  DISP=SHR,DSN=ISF.SISFMLIB
//ISPEXEC  DD  DISP=SHR,DSN=ISP.SISPEXEC
//         DD  DISP=SHR,DSN=SYS1.SBPXEXEC
//ISPPLIB  DD  DISP=SHR,DSN=ISP.SISPPENU
//         DD  DISP=SHR,DSN=SYS1.SBPXPENU
//         DD  DISP=SHR,DSN=ISF.SISFPLIB
//ISPSLIB  DD  DISP=SHR,DSN=ISP.SISPSLIB
//         DD  DISP=SHR,DSN=ISP.SISPSENU
//         DD  DISP=SHR,DSN=ISF.SISFSLIB
//ISPTLIB  DD  DISP=SHR,DSN=ISP.SISPTENU
//         DD  DISP=SHR,DSN=SYS1.SBPXTENU
```

```
//          DD  DISP=SHR,DSN=ISF.SISFTLIB
//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
//SYSPRINT DD  TERM=TS
//SYSIN    DD  TERM=TS
//SYSUDUMP DD  SYSOUT=Z
//SYSABEND DD  DUMMY
//SDSFDUMP DD  DUMMY
```

Working with the AD CD-ROM system, we needed to copy the member(s) in SYS1.LOCAL.ISPFPNLS into ISP.SISPPENU.[69] in order for the above procedure to work properly. We did this to avoid an extra concatenation to ISPPLIB.

Before you can use this logon procedure, you must define it to RACF. The commands (from a READY command prompt or ISPF option 6):

```
RDEF TSOPROC USERPROC UACC(READ)
SETROPTS RACLIST(TSOPROC) REFRESH
```

This RACF definition will allow any user to logon with USERPROC. You will not need to PERMIT individual users to use this procedure, because we used UACC(READ). After adding this member, log onto TSO with an appropriate userid. On the TSO/E LOGON menu (where you enter your password), there is a field for the logon procedure name. Enter the new procedure name in this field. TSO will remember the name (in RACF, if you are using RACF TSO segments) and use it thereafter for that userid.

Many of the ISPF menu options, such as RACF and HCD, will not work when using the logon procedure shown here. These functions require data sets that we removed from the logon procedure, in order to make logon faster. You can have administrative users, such as userid P390, continue to use the original logon procedure (with all the data set allocation contained in it), or have administrative users use the new procedure (shown below) and execute a special CLIST before using RACF, HCD, or other advanced ISPF applications. This is described in 8.26, “How to Set Up an ISPF Administrative Environment” on page 144.

This is certainly not a minimum logon procedure, since it preallocates basic ISPF data sets. Nor does it allocate the minimum number of ISPF data sets. However, we think it is a reasonable logon procedure for “normal” TSO users.

[69] This member contained an extension of the primary ISPF menu. By copying it to SYS1.SISPENU (the primary library of ISPF panels), we could remove SYS1.LOCAL.ISPFPNLS from the logon procedure.

8.10 How to Change the ISPF Menu

{ZH20} ISPF panels (menus) are created by ISPF definitions. Products that use ISPF, such as SDSF or RACF, usually provide their own ISPF panel libraries. The primary panel library for ISPF itself is ISP.SISPPENU, as used in the AD CD-ROM system. ISPF has its own language for describing panels, and it is this language that appears in SYS1.SISPPENU.[70] 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 AD CD-ROM system has a member named ISR@PRIM in ISP.SISPPENU. This member has the definitions for the "ISPF Primary Options Menu." 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 26 on page 121. (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.) 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.[71]

If you want to change the primary menu panel, you should copy it (ISR@PRIM and possibly IBMPRODS) to another member in SISPPENU or LOCAL.ISPFPNLS[72], 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. This 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

```
ISPSTART PANEL(ISR@BACK)
```

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:

[70] A starting reference is *Dialog Tag Language Guide and Reference*, SC28-1219.

[71] There is nothing special about names beginning with ISR@, although such names are traditionally used for the primary menu.

[72] Or any other library that is concatenated to these panel libraries.

```
  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 26. ISPF Primary Options Menu Delivered with the AD CD-ROM System

```
)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.

8.11 Understanding Basic SDSF Security

{ZH22} 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:

1. Using an assembled module to control access characteristics for predefined groups of users,
2. Using a SYS1.PARMLIB member to control access characteristics for groups of users defined in the member, or

3. 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 AD system uses an assembled module, ISFPARMS, to control SDSF security. The module used is the default and sample that is provided with the SDSF product. The key portions of this module are:

```
***********************************************************************
*                                                                     *
*        SAMPLE ISFGRP MACROS.  THE ISFGRP MACRO DEFINES A GROUP OF@P6C
*        USERS AND THE SDSF FUNCTIONS THAT ARE AVAILABLE TO A      @P6C
*        MEMBER.                                                   @P6C
*                                                                     *
***********************************************************************
*                                                                     *
*        SAMPLE ISFGRP MACRO FOR SYSTEM PROGRAMMERS.               @P6C
*                                                                     *
***********************************************************************
         SPACE 5
ISFSPROG ISFGRP TSOAUTH=(JCL,OPER,ACCT),                           @N2C-
               AUTH=(LOG,I,O,H,DA,DEST,PREF,SYSID,ABEND,ACTION,        -
               INPUT,FINDLIM,ST,INIT,PR,TRACE,ULOG,MAS,SYSNAME),   @NUC-
               CMDAUTH=(ALL),                                          -
               CMDLEV=7,                                               -
               DSPAUTH=(ALL),                                          -
               DFIELD2=DAFLD2,                                         -
               GPLEN=2,                                                -
               OWNER=NONE,                                         @NJA-
               APPC=ON,                                            @D2A-
               ACTIONBAR=YES,                                      @NUA-
               ACTION=ALL,                                             -
               ISYS=LOCAL,                                         @NUA-
               DADFLT=(IN,OUT,TRANS,STC,TSU,JOB),                  @L2C-
               VALTAB=TRTAB,UPCTAB=TRTAB2,LANG=ENGLISH,DISPLAY=OFF @SEC
         EJECT
***********************************************************************
*                                                                     *
*        SAMPLE ISFGRP MACRO FOR OPERATORS.                        @P6C
*                                                                     *
***********************************************************************
         SPACE 5
ISFOPER  ISFGRP TSOAUTH=(JCL,OPER),                                @N2C-
               AUTH=(LOG,I,O,H,DA,PREF,DEST,SYSID,ACTION,              -
               FINDLIM,ST,INIT,PR,ULOG,MAS,SYSNAME),               @NUC-
               CMDAUTH=(ALL),                                          -
               CMDLEV=7,                                               -
               DSPAUTH=(USERID,NOTIFY,AMSG),                           -
               GPLEN=2,                                                -
               OWNER=NONE,                                         @NJA-
               APPC=ON,                                            @D2A-
               ACTIONBAR=YES,                                      @NUA-
               ACTION=ALL,                                             -
               ISYS=LOCAL,                                         @NUA-
               DADFLT=(IN,OUT,TRANS,STC,TSU,JOB),                  @L2C-
               VALTAB=TRTAB,UPCTAB=TRTAB2,LANG=ENGLISH,DISPLAY=OFF @SEC
```

```
        EJECT
***********************************************************************
*                                                                     *
*       SAMPLE ISFGRP MACRO FOR END USERS.                        @P6C
*                                                                     *
***********************************************************************
        SPACE 5
ISFUSER  ISFGRP TSOAUTH=(JCL),                                    @N2C-
               AUTH=(I,O,H,DA,ST),                                    -
               CMDAUTH=(USERID,NOTIFY),                               -
               CMDLEV=2,AUPDT=10,                                     -
               DSPAUTH=(USERID,NOTIFY),                               -
               PREFIX=USERID,                                         -
               OWNER=USERID,                                      @NJA-
               APPC=ON,                                           @D2A-
               ACTIONBAR=YES,                                     @NUA-
               ACTION=(11,12,USER),                                   -
               DADFLT=(IN,OUT,TRANS,STC,TSU,JOB),                 @L2C-
               VALTAB=TRTAB,UPCTAB=TRTAB2,LANG=ENGLISH,DISPLAY=OFF @SEC
```

This module defines 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′ output,
- The ISFUSER group, which can display only the jobs belonging to the user and whose jobname begins with the appropriate userid.[73]

Look at the first line of each of these groups. It has a field such as TSOAUTH=(JCL,OPER). If you look back at the standard job we suggest you use to add new users, see 5.2, "How to Add New Users" on page 48, you will notice commands such as PERMIT userid JCL CLASS(TSOAUTH). This is the link between RACF and the ISFPARMS table. 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 essential control, in regard to SDSF display authority using the default ISFPARMS module, is with TSO OPER authority. If a user has OPER authority, he can display output for anyone′s jobs. If he does not have OPER authority, he can display output only for his own jobs. (This is not always true, since SDSF controls can be set up in a variety of ways. One alternative is to use ACCT authority to permit users to view other users′ output. This makes an interesting tradeoff - OPER authority can be removed from most users (preventing them from issuing MVS operator commands from their SDSF aessions), but allowing them to add non-RACF users via the intended use of ACCT and the ACCOUNT command.)

[73] These specific names, ISFSPROG and so forth, are arbitrary and just happen to be the names used for the default SDSF module. If you elect to change and reassemble this module, you could use any names. These names have no external significance outside of the ISFPARMS module and are not related to RACF groups.

For this reason, we suggest you exercise control over the PERMIT OPER or ACCT command when you add new users.[74]. You can add authority for a user by issuing this command from the TSO READY prompt or from ISPF option 6:

```
PERMIT OPER CLASS(TSOAUTH) ID(userid)
PERMIT OPER CLASS(TSOAUTH) ID(userid) DEL      (to remove authority)
```

With OPER authority, a user can issue MVS console commands from SDSF. You may not want to do this.

We suggest that the default SDSF security specifications will be acceptable for your normal users (without OPER or ACCT authority) if, *and only if,* their jobnames always begin with their TSO userid. If this is not possible, you will probably need to change SDSF security controls or give these users OPER (or ACCT) authority.

SDSF uses the following libraries. Abbreviated names (as provided by catalog ALIAS entries in the AD CD-ROM system) are shown here. The full names normally include a "middle" qualifier in the format VnRmMx.

```
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                   */
```

If you want to change ISFPARMS, you might:

- Take a backup copy of your original ISFPARMS (which is in ISF.SYSFSRC)
- Edit ISFPARMS as required. There are many potential parameters and you should reference the proper manual (SC23-3808) before starting.
- Save member ISFPARMS on ISF.SISFSRC
- Backup and then edit member ISFASMP of ISF.SISFJCL. Customize it just to re-assemble and re-link module ISFPARMS
- Submit ISFASMP
- After the end of this JOB verify all steps ended with CODE 0.
- If all steps ended correctly, then refresh LLA with the OS/390 command:

  ```
  F LLA,REFRESH
  ```

- Test your new controls.

Converting SDSF fully to RACF controls 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. The version of SDSF with OS/390 R3 should provide a command named ISFARC. This command, issued from ISPF option 6, provides a menu and a number of steps to automatically convert SDSF controls in SYS1.PARMLIB (but not in the macro form) to RACF. You need to work through each step in the conversion program (via its ISPF menu) to

[74] PERMITing a user to ACCT also gives the user the ability to manipulate SYS1.UADS. You do not want to do this

accomplish the conversion. The ISFARC command produces a file of RACF commands which you can execute. You could convert (without executing the resulting RACF commands) several versions of the PARMLIB controls, performing your customization in PARMLIB instead of in RACF. When you are satisfied with the operation of the PARMLIB controls, you would then convert these to RACF and continue your finer-grain customization there.

8.12 How to Permit SDSF Users to Display All Jobs

{ZH23} See 8.11, "Understanding Basic SDSF Security" on page 121 for more details about SDSF security. When internal SDSF security is used and is set up as described[75] in 8.11, "Understanding Basic SDSF Security," you (if you have RACF SPECIAL authority) can alter any user to permit him to use SDSF to view output from all jobs. The RACF command is:

```
PERMIT OPER CLASS(TSOAUTH) ID(userid) ACC(ALTER)
SETROPTS CLASS(TSOAUTH) REFRESH
```

Unfortunately, this permission also permits the useid 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.

(In some cases, the ACCT authority has been used instead of OPER. In this case the command would be:

```
PERMIT ACCT CLASS(TSOAUTH) ID(userid) ACC(ALTER)
```

although this use of ACCT is not common.)

8.13 How to Change JES2 Parameters

{ZH24} JES2 uses member JES2PARM of SYS1.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=
NJEDEF    OWNNODE=
OUTDEF    JOENUM=
SPOOLDEF  BUFSIZE=, DSNAME=, SPOOLNUM=, TGNUM=, TRKCELL=, VOLUME=
```

For example, expanding spool space (as described in 8.3, "How to Increase SPOOL Space" on page 112) will require a cold start.

A cold start begins at the message:

[75] The AD CD-ROM system is set up this way.

```
$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 $S (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 restore SYS1.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 SYS1.PARMLIB(JES2BACK), you should be able to bypass any errors you make updating JES2PARM.[76] To use JES2BACK in the AD system, stop JES2 and restart it with the command:

```
S JES2,HASPPARM=ALTPARM
```

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 JOBCLASSes, SYSOUT classes, and SYSOUT priorities defined, although the existing definitions do no harm.

For further reference please see *OS/390: JES2 Initialization and Tuning Reference.*

8.14 How to Work with MIH

{ZH26} This section will describe the "Missing Interrupt Handler" in your OS/390 System. You will need access to SYS1.PARMLIB to customize and adjust the MIH parameters. The IECIOSxx member is used by MIH, and you may need to add this member to SYS1.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 appropriate for LAN communications devices.

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

[76] 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 to fix whatever problem exists in JES2PARM.

The IECIOS 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 reIPL. 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 DVTHRSH=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 IEASYS00 (or IEASYSxx) PARMLIB member. This parameter should be added to IEASYS00. *(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. You would normally use 00 as the value for xx.

The IECIOSxx member is normally read only during IPL. If you change it after IPL, you can issue the:

```
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=(E22),TIME=00:00
```

The P/390 device managers for LAN (SNA, TCP/IP) emulated devices require the MIH time to be set to zero. These devices appear as channel-to-channel (3088) devices to OS/390; the AD system includes a range of 3088 addresses at E20-E27 and E40-E47. You would normally use some of these addresses to emulate 3172 control units for LAN (or WAN) connections. A PARMLIB(IECIOS00) member might be:

```
MIH TIME=00:00,DEV=(E20-E23)
```

with the corresponding IOS=00 entry in IEASYSxx.

The R6 AD system has MIH set correctly for E20 and E21.

If you add an IECIOSxx member, remember to insert IOS=xx lines in all the IEASYSxx members that might be used.

8.15 How to Provide More Paging Space

{ZH28} OS/390 can support a large amount 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   100%  FULL  0122  SYS1.PLPA.PAGE
COMMON  28%    OK  0122  SYS1.COMMON.PAGE
LOCAL   17%    OK  0122  SYS1.LOCAL1.PAGE
NO SWAP DATASETS ARE IN USE
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 ( 350 ) )
```

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 350 cylinders in the example, assumed to be on a 3380 volume, represents about 260 MB of additional virtual storage paging space.

After creating the new paging data set, you can tell OS/390 to use it 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 member IEASYSxx of 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.

8.16 How to Compress a PDS

{ZH29}

Partitioned Data Sets (PDSs, libraries) 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. When you first allocate a PDS, you must specify the fixed size of this initial directory internal file.

In general, you cannot directly update a member of a library.[77] If you edit a member, the altered version is written at the end of the library, 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.[78] 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. The easiest 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. Compressing a PDS that has been actively updated for some time can cause considerable disk activity and take some time.[79]

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.

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
/*
```

[77] This can be done by special programming, but such usage is not common.

[78] By lost, we mean that no directory entry points to it and there is no easy way to use that space.

[79] In principle, there is some risk during compress. 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.

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.[80] 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 user, and 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.

8.17 How to Allow More TSO Sessions

{ZH30} By default, the AD CD-ROM system permits only ten concurrent TSO sessions. There is no particular reason for this limit, and you can change it. You must change two elements:

1. Edit SYS1.PARMLIB(TSOKEY00). You will find USERMAX=10; change this to a larger number, such as 30.
2. Edit SYS1.VTAMLST(A0600). You will find stanzas named A06TSO01, A06TSO02, up to A06TSO10. Create more of these stanzas, named A06TSO11 through A06TSO30. 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.

You can stop TSO and VTAM and restart them, or reIPL the system, to have the increased number of TSO users. For example,

```
P TCAS              stop TSO
Z NET,QUICK         stop VTAM
   wait until VTAM stops completely
S VTAM              start VTAM
S TCAS              start TSO
```

This assumes your VTAM started procedure name is VTAM. To verify this, use the D A,L command at the MVS console and observe the VTAM name. Alternate

[80] You can use ISPF option 3.4, followed by an S line command, to see how much of its allocated spaced is being used any 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.

names are often VTAMLCL or VTAMIMS. Find the corresponding procedure in SYS1.PROCLIB and see it if has a DD statement for SYS1.VTAMLST. If not, you will need to update whatever data set it uses in place of VTAMLST.

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 SYS1.VTAMLST (or alternate). For the AD system, member A0TCP in SYS1.VTAMLST contains the matching LUs; you can increase the number by following the existing pattern.

8.18 How to Manage Dump Data Sets

{ZH32} 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.

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. Overviews of both methods are presented here.

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.

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 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 the system default name pattern is:

SYS1.DUMP.D&DATE..T&TIME..&SYSNAME..S&SEQ.

Where &DATE, &TIME, &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*.

If you use automatic dump data sets, the initialization commands are usually in SYS1.PARMLIB(COMMNDxx), although an operator can enter the commands directly at the OS/390 console. An installation can specify the name pattern to be used with the following operator command:

```
DUMPDS NAME=name-pattern
DD NAME=name-pattern
```

The use of the automatic facility can be controlled by the following operator command:

```
DUMPDS ALLOC=ACTIVE|INACTIVE
DD ALLOC=ACTIVE|INACTIVE
```

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 default name pattern you would use ispf option 3.4 with a Dsname level of SYS1.DUMP to list the data sets.

To determine the contents of each dump data set is more difficult. You must either correlate the date and time of the dump (a part of the dump data set name) with the messages in the system log issued when the dump was taken or go into IPCS and interrogate the dump.

When you have determined that a dump is not required, it can be cleared by deleting the data set using normal ISPF or JCL facilities.

The AD CD-ROM system uses the automatic allocation method for managing dump data sets, using the default name pattern. The dump data sets are allocated on volume SCPMV5. This is achieved with the following 3 lines of member COMMND00 in SYS1.PARMLIB.

```
COM='DUMPDS ADD,VOL=(SCPMV5)'
COM='DUMPDS NAME=SYS1.DUMP.D&DATE..T&TIME..&SYSNAME..S&SEQ'
COM='DUMPDS ALLOC=ACTIVE'
```

(In later releases of the AD CD-ROM, these commands might be moved to SYS1.PARMLIB(VTAMAPPL) instead of the COMMND00 member.) There are also 3 data sets allocated on SCPMV5 for use with the preallocated method of management (SYS1.DUMP00, 01 and 02). You can change to the preallocated method by deleting the last of the statements above or changing it to read:

```
COM='DUMPDS ALLOC=INACTIVE'
```

We believe the preallocated method may be better for new P/390 and and R/390 environments. Taking and interrogating system dumps should be an unusual situation and if the need arises the preallocated dump data sets can be easily managed. Automatically allocated dump data sets can proliferate very easily and because of their size easily fill the disk volume(s) assigned. If you decide to use the automatic method you must be willing to periodically check and cleanup your dump volumes.

Complete details for managing dump data sets can be found in *OS/390 System Commands*.

8.19 How to Use Job and Print Classes

{ZH34} 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 SYS1.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 new P/390, 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 R6 AD CD-ROM 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         Drained
 6              ABCD         Drained
 7              ABCD         Drained
 8              ABCD         Drained
 9              ABCD         Drained
 10             ABCD         Drained
 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. Twelve initiators is excessive for a P/390 system, but the drained initiators do no harm. Earlier systems provided several specific job classes with BLP authority. The R6 system does not do this. You may need to define (by editing JES2PARM) a specific job class that permits BLP.

You can use the following operator commands to work with initiators:

```
COMMAND             Purpose.................
  $DI               Display initiator status
  $PIE              Drain (stop) initiator E
  $SIB              Start initiator B
```

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 (A, B, C, .. 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) 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 AD CD-ROM has these definitions:

```
OUTCLASS      Characteristics....
  A           write, print
  B           write, punch
  C           write, print
  D           write, 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 R6 AD CD-ROM JES2PARM
```

A small P/390 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. Edit SYS1.ADCD06.PARMLIB(JES2PARM)
2. Use the ISPF editor CREATE JES2BACK command (with a count of 9999 in the prefix field of the first line data) to create a backup member of your current, working JES2PARM member.
3. Find the PRT(1) line, which is a comment line.
4. Remove the comment characters (defore and after the line).
5. Remove the FCB parameter.
6. 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.)
7. Change the unit to UNIT=00E (which is generated as a 1403 printer in the AD system.
8. Save the changed member.
9. Restart JES2. (The easiest way to do this is to shutdown the system (using the S SHUTDOWN command in the R6 AD system) and restart it.
10. When the system restarts, use the $SPRT1 command to start the JES2 printer.
11. Submit a job with SYSOUT=C (or whatever class(es) you specified for the printer).
12. The SYSOUT=C output should be sent to your server for printing.

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
..
//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 CD-ROM 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 class A if you want to print it.

For further reference please see *JES2 Initialization and Tuning Guide* and *JES2 Initialization and Tuning Reference*. See 8.13, "How to Change JES2 Parameters" on page 125 for comments about changing JES2PARM, and 15.3, "How to Print with a PC Laser Printer" on page 316 for information about printing with PC printers.

8.20 How to Obtain JOB ENDED Messages

{ZH36} 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 P/390 OS/390, you may find these messages useful.

Edit SYS1.PARMLIB(MPFLST00). Find the line with $HASP395, and delete that line. 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
```

8.21 How to Use TSO Edit

{ZH01} This section will describe briefly the process you need to follow to use 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:

```
EDIT 'yourid.tso.cntl(newmemb)' EMODE TEXT
```

The above command[81] can be entered from the TSO foreground panel, 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 27 on page 137.

[81] 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.

```
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 27. Sample Screen Using EDIT Command from TSO Native Session

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)
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*.

8.22 How to Use RMF (Resource Management Facility)

{ZJ02} RMF can be used to provide very detailed performance measurement data, a quick ′snapshot′ of performance can be obtained by using the DA command of the SDSF product.

RMF is composed of a number of components. These components provide different types of output, collect different data and are generally configured in different ways. The following provides an overview of these components and how they are handled in the AD system.

8.22.1.1 RMF Collection Session and Default Monitor I Session (ZZ)

This is the base component of RMF and must be active for all other components to work. It is started by an OS/390 operator command such as:

```
S RMF.RMF
```

If you expect to use RMF frequently, you might want to start it automaticaly with a line in SYS1.PARMLIB(COMMNDxx). The AD CD-ROM does this with:

```
COM=′S RMF.RMF,,,MEMBER(09)′
```

Data collection is controlled by SYS1.PARMLIB(ERBRMFxx), such as ERMRMF09 in the example above. Output can be written to SMF (if SMF recording is active, which it is not in the AD system), or used by interactive TSO functions described below. When used with a P/390 or R/390, RMF will not work with some of the finer I/O measurements that are obtained on a mainframe, because the P/390 and R/390 systems do not emulate various low-level I/O sampling functions.

To stop the RMF collection address space, use the command:

```
P RMF
```

8.22.1.2 RMF Monitor II Sessions

RMF Monitor II sessions provide a realtime snapshot of system performance. Many different views are available, both from a system perspective as well as from an individual address spaces (STC, JOB or TSO user). The RMF Collection session must be active to use Monitor II. It is accessible under ISPF with full ISPF PFKey scrolling support (via option M.8 in the AD CD-ROM system). It is available from TSO using command RMFMON, or from any ISPF command line using command TSO RMFMON. Within RMFMON use:

- PFKeys 1-12 for basic RMF II panels
- Use command M to return to the RMFMON menu
- Use command Z to terminate RMFMON

8.22.1.3 RMF Monitor III Sessions

Monitor III provides:

- Snapshot views of the system from the perspective of system throughput or system delay (RMF III is also called the Workload Delay Manager)
- Snapshot views of individual address spaces showing how resources are being used to perform work or detailed information on what is delaying the work for the address space
- A mechanism for collecting this information over the long term. This can be via in-storage buffers, RMF III VSAM data sets or SMF

The RMF III gatherer address space (RMFGAT) is required before Monitor III display sessions can be used. The following MVS operator command will start the gatherer session:

```
F RMF,S III
```

This will cause RMF to start the RMFGAT address space using the supplied defaults (member ERBRMF04 in SYS1.PARMLIB). Care should be taken using the gatherer because data is collected in large in-storage buffers and this may cause paging to increase in a busy system.

The Monitor III session is controlled by SYS1.PARMLIB member ERBRMFxx If you wish to collect data you must allocate the VSAM data sets and update ERBRMFxx. Full details are provided in the *RMF Users Guide*. Monitor III display sessions are accessible under ISPF with full ISPF PFKey scrolling support (option M.8 in the AD system). The RMF III workload manager panels can be also be accessed under ISPF from any command line using the command TSO RMFWDM.

To stop the Monitor III session, use the operator command:

```
F RMF,P III
```

8.22.1.4 RMF Post Processor Facility

This facility produces long term overview reports by using data produced by the online data collection components. This data must have been collected via SMF. The Post Processor is generally executed as a batch job and can read archived SMF data sets or online SMF collection data sets (but not the current in-use SMF collection data set). The AD system, as supplied, does not collect SMF data. See 8.1, “How to Collect SMF Data” on page 111 for details about enabling SMF collection. RMF and SMF will produce a large amount of data if turned on using the parameters supplied. You should review the RMF parameters in ERBRMFxx if you take this step to ensure you are collecting only data usefull to your purpose.

Full details and examples of the batch JCL and report control statements are provided in the *RMF Users Guide*. There is also a menu driven interface to produce the JCL and control statements under the RMF ISPF option.

8.22.1.5 RMF Spreadsheet Convertor

This facility allows RMF data output to be manipulated and converted for use with popular PC spreadsheet products. Full details are provided in the *RMF Users Guide*.

8.22.1.6 RMF Sysplex Data Gatherer

This facility allows RMF data output to be managed and viewed within a Sysplex Complex. This environment is not supported with the P/390 or R/390 systems but if you require more information, full details are provided in the *RMF Users Guide*.

8.23 How to Make a Trivial IDCAMS Job

{ZJ05} 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(MVSV5.MASTER.CATALOG)
/*
```

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[82], and this is seldom what you want when working with IDCAMS.

We suggest you create a small batch job, like the one shown, in one of your TSO libraries. You can overtype the command contained in the job and submit the job whenever you need to use IDCAMS functions.

8.24 How to View SYSOUT with ISPF

[82] This can be controlled by the PREFIX parameter of the PROFILE command, but changing this can be inconvenient and the current prefix in use is sometimes not obvious.

```
  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 28. ISPF - Outlist Utility Screen

{ZJ12} 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 28 .

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 29 on page 142 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).

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. However, if you run the job in IPO1.JCLLIB(TSOFF53), which installs an exit in TSO that modifies the security functions of the OUTPUT, STATUS, and CANCEL commands, and if you have OPER authority, then you can view the output for any job. See 8.11, “Understanding Basic SDSF Security” on page 121 for more information about OPER authority. (If you run this job, remember to refresh LLA afterward.)

```
                         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 29. ISPF - Foreground Print Options

8.25 How to Use HCD

{ZJ20} A step-by-step walkthrough using HCD to alter a device definition may be helpful.[83] It uses a particular example of redefining a tape drive, but the general process applies to most HCD usage.

The R6 version of the AD system uses SYS1.IODF00 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:

1. Begin with a working IODF, such as SYS1.IODF00.
2. 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.IODF00.
3. 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.IODF01.
4. Any IODF name ending with WORK is not a production IODF.

The progression of names is:

```
     SYS1.IODF00 --> SYS1.IODF00.WORK --> SYS1.IODF01 -->
     (production)                         (production)

 --> SYS1.IODF01.WORK --> SYS1.IODF02 --> etc
     (production)
```

[83] This same example appeared in *Printing with MVS on the IBM PC Server System/390*, and was originally provided by Gordon Chamberland, of Interprocess Systems, Inc.

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.

In the R6 AD system, SYS1.IODF00.WORK is already defined. However, we could not access it. You need to ???

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).

To alter your I/O definition file (SYS1.IODFxx), use the following steps:

1. Under ISPF, select option 6.
2. On the command line enter CBDCHCD. This will invoke the HCD facility. If HCD appears on your primary ISPF menu, you can invoke it from that menu. (In the AD system, HCD appears on the second page of the primary menu.)
3. Ensure that the IODF file name is set to ′SYS1.IODF00′ (or whatever is the name of your current production IODF). Select option 6 (maintain IODF files). (If you have already created new IODF files, this initial name may be different, of course.)
4. Select option 6 (upgrade IODF files)
5. For the R6 AD system, you will need to use the HCD delete option to delete ′SYS1.IODF01.WORK′. This data set is defined in the distributed system, but does not really exist. You must delete this definition before you can define it again.
6. Specify 1 (to create a new work IODF). Specify the new IODF name as ′SYS1.IODF01.WORK′ and specify the volume as SCPMV5. The step is complete when you see a message such as: ″Back-level IODF upgraded″. These first five steps assume that your IODF may be downlevel. These steps will upgrade (if needed) the IODF and create a new work IODF at the same time. (This is not necessary with every release, but it will do no harm.)
7. Use PF3 until you return to the main HCD menu.
8. Ensure that the new IODF name is specified: SYS1.IODF01.WORK[84] Select option 1 (Define, Modify, etc.).
9. Select option 5 (I/O devices).
10. 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.
11. Select option 2 (Change).
12. The next panel gives you the option to specify or revise several fields. Just press ENTER to continue.
13. On the next panel, place a slash (′/′) in the prefix area to select the CBIPO OS and press ENTER.
14. Select option 1 (Connect, Change)
15. On the next panel (Define Device Parameters / Features), change the setting for the COMPACT feature from NO to YES.
16. On the next panel (Assign/Unassign Device to Esoteric), note that a new generic will be shown (3480X). This is what you should specify in your JCL

[84] If SYS1.IODF01.WORK does not yet exist, specify SYS1.IODF01. When you start to modify it, HCD will offer to create a new work data set for you. At this time specify the name SYS1.IODF01.WORK. There is no requirement to create a work data set with the same base name as a production IODF, but most users do so.

(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.

17. Press PF3 until you return to the main HCD menu.
18. Select option 2 (Activate).
19. Select option 1 (Build new IODF). Set the work IODF name to 'SYS1.IODF01.WORK'. Set the production name to 'SYS1.IODF01'. Set the volume serial to SCPMV5. Select option 1 to use the work IODF in use at present. Press ENTER.
20. At the next panel (Specify or Revise), just press ENTER. Success is near when you see the message: "Production IODF SYS1.IODF01 created".
21. Press ENTER.
22. 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:
23. From the main ISPF menu select option 2 (Edit).
24. Set the data set name to 'SYS1.IPLPARM'.[85]
25. On the command line enter S LOADWS.
26. On the command line of the new (empty) edit member, enter COPY LOADW1. (The name LOADW1 is arbitrary, but it must be in the in the format LOADxx.)
27. Change the IODF parameter in column 10 from 00 to 01.
28. Save the new LOAD01 member, exit completely and shutdown MVS.
29. Invoke the P390 Configurator, normally by clicking on the Configurator icon.
30. On the Update System Environment screen (F4), change the load parameter from 0122WS to 0122W1.
31. Exit the P390 Configurator and save the new settings (F6).

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.

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.

8.26 How to Set Up an ISPF Administrative Environment

{ZJ25} 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[86] to these functions, and you can make the logon process for normal TSO users much faster by deleting

[85] This example assumes you are using SYS1.IPLPARM for your LOADxx members. If SYS1.IPLPARM does not exist, the system will look for these members in SYS1.PARMLIB. This example assumes you normally use LOADWS to start your system. You should modify the following instructions to reflect whatever LOAD parameter you normally use.

[86] By this, we mean access to the ISPF panels and functions provided by these products

unnecessary allocations, as described in 8.9, “How to Make TSO Logon Faster” on page 118.

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:

1. Use a different TSO logon procedure when you want a large number of preallocated functions, or
2. Use the normal (smaller) 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 it 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 AD system has three primary TSO logon procedures already defined.

1. IKJACCNT with emergency access to TSO, no ISPF environment.
2. ISPFPROC with full ISPF functions.
3. OMVSPROC With full ISPF functions and OpenEdition MVS.

IKJACCNT is too limited for normal usage. ISPFPROC allocates a large number of data sets and is suitable for an administrator. OMVSPROC is the same as ISPFPROC, but starts OMVS automatically. You can add your own logon procedure according to your users needs or you can set up a basic logon procedure for “normal” TSO users. An example of a basic logon procedure for normal TSO users is shown in 8.9, “How to Make TSO Logon Faster” on page 118.

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.)

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.[87] You can create this CLIST in SYS1.CLIST(MOREPWR), or in any other CLIST library you choose.

```
PROC 0 VOL(SCPMV5)
 CONTROL   MSG NOFLUSH
PROFILE NOMODE MSGID PROMPT INTERCOM WTPMSG
/* -------------------------------------------------  */
/*   Alloc clist for all functions   OS/390 R3 AD      */
/* -------------------------------------------------- */
FREE  F(SYSUADS)
FREE  F(SYSLBC)
FREE  F(SYSPROC)
FREE  F(SYSEXEC)
FREE  F(SYSHELP)
FREE  F(ISPLLIB)
FREE  F(ISPMLIB)
FREE  F(ISPTLIB)
FREE  F(ISPEXEC)
FREE  F(ISPPLIB)
FREE  F(DITPLIB)
FREE  F(ISPSLIB)
FREE  F(SYSTCPD)
FREE  F(SDSFMENU)
FREE  F(ISPTABL)
FREE  F(SMPTABL)
FREE  F(CIDTABL)
ALLOC F(SYSUADS) DA('SYS1.UADS') SHR
ALLOC F(SYSLBC) DA('SYS1.BRODCAST') SHR
ALLOC F(SYSPROC) DA('ISP.SISPCLIB',              -
           'SYS1.CLIST',                         -
           'SYS1.DGTCLIB',                       -
           'SYS1.HRFCLST',                       -
           'SYS1.ICEISPC',                       -
           'ICQ.ICQCCLIB',                       -
           'SYS1.SBLSCLI0',                      -
           'SYS1.SBPXEXEC',                      -
           'SYS1.SCBDCLST',                      -
```

[87] 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.

```
          'GIM.SGIMCLS0',                     -
          'SYS1.SERBCLS',                     -
          'CBC.SCBCUTL',                      -
          'MQM.V1R2M0.SCSQCLST',              -
          'EOY.SEOYCLIB',                     -
          'IOE.SIOEEXEC',                     -
          'ASU.SASUEXEC') SHR
ALLOC F(SYSEXEC) DA('ISP.SISPEXEC',           -
          'SOMMVS.SGOSREXX',                  -
          'DCE.SEUVEXEC',                     -
          'SYS1.SBPXEXEC',                    -
          'MQM.V1R2M0.SCSQEXEC',              -
          'IOE.SIOEEXEC',                     -
          'ASU.SASUEXEC') SHR
ALLOC F(SYSHELP) DA('SYS1.HELP',              -
          'ISP.SISPHELP',                     -
          'SYS1.SEDGHLP1',                    -
          'TCPIP.SEZAHELP') SHR
ALLOC F(ISPLLIB) DA('SYS1.ICEISPL') SHR
ALLOC F(ISPMLIB) DA('ISP.SISPMENU',           -
          'SYS1.DFQMLIB',                     -
          'SYS1.DGTMLIB',                     -
          'ICQ.ICQMLIB',                      -
          'SYS1.ICEISPM',                     -
          'SYS1.HRFMSG',                      -
          'SYS1.SBPXMENU',                    -
          'SYS1.SCBDMENU',                    -
          'SYS1.SEDGMENU',                    -
          'TCPIP.SEZAMENU',                   -
          'GIM.SGIMMENU',                     -
          'ISF.SISFMLIB',                     -
          'SYS1.SERBMENU',                    -
          'CBC.SCBCIMG',                      -
          'SOMMVS.SGOSMSGS',                  -
          'EOY.SEOYMENU',                     -
          'ASU.SASUMENU') SHR
ALLOC F(ISPEXEC) DA('ISP.SISPEXEC',           -
          'SYS1.SBPXEXEC',                    -
          'MQM.V1R2M0.SCSQEXEC',              -
          'EUV.SEUVEXEC') SHR
ALLOC F(ISPPLIB) DA('ISP.SISPPENU' ,          -
          'SYS1.LOCAL.ISPFPNLS',              -
          'SYS1.DFQPLIB',                     -
          'SYS1.DGTPLIB',                     -
          'ICQ.ICQPLIB',                      -
          'SYS1.ICEISPP',                     -
          'SYS1.HRFPANL',                     -
          'SYS1.SBLSPNL0',                    -
          'SYS1.SBPXPENU',                    -
          'SYS1.SCBDPENU',                    -
          'SYS1.SEDGPENU',                    -
          'MQM.V1R2M0.SCSQPNLE',              -
          'EUV.SEUVPNL',                      -
          'TCPIP.SEZANPNL',                   -
          'TCPIP.SEZAPENU',                   -
          'GIM.SGIMPENU',                     -
          'ISF.SISFPLIB',                     -
          'SYS1.SISTPNL0',                    -
          'SYS1.SISTPNL1',                    -
```

```
           'SYS1.SERBPENU',                            -
           'CBC.SCBCPNL',                              -
           'SOMMVS.SGOSPNLS',                          -
           'EOY.SEOYPENU',                             -
           'ASU.SASUPENU') SHR
ALLOC F(DITPLIB) DA('DIT.SDITPLIB') SHR
ALLOC F(ISPSLIB) DA('ISP.SISPSLIB',                    -
           'GIM.SGIMSENU',                             -
           'ISP.SISPSENU',                             -
           'ISF.SISFSLIB',                             -
           'SYS1.DGTSLIB',                             -
           'SYS1.HRFSKEL',                             -
           'SYS1.ICEISPS',                             -
           'SYS1.SBLSKEL0',                            -
           'SYS1.SBLSMSG0',                            -
           'MQM.V1R2M0.SCSQMSGE',                      -
           'MQM.V1R2M0.SCSQSKL',                       -
           'SYS1.SISTMSG0',                            -
           'CBC.SCBCSKL',                              -
           'SOMMVS.SGOSSKEL') SHR
ALLOC F(ISPTLIB) DA('ISP.SISPTENU',                    -
           'SYS1.DGTTLIB',                             -
           'SYS1.ICQTLIB',                             -
           'ICQ.ICQTABLS',                             -
           'SYS1.ICEISPT',                             -
           'ASU.SASUTLIB',                             -
           'SYS1.SBLSTBL0',                            -
           'SYS1.SBPXTENU',                            -
           'SYS1.SCBDTENU',                            -
           'GIM.SGIMTENU',                             -
           'MQM.V1R2M0.SCSQTBLE',                      -
           'ISF.SISFTLIB',                             -
           'SYS1.SERBT',                               -
           'SYS1.SERBTENU',                            -
           'EOY.SEOYTENU',                             -
           'SYS1.SMP.OTABLES') SHR
ALLOC F(SYSTCPD) DA('SYS1.TCPIPMVS.TCPIP.DATA') SHR  <===may change
ALLOC F(SDSFMENU) DA('ISF.SISFPLIB') SHR
ALLOC F(ISPTABL) DA('SYS1.SMP.OTABLES') SHR
ALLOC F(SMPTABL) DA('SYS1.SMP.OTABLES') SHR
ALLOC F(CIDTABL) DA('SYS1.CIDTABL') SHR
```

The above CLIST is executed by entering this TSO command:

```
EXEC 'SYS1.CLIST(MOREPWR)'
```

assuming the CLIST was stored as SYS1.CLIST(MOREPWR).[88]

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 logon procedure the user is already using. When making this CLIST, be especially careful with the continuation marks.

You might want to make another CLIST to FREE the datasets allocated by the allocation CLIST, but this is seldom necessary.

88 This CLIST is not provided by the AD system. You need to make your own.

8.27 How to Interpret System Data Set Names

{ZJ27} The logon procedures and CLISTs described in 8.9, "How to Make TSO Logon Faster" on page 118 and 8.26, "How to Set Up an ISPF Administrative Environment" on page 144 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.....       ISPF libraries for SDSF functions
....DGT.....       ISPF libraries for ISMF functions
EOY.SEOY....       ISPF libraries for BookManager functions
....HRF.....       ISPF libraries for RACF functions
SYS1.ICE....       ISPF libraries for interactive Sort functions
ICQ.ICQ.....       ISPF libraries for TSO/E extended functions
....SBLS....       ISPF libraries for IPCS functions
SYS1.SBP....       ISPF libraries for OMVS functions
SYS1.SCBD...       ISPF libraries for HCD functions
....SEDG....       ISPF libraries for DFSMS RMM (removable media)
GIM.GIM.....       ISPF libraries for CBIPO functions
....SERB....       ISPF libraries for RMF functions
CBC.SCBC....       ISPF libraries for C and C++ functions
ASU.SASU....       ISPF libraries for DCE functions
DFS.SIOE....       ISPF libraries for DCE/DFS functions
SOMMVS.SGOS....    ISPF libraries for SOM functions
IEL.SIEL....       ISPF libraries for PL/1 functions
TCPIP.....         ISPF libraries for 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:

```
ASMA........       High Level Assembler
ASMT........       High Level Assembler Toolkit
ABJ.........       CCCA
ANF.........       IP Printway
AOP.........       (printer interface for ISPF)
API.........       NetSpool
APK.........       (for AFP printing)
BFS.........       LanServer
CSF.........       Cryptographic APIs and tools
CEE.........       Language Environment (LE) data sets
DIT.........       DITTO
EDC.........       C/370 compiler
EMW.........       NetQuestion
EQW.........       Cooperative Development
EUV.........       DCE
EWX.........       LANRES
EZM.........       NetView
FFST........       FFST (Service Tool)
GLD.........       LDAP
ICA.........       Firewall
ICE.........       OS/390 Sort
IEL.........       PL/1
IGY.........       COBOL for MVS & VM
IOA.........       OSA/SF
IOE.........       DFS (DCE)
```

```
MQM.........     MQ Series
```

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.

8.28 How to Backup and Restore OS/390

{ZJ29} There are a number of aspects to consider for disk backup planning:

- Disk hardware errors. P/390 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 (SYS1.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:
  1. You cannot IPL the system, and must use standalone or OS/2 functions to recover the system to where it can be IPLed.
  2. 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 emulated volume every night -- especially using the standard 4mm tape drive.
- 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, although the *preconfigured system* (a smaller OS/390 CD-ROM system) could be used as a recovery system, or you could construct a minimal system yourself. Having a second system makes recovery of a non-IPLable system much easier.

You have more backup options with a P/390 than with a "real" mainframe, because you can use OS/390 backup facilities or OS/2 backup facilities (or AIX backup facilities on an R/390). 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:SCPMV5.122  E:SCPMV5.BAK
  ```

  This will take several minutes, because the files are large, but is very simple to do. You can recover the SCPMV5 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.
- You can use a compression product to do the same thing; for example, use one of the PKZIP products. 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 a standard OS/2 tape backup product, such as Novaback or Sytos Premium. These are well known, well documented products widely used to backup PC disks. If you normally use your 4mm tape drive under OS/390, and want to also use it with one of these OS/2 tape backup products, you must change CONFIG.SYS and reboot your system.
- You can use XTAPE, a product from Interprocess Systems, Inc., that performs OS/2 backups to tape using the same device manager that can be used for P/390 functions -- avoiding the CONFIG.SYS change.

The disadvantage of OS/2-based 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 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 P/390 OS/390 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 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 irregardless 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 a P/390 environment.

A major consideration when deciding on backup/restore strategy is the hardware available. The following devices are good candidates for creating backup medium:

- All systems incorporate a 3.5inch 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 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.

8.29 How to Continue TSO Command Lines

{ZJ30} 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 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
```

8.30 How to Specify a TSO Region Size

{ZJ32} Specifying a region size[89] can be confusing since both 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.

In the case of typical P/390 development environments, there is usually no reason to have 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.

8.31 How to Copy an ADRDSSU Dump

{ZJ34} 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

[89] 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.

by common utility programs.[90] 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 ("square tape") to a 4mm tape on a P/390 system.

8.32 How to Set Your TSO Prefix

{ZJ36} 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 you prefix as the high-level qualifier (HLQ) of any data set that you name *unless you include 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.

90 The advantage in doing this is that more data fill fit on a tape and the average transfer rate of a dump or restore job is higher.

8.33 How to Escape from a Command or Program

{ZJ38} 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.

8.34 How to Use VIO

{ZJ40} 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 128MB or 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 esoteric[91] 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).

[91] Do not let this word confuse you. Esoteric device names are names that are not hardware device names. For example, UNIT=SYSDA uses an esoteric name, while UNIT=3380 uses a hardware device name (or *generic* name).

- 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 (emulated) disk. The allocation log, printed at the beginning of most batch jobs, will indicate whether VIO was used.

8.35 How to Add a Program to OS/390

{ZJ42} 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.

You normally place a load module (the executable form of a program) in a load library by either of these methods:

1. 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)
   ```

2. 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.[92] 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. See 8.51, "How to Use LLA (Library LookAside)" on page 169 for more information about LLA.

8.36 How to Start a JCL Library

{ZJ44} 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.

[92] They must be link edited with AC=1 and explicitly use authorized functions if they are to take advantage of this potential.

```
 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 30. Submitting a Job while Editing the Data Set

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:

1. 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 JOBCARD).
2. 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:

1. One or more DD (data definition) statements, to identify and describe the input and output data to be used in the step.

Before creating a job, you need to know:

1. 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 you installation.
2. 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.
3. The job to be done and the resources needed. You need to determine what work you plan to have OS/390 perform:
   a. What programs and data sets you will use.
   b. What DDnames are used by your application.
   c. 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 for viewing.
4. How to view and understand held output. Running your job will produce three types of held output:
   a. System messages (JES and MVS)

b. Your JCL code with procedures expanded, overrides applied, and symbolics resolved.
c. 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
- 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 Figure 30 on page 157 for an example; 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 8.32, "How to Set Your TSO Prefix" on page 154 for a discussion of your PREFIX.

8.37 How to Use ALLOC and FREE

{ZJ46} 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 all) of the parameters that can be specified through JCL. Either traditional MVS data sets or HFS files can be allocated.

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 completely unallocates a data set. The batch equivalent of FREE is the end of a job step.

If you create a new data set with the ALLOC command, 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
```

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 8.32, "How to Set Your TSO Prefix" on page 154 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 by:

```
ALLOCATE DA('ID.NEW1.DATA') SPACE(10,5) TRACKS LRECL(80) -
BLKSIZE(8000) RECFM(F,B) 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.
- 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. When you free SYSOUT data sets you can change their output class to make them available for processing by an output writer or to route them to other user. UNALLOC is the alias name for the FREE command.

You can allocate a new data set with the attributes of an existing model data set by typing:

```
ALLOCATE DA('ID.COPIED.DATA') LIKE('USERID.MODEL.DATA')
```

You can allocate a new data set with different space allocation and the same DCB of an existing model data set by typing:

```
ALLOCATE DA('ID.NEW2.DATA') SPACE(10,5) TRACKS LIKE('ID.MODEL.DATA')
```

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/USERID/OUTPUT.DBP')      +
      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)
```

Free of all dynamically allocated files, attribute lists, and data sets:.

```
FREE ALL
```

8.38 How to Execute a CLIST or REXX Script

{ZJ48}

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 Figure 31 on page 161 illustrates the required syntax:

```
  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 31. ISPF Command Shell Panel with Example CLIST/REXX Execution

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 Figure 31

```
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 called 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 32

```
  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 32. ISPMF Command Shell Panel with Direct CLIST/REXX Execution

Alternately direct from an ISPF command line as:

```
COMMAND ===> tso clistxyz
```

or in native TSO (READY mode):

```
clistxyz
```

8.39 How to Transfer Data between OS/2 and OS/390

{JZ50}

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.[93] 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 or in ISPF option 6 before starting a transfer. Any other status will probably cause a failure.
- 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
or   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 pulldown 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).

Another method of transferring data is via ftp, assuming OS/390 TCP/IP is operational.

[93] We assume you are using CM/2 or a similar emulator that supports the clipboard cut and paste functions.

8.40 How to Select BLKSIZE for Load Libraries

{ZJ51} 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 sizes[94] may cause slightly slower system operation when loading the 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.

8.41 Should I Set Up and Use OS/390 BookManager/READER?

{ZJ52} The BookManager/Reader that is distributed with the AD system is not customized and ready for use. The required DD statements are in the default ISPF logon procedure, but several data sets need to be added to LPA and LNKLST.

We find that most users prefer to use the OS/390 documentation available on CD ROM, using the OS/2 or Windows versions of the reader programs.

8.42 What are BPXAS Jobs?

{ZJ53} The MVS console frequently shows BPXAS being started. What is this?

These are address spaces used for *fork* processing in OpenEdition. 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.

[94] Many older load libraries have BLKSIZE=6144. This size was tuned for 2314 disks many years ago, and has remained common ever since -- mostly because few users understood how the linkage editor handled block sizes.

8.43 How to List System Variables

{ZJ54} The following Rexx clist provides an example of using 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: SYSTERMID = ' SYSVAR('SYSTERMID')
X = SYSCPUS('CPUS.')
SAY 'CPU SERIAL = ' CPUS.1
/* SOME OTHER POTENTIAL SYSTEM VARIABLES:                    */
/* SYSNAME SYSMVS SYSOPSYS SYSDFP SYSSMS 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.

8.44 How to Write To stdout From Assembler

{ZJ55} *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.

8.45 How to Find Your TSO Logon ID

{ZJ56} The following C program will display your TSO logon id. 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 EDCCLG,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 OE, 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.

8.46 How to Make C/C++ Compiles Faster

{ZJ57} One technique for improving compile time is to ensure all header files use large block sizes. Most headers stored in MVS data sets have LRECL=80,RECFM=FB. The BLKSIZE should be half track, if possible. For a 3390, this would be BLKSIZE=27920.

Large applications, with many, many embedded header files might experience substantial improvements in compile speed if their header files currently have block sizes considerably smaller than half track. (The header files supplied with the current AD systems have half-track blocking.)

Another factor in C compilation time is the location of the LE run time library, CEE.SCEERUN or SYS1.SCEERUN. This should be in LPA and there should be no STEPLIB statement in the compile step pointing to it. (The R6 AD system follows these suggestions.)

8.47 How to Expand a VTOC Size

{ZJ58} When you initialize a DASD volume for OS/390 use, you must specify a VTOC size. Many examples use 14 tracks.[95] 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.

You can use DSF 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(130) 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 130 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).

8.48 How to Determine If New Instructions Are Available

{ZJ59} A number of new instructions and instruction sets have been added to the S/390 architecture during the last several years. Not all S/390 processors have all instructions. During the IPL process, OS/390 sets indicators for these various instructions and instruction sets.

CVTSCPIN points to an area known as the SCCB. This area is mapped by the IHASCCB macro in SYS1.MACLIB. There are many bit indicators in this mapping; you should inspect the macro source for more information. The R6 system does not include mapping for bits that indicate the presence of the immediate-and-relative, the extended length (MVCLE and CLCLE) instructions, and the BSA instruction. If you find the SCCBPLO equate in the macro, you can extend it as follows:

```
x'40'       SCCBPLO  The PLO instruction is present
x'10'                Immediate and relative instructions present
x'08'                MVCLE and CLCLE present
x'04'                BSA present
```

These last three bit indicators are not defined programming interfaces, so you would use them at your own risk.

This topic is of possible interest only to assembly programmers. Users of high level languages have no direct control over the instructions generated by the

[95] 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.

compilers. Assembler programmers may elect to test these bits (and many other bits mapped by IHASCCB) if they wish to use the "new" instructions. In our opinion, this is not a reasonable course for anyone developing production software --- unless the program has *very* unusual requirements.

8.49 How to Repair System With TSO Down

{ZJ60} 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 (including R/390s and the current Integrated Servers) can emulate a card reader. Input for the reader comes from OS/2 (or AIX). 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 configurator as part of the definition for a 2540 device. This is described elsewhere; you can read the AWS2540.DOC file for more information.

In order to use this function, you need to have a card reader defined to JES2. The current AD systems (R5 and R6) do not have a card reader defined in JES2. (In our example, you cannot change the JES2PARM definitions in PARMLIB, because you cannot log onto TSO to do it.)

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 JES2PARM 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 $S to start JES2 processing. You may need a $SRDR1 command to start the reader. After JES2 is finished initializing, you can use OS2 (or AIX) to copy your job to the correct OS/2 (AIX) 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 (or AIX) 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.

8.50 How to Understand Condition Codes

{ZJ61} 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 operand is true, then the current step (the one containing the COND) is skipped.

The first example can be read "bypass this step if 4 is greater 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.

8.51 How to Use LLA (Library LookAside)

{ZJ62} 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 pagable. 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. (This command is undocumented and probably unsupported. Do not use any additional operands with it. 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* and *Reference*. 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.

8.52 How to Place SCEERUN and Old COBOL Libraries

{ZJ63} 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 R6 AD CD-ROM has SCEERUN in the LNKLST. However, the installed COBOL compiler (in the R6 AD system) is an older compiler and you will need to make adjustments to obtain the proper libraries. See 2.17, "Minor Adjustments" on page 23 for specific information for the AD R6 system.

8.53 How to Select the Number of Page Data Sets

{ZJ64} There are two issues here: (1) performance, and (2) capacity. There are always at least three paging data sets: (1) PLPA, (2) COMMON, (3) everything else. See 2.9, "Paging Data Sets" on page 14 for a description of the paging data sets on the distributed AD CD-ROM system. See 8.15, "How to Provide More Paging Space" on page 128 for instructions about adding more paging data sets.

Conventional wisdom, for a small MVS 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.[96] However, informal observation of P/390s (and R/390s, and probably Integrated Servers) indicate that there is little, if any, gain from spreading out the paging data sets -- especially if RAID 5 disks are used on the underlying server.

These systems, using emulated CKD volumes, are not very suitable for high paging-rate situations. Again, informal observations indicate that sustained paging rates over 10-20 per second are "high." In principle, using a number of dedicated minivolumes for paging data sets should be beneficial, but we do not have any measurements to prove this.[97]

The capacity or amount of paging space (regardless of how many paging data sets are used to hold this amount) has two areas of concern. There must be sufficient capacity for your normal usage. For example, we needed to add an additional paging data set of 250 MB to the AD CD-ROM when we had eight students compiling and running Java and DB2, with several Web servers running. You can use the MVS command `D ASM` to observe how full your paging data sets are at any given time.

There is a second capacity issue: if a program enters a loop obtaining storage (malloc or GETMAIN loop), it can potentially obtain almost 2 GB of virtual storage. If it alters this storage, it may be paged out and may fill your page data sets. In a development environment, this will probably happen at one time or another. You can reduce the likelihood of it happening by using reasonable REGION sizes for jobs (or MAXASSIZE for OpenEdition). Thus a safer paging

[96] This is the well-known requirement for *high-volume, low-utilization data sets.*

[97] If you have good MVS data in this area, please contact the author!

capacity is your normal requirement plus 2 GB. P/390 owners are seldom willing to devote this much disk space for paging, since it is much more than is needed for normal situations. This means that developers may encounter situations where the system runs out of paging space.

It is possible to add paging data sets while the system is running, but only if sufficient free space is available on the defined volumes. (If you create a new CKD volume on a P/390, you must restart the P/390 subsystem and IPL MVS in order to use it.) If OS/390 runs out of paging space, the operator may be able to recover the situation but it is possible that the system will crash. This situation is unlikely on a production system, due to reasonable REGION sizes, ample paging space, and better tested programs.

8.54 How to Control SuperZAP

{ZJ65} 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:

1. Copy the AC=1 version (the distributed version) to a private, authorized library that is protected with UACC=NONE, and
2. 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. SYS1.MIGLIB is authorized by virtue of being in LINKLST and LINKAUTH=LINKLST is specified in the AD system.

8.55 How to Use ISRDDN

{ZJ66} 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.ADCD06.PARMLIB[98] and SYS1.PARMLIB and they are concatenated in that order. If member IEASYSCS exists in both PARMLIBs, it is the one in

[98] Release 5 uses SYS1.ADCD05.PARMLIB, and so forth

SYS1.ADCD06.PARMLIB that is used, because SYS1.ADCD06.PARMLIB is first in the concatenation. If you want to update member VATLSTAA in PARMLIB, you must first look in SYS1.ADCD06.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.ADCD06.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 up to four concatenated PDSs.* If there are more than four 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 four PDSs.

8.56 How to Edit An Alias

{ZJ67} 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.[99]

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:

[99] 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.)

```
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 TTR[100] 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.[101]

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.[102] 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 products? We rather strongly suggest the answer is NO, unless you have a specific need for them.

8.57 How to Select the LE Runtime Level Needed

{ZJ68} Different C compilers levels require different LE runtime library levels. The following table may help:

100 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 pointer to the member.

101 Furthermore, this situation continues to exist after the PDS is compressed.

102 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.

```
Year  Product            RTL required

1988  C/370 V1R1         C/370 V1R1, V1R2, V2R1, any LE level
1989  C/370 V1R2         C/370 V1R2, V2R1, any LE level
1991  C/370 V2R1         C/370 V2R1, any LE level
1991  AD C/370 V1R1      any LE level
1994  AD C/370 V1R2      C/370 V2R1, LE V1R3 or later
1995  C/C++ MVS V3R1     LE V1R4 or later
1995  C/C++ MVS V3R2     LE V1R5 or later
1996  OS/390 R1 C/C++    OS/390 R1 LE or later
1996  OS/390 R2 C/C++    OS/390 R2 LE or later
1997  OS/390 R3 C/C++    OS/390 R3 LE or later
1997  OS/390 V2R4 C/C++  OS/390 R4 LE or later
1998  OS/390 V2R5 C/C++  OS/390 R5 LE or later
1998  OS/390 V2R6 C/C++  OS/390 R6 LE or later
```

You must be concerned with both the LE link-edit libraries and the execution-time LE libraries. You must link edit with LE libraries that are lower or equal to the run-time libraries.

For example, I can compile with the OS/390 R1 C compiler, link edit with the OS/390 R2 LE libraries, and execute with the OS/390 R3 execution-time libraries.

The equivalent information for COBOL is:

```
Year  Product              RTL required
1984  OS/VS COBOL R2.4**   OS/VS COBOL, VS COBOL II, or any LE
1988  VS COBOL II R3.0**   VS COBOL II or any LE
1992  VS COBOL II R4.0     VS COBOL II or any LE
1991  COBOL/370 R1**       any LE
1995  COBOL for MVS&VM R2  LE V1R5 or later  (aka COBOL/370 R2)
1997  COBOL for MVS&VM V2R1 OS/390 R3 LE or lager

**no longer supported
```

8.58 What Are Disk Size Limits?

{ZJ69} Except for 3390-9 volumes (which are not emulated by P/390 or R/390 systems, but which could be attached via an Escon adapter in an Integrated Server), DASD volumes do not have over 65536 tracks. A few limitations are:

- The VTOC must reside within the first 65536 tracks
- Older types of data sets cannot be over 65536 tracks. (They may extend over five volumes, but the total cannot exceed this number.)
- Newer types of data sets (VSAM, PDSE) can extend past 65536 tracks
- Regular VSAM data sets cannot be over 4GB. Expanded VSAM data sets can be over 4GB.
- HFS files cannot extend beyond one volume (as of V2R6), or beyond 2GB.

Chapter 9. Open Edition

9.1 How to Clear 3270 Screen in C Program

{ZK10} 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");

system("tput clear");
```

The second method is obviously not portable.

9.2 How to Set and Display Extended Attributes

{ZK11} 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.[103] 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 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,

103 In addition to this flag, the program must have been linked with AC=1 in order to be APF authorized.

detgid, 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. (The HFS value for TYPE depends on the details in BPXPRMxx. For the AD system, the value is HFS.)

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  BPX.FILEATTR.*       (see if profile exists)
RDEF   FACILITY BPX.FILEATTR.* UACC(NONE)
PERMIT BPX.FILEATTR.*  CLASS(FACILITY) ID(goodguy) ACC(ALTER)
SETROPTS RECLIST(FACILITY) REFRESH
```

The RLIST is to determine if the BPX.FILEATTR.* profile already exists. If it does not, use the indicated RDEF command to create it. A BPX.FILEATTR.* profile includes authority for setting both authorized program indicators and program controlled file indicators. It is possible to use a more precise profile name to control each of these attributes separately, but we expect the single profile shown here will be more useful.

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 or moved (**cp** or **mv** commands), 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.

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.

9.3 How to Assign a Default Printer for the lp Command

{ZK12} 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 $HOME/.profile. This works if you are not using the OS/390 Print Server.

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.

9.4 How to Execute an Authorized Program

{ZK14} To exist as an authorized program in a library, a module must be:

1. Link edited with the AC=1 parameter.
2. Placed in an authorized library (as defined in PARMLIB) *or* was placed in an HFS file that has 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 9.2, "How to Set and Display Extended Attributes" on page 177 for a discussion of extended attributes.

An OE user can execute an authorized program that is stored in an MVS data set. See 9.19, "How to Use the Sticky Bit" on page 185 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:

1. 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,
2. 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 program[104]

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,[105], 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.

9.5 How to Create an MVS Load Module From OE

{ZK15} 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.)

104 Of course, the userid involved must have authority to execute programs from the library containing the load module, or from the HFS file containing the executable. Once in execution, the program might use SAF/RACF APIs to determine of the userid under which the program is executing has the required access to various resources.

105 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.

9.6 How to Use the at Command

{ZK16} 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.

9.7 How to Use NOHUP To Start a Daemon

{ZK17} 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 parents 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
```

9.8 How to Use the TSO Command from OE

{ZK18} 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.

9.9 How to Source a Script (Dot Command)

{ZK19} 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.

9.10 Understanding MAXZSSIZE

{ZK20} MVS batch and TSO address spaces work with REGION parameters (on JOB and EXEC statements, in TSO the TSO segment) 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 40 MB.

Changing MAXASSIZE in BPXPRMxx changes the maximum address space size for all OpenEdition users.

Batch jobs run under BPXBATCH are limited by MAXASSIZE, and not by any REGION size specified in their JOB or EXEC statements.

9.11 How to Restart inetd

{ZK21} 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.

9.12 How to Reclaim Disk Space in an HFS

{ZK22} 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.[106] 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.

9.13 How to Determine Size of Loaded Program

{ZK23} 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.

9.14 How to Prevent User From Using OE

{ZK24} 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)
```

9.15 How to Estimate SMF Output From OMVS

{ZK25} 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:

[106] The author has not found any documentation about this and is only reporting comments from other users.

```
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 _PBX_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 the *Initialization and Tuning Reference* for more information about SMFPRMxx controls.

9.16 How to Move an Executable from HFS to LPA

{ZK26} 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 require 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.

9.17 How to Execute AMODE24 Programs from OE

{ZK27} 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.)

9.18 How to Restart the Web Server

{ZK28} You can perform a quick restart of the Domino Go Web Server (or its follow-on replacements) with the following MVS operator command:

```
F IMWEBSRV,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.

9.19 How to Use the Sticky Bit

{ZK29} If the sticky bit is on (in an HFS file) and you execute this file, the system will:

1. Truncate the name to eight characters,
2. Drop any suffix (such as .so or .exe),
3. Change it to upper case,
4. 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),
5. 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 bee 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.)

9.20 Understand CR, LF, NL in ASCII and EBCDIC

{ZK30} 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.)

9.21 What MVS Data Sets Can Move to OE via OPUT?

{ZK31} 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.

9.22 What is the Largest File Size in HFS?

{ZK32} 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*2.8GB = 165GB.

With the same release, the largest file size is approximately 2**47 (if sparse).

9.23 How to Begin Using dbx

{ZK33} The AD CD-ROM R6 system, as delivered, cannot use dbx. We expect this same situation will exist in the R7, R8, and possibly later systems. 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 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, dbx will indicate that any program you attempt to test under dbx will branch to location zero and ABEND with 0C4.

Chapter 10. SMP/E

{ZL00} 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 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.

10.1 How to Learn 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.) In this context, the word *system* applies to practically any software you have installed, and is not limited to the *system* or *operating system*.

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 lever language equivalents) are assembled or compiled together to produce an OBJECT module. One or more object modules are linkedited 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.[107] 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.

Look at the basic processing commands you need to use SMP/E:

1. 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 *SMP/E R8.1 Reference*.

2. 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.

3. 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.

4. 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.

5. 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.

6. 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,

[107] The AD CD-ROM systems are created using a standard ServerPac.

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.

7. 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 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.

8. 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[108]. 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.

10.2 How to Display PTF Status

{ZL30} 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 contains[109] something like this:

[108] GG24-3481 was written to give directions over MVS Software Management, as OS/390 has similar functions than MVS has, you can make reference to it to obtain all information you need to manage SMP/E

[109] The screen images illustrated in this section do not reflect all the contents of the actual panels. A subset is illustrated here for brevity.

```
                         SMP/E PRIMARY OPTION MENU

===> 3

       1  ADMINISTRATION
       2  SYSMOD MANAGEMENT
       3  QUERY
       4  COMMAND GENERATION
       5  RECEIVE

SMPCSI DATA SET ===> 'mvs5.global.csi'

Generate DD statement ===> yes
```

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:

```
                         QUERY SELECTION MENU

===> 2

       1  CSI QUERY
       2  CROSS-ZONE QUERY
       3  SOURCEID QUERY
```

You would select option 2 (CROSS-ZONE QUERY). The next screen should be:

```
                         CROSS-ZONE QUERY

===>

       ENTRY TYPE ===> sysmod

       ENTRY NAME ===> hbb6601
```

Enter the PTF number on this panel. The ENTRY TYPE will usually be sysmod. This should produce the screen:

```
                  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
```

This screen idicates 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.

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. In this case, the installer *should* use a middle qualifier of GLOBAL, but this is not an absolute requirement.

Chapter 11. OS/390 Operator Functions

{ZN00} MVS has a strong concept of a system operator (or operators). An operator works through an MVS console. On a typical P/390 installation, this is one 3270 window on the primary PC display. On a P/390, as on any MVS system, it is possible to define and use multiple MVS consoles[110].

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.

11.1 Basic Operator Commands

{ZN01} Many of us are a little rusty when remembering OS/390 console commands. The following material is not intended to introduce you to operating an MVS system. It is merely a reminder for the proper syntax of a few of the common MVS (and JES2) commands. The commands listed here are issued from an MVS console, not a TSO terminal.

R n,NOREQ Normal JES2 startup reply
K E,D Erase bottom display area on console
K A,NONE Undefine the display area at bottom of screen
K E,1 Delete line 1 (top of screen, usually)
K E,SEG Delete contents of message segment
K N,PFK=(5,CMD=′S GTF,580′),CON=N Define a PFK
K S,DEL=W Place console in wrap mode (like JES3)
K S,REF Display current console mode
K S,DEL=RD Set "roll delete" mode (most common mode)
D U,,,380,2 Display two devices, starting at address 380
D U,VOL=volser
D U,DASD,ALLOC,C00,16
D U,DASD,ONLINE Display all online DASD volumes
D U,DASD,ONLINE,C00,8
DS P,C00 Display (newer command) device at address C00
DS P,580,8,ON Display 8 online devices starting with 580
DS S,C05 Display SMS-controlled device C05
D ASM,ALL Display all paging data sets
D C,K Display list of Control command operands
D DUMP Status of dump data sets
D A,L List active jobs and users
D OMVS Display summary of OpenMVS status information
D PFK Display PFK definitions
D PROG,APF Display libraries in APF list
D R,L List unanswered messages or actions
D SMS or D SMS.OPTIONS Display SMS status
D T Display time and date
C U=userid Cancel TSO user
DD CLEAR,DSN=ALL Clear all dump data sets
FORCE jobname Last resort

110 Also known as Master Consoles and/or MCS Consoles

M C09,VOL=(SL,volser),USE=STORAGE Typical mount command
SE ′message′ Message to all TSO users
SETPROG APF,FORMAT=DYNAMIC Change to dynamic APF
SETPROG APF,ADD,DSNAME=my.lib.load,VOLUME=volser Add library
V 583,ONLINE,UNCOND Vary device online
W t Write system log to class t
$A Release all held jobs
$A ′jobname′ Release specific job. Quotes required.
$C ′jobname′ Cancel job or user
$C PRTn Cancel job on printer
$DA,ALL Status of all JES2 functions
$D ′jobname′ Display JES status of job or user
$D SPOOL Display utilization of spool space
$D U,PRTS Display all printers
$E ′jobname′ Restart job after it completes
$H A or $H ′jobname′ Hold jobs
$I PRTn Interrupt printing and return job to queue
$PI3 Stop an initiator
$P ′jobname′ Purge a job (including spooled output)
$SI3 or $SPRT2 Start an initiator or printer
$T I3,C=AB Assign job classes for an initiator
$T PRT3,Q=AT Change output classes for printer
V NET,ACT,ID=xxx Activate new VTAMLST member or other node
V NET,INACT,ID=xxx Deactivate node
Z NET,QUICK Stop VTAM

11.2 How to Reply to Console Messages

{ZN03} 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,'xxxxx'
```

where 02 is the reply number and xxxx is your reply to the message. After JES2 is operational, you can use shorter forms of reply, such as:

```
2xxxx        (cannot use if JES3 modules are installed)
2,xxxxx      (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.

The R5 and R6 AD CD-ROM systems have JES3 modules installed, so you need to use the abbreviated reply with the comma in it.

11.3 How to Shut Down OS/390

{ZN05} There is not standard way to shut down OS/390. In the “old days,” shutdown usually consisted of issuing the following commands from the MVS master console:

```
1. Stop TCAS   (P TCAS)
2. Stop VTAM   (Z NET,QUICK)
3. Wait a few seconds, and D A,ALL to determine what is still
   running.  Issue appropriate CANCEL commands.
4. Stop JES2   ($PJES2)
5. Write MVS statistics  (Z EOD)
```

Current systems are much more complex, with DB2, OE daemons, TCP/IP, Web Servers, WLM and so forth. With the AD system on a P/390 one approach -- one that offends more traditional systems programmers -- is to have everyone log off TSO, check to ensure that no batch jobs are running, and then crash the system (by clicking on the STOP P/390 icon). Other than a possible message about JES2 waiting for shared DASD when it is restarted (and which forces you to make one more operator reply during the next IPL), we have had no serious side effects from stopping OS/390 this way. However, we recognize that this is not practical in a Sysplex environment and is not a recommended procedure in any environment.

(We do not recommend crashing OS/2. It is easier to shutdown normally, and a proper shutdown avoids all the chkdsk delays when it is rebooted.)

See 2.7, “Shutting Down the System” on page 12 for a description of a shutdown script that is provided with the AD R6 system. This appears to be an excellent script. Reading it (and using it) makes a good point: you can use a script that contains stop/cancel commands for programs that are not running. MVS will simply issue an error message and wait for the next command.

These “scripts” depend on the VTAMAPPL application program. Despite its name, this program has nothing to do with VTAM. It simply reads a data set of MVS commands (with an additional command, PAUSE nn, to permit a defined period to be introduced between other commands) and issues them to MVS. The input to VTAMAPPL is known as a script. The AD system uses VTAMAPPL scripts during IPL (as an extension of COMMNDxx PARMLIB members) and for a shutdown script.

We do not know the status of the VTAMAPPL program (whether it is a “standard” or “supported”) program. It is included with recent ServerPac systems (including the AD system).

11.4 Common VTAM Commands

{ZN10} 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:

- VTAM Display commands.

```
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           */
```

- VTAM Halt commands.

```
Z NET,CANCEL      /* VTAM abnormal termination command      */
Z NET,QUICK       /* VTAM normal termination command        */
```

- VTAM Modify commands. The modify commands are intended for experienced VTAM systems programmers. They are probably not needed for basic use of VTAM functions, and examples are not given here.
- VTAM Vary commands.

```
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                  */
```

11.5 How to Manage Spool Space

{ZN15} 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 will 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 8.11, “Understanding Basic SDSF Security” on page 121
- 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 ie 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 ie. Display Active)

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:

- **$D′xxxxxxx′** displays status and job_id of all jobs/STC/TSO with the name xxxxxxx. The single quote marks around the job name are required.
- **$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:

```
$PJmmmm-nnnn
```

where mmmm is the number of the first job to cancelled and nnnn is the last job number. 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 $S 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 4.5, "How to Work With PARMLIB" on page 41 for details on safely modifying SYS1.PARMLIB).

```
$TA,T=00.10,'$OQ,Q=OTX,CANCEL,A=8'
$TA,T=00.15,'$PQ,Q=AJ6,A=8'
$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.

11.6 How to Hot Start JES2

{ZN20} 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.[111] 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)
```

[111] 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.*

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.

11.7 How to Identify Data Set User

{ZN22} 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:

```
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 obtain a “Data Set In Use” message, you can press PF1 twice to get a list of the enqueued owners.

11.8 How to Display Basic IPL Parameters

{ZN23} After most of the IPL activity completes, the command:

```
D IPLINFO
```

will display the most basic parameters for this IPL, including the LOADxx member used and the IODF used.

11.9 How to Recover the MVS Console

{ZN24} MVS does not like to lose its master console. Recovery from console problems is somewhat unusual.

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 or CM/2 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.

11.10 Capturing SVC Dumps

{ZN25} "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 systems 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, that 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 dataspaces 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 datasets at the same time.

Two local paging datasets do not make a robust paging subsystem.[112] For typical OS/390 systems, we usually recommend a minimum of four local page datasets; 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.[113]

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 2000M (a good sized DB2 dump)
- The average transfer rate to the page dataset is 2.5M per second

then performance has these characteristics:

- With two local page datasets 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 datasets 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 datasets 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 datasets 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 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 it complete a new dump request will be accepted, assuming that DUMPSRV is not currently sitting with captured dumps at its MAXSPACE value

112 However, it may be for a P/390-based 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.

113 Again, note that these comments do not apply well to a P/390-based system, since other constraints limit DASD performance and bandwidth.

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.

Chapter 12. VTAM Functions

{ZP00} VTAM administration tends to be regarded as an activity apart from other MVS administration. One reason is that there are usually VTAM specialists who are separate from the main body of systems programmers in many larger installations.

12.1 How to Work with VTAMLST

{ZP01} 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'
```

There is a procedure in SYS1.PROCLIB(VTAM) to start VTAM. It usually looks something like this:

```
//VTAM   PROC
//VTAM    EXEC PGM=ISTINM01,REGION=6000K,
//             DPRTY=(15,15),TIME=1440,PERFORM=8
//VTAMLST   DD DSN=SYS1.VTAMLST,DISP=SHR
//VTAMLIB   DD DSN=SYS1.LOCAL.VTAMLIB,DISP=SHR
//          DD DSN=SYS1.VTAMLIB,DISP=SHR
//SYSABEND  DD SYSOUT=*,HOLD=YES
```

When VTAM is started (as a result of executing the procedure in SYS1.PROCLIB), VTAM reads SYS1.VTAMLST(ATCSTR00) (or ATCSTRxx if another LIST number is specified in the start command). This 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.VTAMLST for the AD system:

```
SSCPID=06,NOPROMPT,                                              X
CONFIG=00,MAXSUBA=31,SUPP=NOSUP,                                 X
SSCPNAME=USS3270,                                                X
NETID=P390,                                                      X
HOSTSA=6,                                                        X
CRPLBUF=(208,,15,,1,16),                                         X
IOBUF=(100,128,19,,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.

VTAM also reads SYS1.VTAMLST(ATCCONxx) (usually ATCCON00) and automatically starts all the "major nodes" (members in SYS1.VTAMLST) that are listed there. A typical ATCCON00 member is:

```
A0600,NSNA70X,NSNA90X,DYNMODEL,etc
```

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 ISTEXCUV IS ACTIVE
IST093I ISTDSWMN ACTIVE
IST984I USER EXIT ISTEXCSD IS ACTIVE
IST093I A0600 ACTIVE
IST984I USER EXIT ISTEXCGR 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.[114] 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
```

[114] There is nothing special about this name. It was selected years ago as part of an example.

```
*
A06CICS2 APPL  EAS=160,               EST CONCURRENT SESSIONS    *
               ACBNAME=CICS2,         APPLID FOR ACB             *
               AUTH=(ACQ,BLOCK,PASS)  CICS ACQUIRE & PASS TMLS
*                                     CICS REQUEST BLOCKED INPUT
*
A06IMS   APPL  EAS=160,               EST CONCURRENT SESSIONS    *
               ACBNAME=IMS,           APPLID SESSIONS            *
               AUTH=(ACQ,BLOCK,PASS)  IMS ACQUIRE & PASS TMLS
*
A06JES2  APPL  EAS=1,                 EST CONCURRENT SESSIONS    *
               ACBNAME=JES2,          APPLID FOR ACB             *
               AUTH=(ACQ,PASS)        JES2 ACQUIRE TERMINALS
*
A06GDDM  APPL  AUTH=(ACQ,PASS),       GDDM CAN AQUIRE/PASS       X
               ACBNAME=ADMPRINT,      APPLID FOR ACB ADMPRINT    X
               EAS=5
*
A06TSO   APPL  AUTH=(NOACQ,NOBLOCK,PASS,NOTCAM,NVPACE,TSO,NOPO), X
               EAS=1,ACBNAME=TSO
*
A06TSO01 APPL  AUTH=(NOACQ,NOBLOCK,PASS,NOTCAM,NVPACE,TSO,NOPO), X
               EAS=1,ACBNAME=TSO0001
*
A06TSO02 APPL  AUTH=(NOACQ,NOBLOCK,PASS,NOTCAM,NVPACE,TSO,NOPO), X
               EAS=1,ACBNAME=TSO0002
*
     .
     .
A06TSO10 APPL  AUTH=(NOACQ,NOBLOCK,PASS,NOTCAM,NVPACE,TSO,NOPO), X
               EAS=1,ACBNAME=TSO0010
*
A06DSPRT APPL  AUTH=(ACQ),ACBNAME=DSPRT
*
MVSLU01 APPL   ACBNAME=MVSLU01,       ACBNAME FOR APPC           C
               APPC=YES,                                         C
               AUTOSES=0,                                        C
               DDRAINL=NALLOW,                                   C
               DLOGMOD=APPCHOST,                                 C
               DMINWNL=5,                                        C
               DMINWNR=5,                                        C
               DRESPL=NALLOW,                                    C
               DSESLIM=10,                                       C
               LMDENT=19,                                        C
               MODETAB=APPCTAB,                                  C
               PARSESS=YES,                                      C
               SECACPT=CONV,                                     C
               SRBEXIT=YES,                                      C
               VPACING=1
```

Notice that there are ten TSO definitions (not all shown in the example). If you want more then ten concurrent TSO sessions, you need to increase the number of definitions, using the existing definitions as a pattern. Other applications, such as CICS and IMS, do not have one definition per user terminal.

VTAM is very particular about line format. It uses a rigid assembly-language format. In particular, a continuation character for a line must be placed in column 72 and the continued line must begin in or before column 16. Do not forget the commas when using a continuation line. Errors may not be detected, and may cause a variety of strange problems.

In the AD system, member NSNA70X contains 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 this member 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
```

Member NSNA90X contains more local 3270 definitions, no different than the NSNA70X definitions. P/390 users sometimes associate the 9xx terminals with LAN-attached users, but there is no requirement to do this.

12.2 How to Stop and Start VTAM

{ZP02} 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 ISTPDILU NODE NOW INACTIVE
IST105I ISTRTPMN NODE NOW INACTIVE
IST105I ISTDSWMN NODE NOW INACTIVE
IST105I LOCAL900 NODE NOW INACTIVE
IST105I LOCAL700 NODE NOW INACTIVE
IST105I A0600 NODE NOW INACTIVE
IST105I ISTGROUP NODE NOW INACTIVE
IST412I VTAM COMMAND PROCESSING TERMINATED
IST984I USER EXIT ISTEXCUV IS INACTIVE
IST984I USER EXIT ISTEXCSD IS INACTIVE
IST984I USER EXIT ISTEXCGR 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
```

The above 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
```

You should have MASTER console authority to perform MVS FORCE commands and you should use always the ARM parameter with this command to avoid system disruptions. The ARM parameter does normal task termination routines without causing address space destruction. After using FORCE you might have to reIPL the system.

If you are using an alternate startup member in SYS1.VTAMLST, you may need to include a MEMBER=(xx) parameter, but most systems do not use this. After starting VTAM, wait for the message:

```
IST020I VTAM INITIALIZATION COMPLETE FOR VnRm
```

before starting TCAS or other applications.

12.3 How to Define a New VTAM Application or Terminal

{ZP03} 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 SYS1.VTAMLST. For a new application, you could add stanzas to SYS1.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 SYS1.VTAMLST(TCICS) containing:

```
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 SYS1.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:

```
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.

12.4 How to Create a VTAM Logo

{ZP04} The VTAM logon is what you see on a 3270 screen before logging into an application. It is created by a module sometimes named USSTAB.[115] 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 SYS1.VTAMLST in the terminal definitions. For the AD R6 system, look in SYS1.VTAMLST(NSNA70X), for example.

The module should be placed in SYS1.VTAMLIB (not SYS1.VTAMLST). In the AD system, it is placed in SYS1.LOCAL.VTAMLIB, which is concatenated to SYS1.VTAMLIB. The source code is in SYS1.P390.SOURCE in members USSTAB, USSN, and USSS. The easiest way to change the logo is to modify this source code, reassemble it, and try it. The JCL can be found in SYS1.P390.CNTL in members JSSN and JSSS.

```
//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).  (Older version.  Not R6)
//*
//S0       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))
```

115 There are two modules involved, one for non-SNA terminals and one for SNA terminals. This section describes the non-SNA terminal module.

```
//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  *
********************************************************
*** SMAPLE USSTABLE FOR P390.                        ***
********************************************************
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'            PPPPPP    /     333333  9999999  0000000              '
 DC  CL80'           P    P    /         3  9     9  0     0                '
 DC  CL80'          PPPPPP    /      33333  9999999  0     0                '
 DC  CL80'         P         /          3        9  0     0                 '
 DC  CL80'        P         /     333333        9  0000000                  '
 DC  CL80'                                                                  '
 DC  CL80'                PC SERVER 500 S/390                               '
 DC  CL80'                                                                  '
 DC  CL80'                                                                  '
 DC  CL80'    THIS SCREEN IS WRITTEN USING SYS1.LOCAL.VTAMLIB(USSTAB).'
 DC  CL80'    SOURCE FOR CHANGING IT IS IN SYS1.P390.CNTL(USSTAB).    '
 DC  CL80'    THE DEFAULT VTAM USSTAB IS IN SYS1.VTAMLIB(USSTAB).     '
 DC  CL80'                                                                  '
 DC  CL80'        MVS/ESA SP VERSION 5 SYSTEM WITH JES2                     '
 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,VOL=SER=SCPMV5,UNIT=3380
//SYSUT1   DD UNIT=SYSALLDA,SPACE=(6160,(230,760))
//OBJECT   DD DSN=*.S0.SYSLIN,VOL=REF=*.S0.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.

12.5 How to Connect a Simple SDLC 3174

{ZP05} Configuring an SNA SDLC connection is somewhat complex, and the background information provided here may make the process more meaningful.

A major change took place in communications support with Version 2.1 of the P/390 support programs. Three new device managers for communications were added:

1. 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.
2. 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.
3. 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.[116]. 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.

From a usage point of view, LAN3172 is the same as the older AWS3172 and will not be discussed here.

WAN3172 is a new function for P/390 OS/390. 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

[116] This accounts for many messages and panels that say "3172" instead of "WAN3172" or "LAN3172"

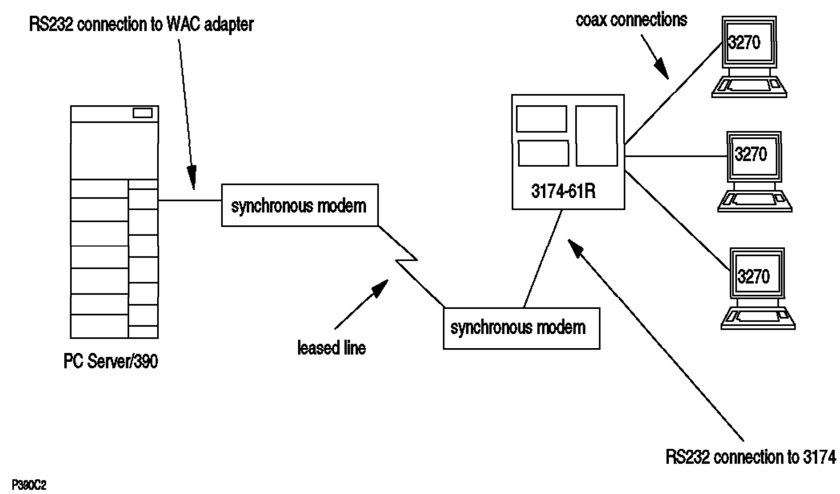


Figure 33. Trial SDLC Configuration. Note that synchronous modems are needed; typical PC modems are asynchronous and cannot be used for a basic SDLC connection. The 3174 is a widely-used control unit for 3270-family terminals. The 3174-61R is a remote version, used with up to eight terminals. Our unit was at microcode level C2.0.

WAN3172 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 SYS1.VTAMLST parameters than would be used with real SDLC connections.

WAN3172 can use two different hardware adapters:

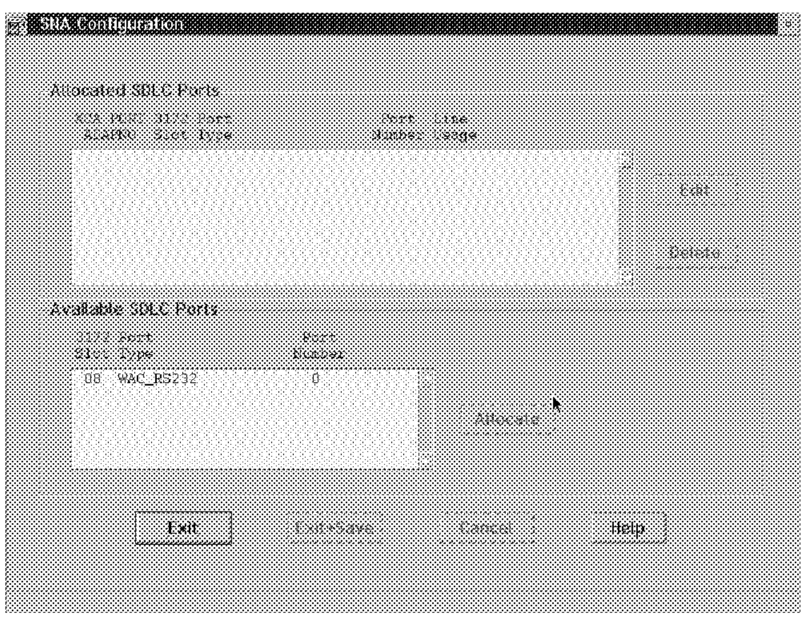


Figure 34. Basic SDLC Configuration Panel. The normal usage is to click on an adapter or line in the lower window, and then click on allocate. This will move that line to the upper window. Click on it there, and then click on Edit to customize that line.

- 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 ("ARTIC" or "RTIC") 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.)

12.5.1.1 Installing the Device Managers

{ZP05B} All the files and setup required for WAN3172 are not automatically installed when the P/390 support programs are installed. See *IBM PC Server 500 System/390 Installation, Configuration, and User's Guide for MVS/ESA*, IBM order number SA22-7210-01, or the redbook *P/390 and R/390 OS/390 Introduction*, order number SG24-2538-01, for installation details.

12.5.1.2 Configuring the Device Managers

{ZP05C} *IBM PC Server 500 System/390 Installation, Configuration, and User's Guide for MVS/ESA* (IBM order number SA22-7210) provides information about device manager setup, and the DOC files with the P/390 support diskettes 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. The 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.) For our example, described here, 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.

```
>E20 >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 34 on page 213. In this panel, you must:

1. 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 on *Allocate.*
2. This should produce the display shown in Figure 35 on page 215. For our purposes, we selected the following options:
   - Role: PRIMARY
   - Dial: NO
   - DATMODE: HALF
   - NRZI: YES
   - Maximum Link Stations: 1
   - Maximum Data Size: 521

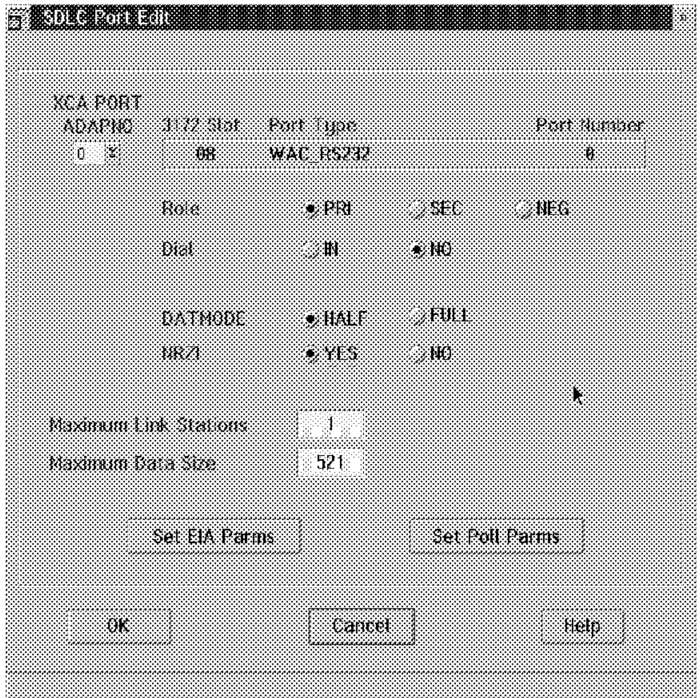


Figure 35. SDLC Port Edit Panel. This panel is entered by clicking on Edit in the basic SDLC Configuration panel.

3. After setting these parameters, click on *Set EIA Parms*. This should produce the panel shown in Figure 36 on page 215. In this panel, set:
   - Clocking: DTE
   - Line Speed: 19200
   - RTS: CONSTANT

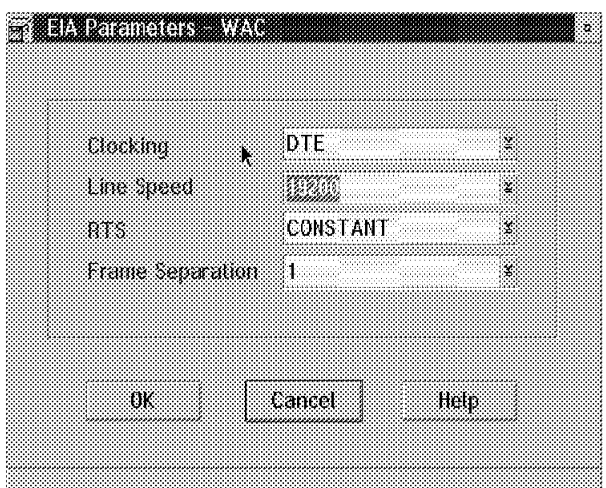


Figure 36. EIA Parameters Panel

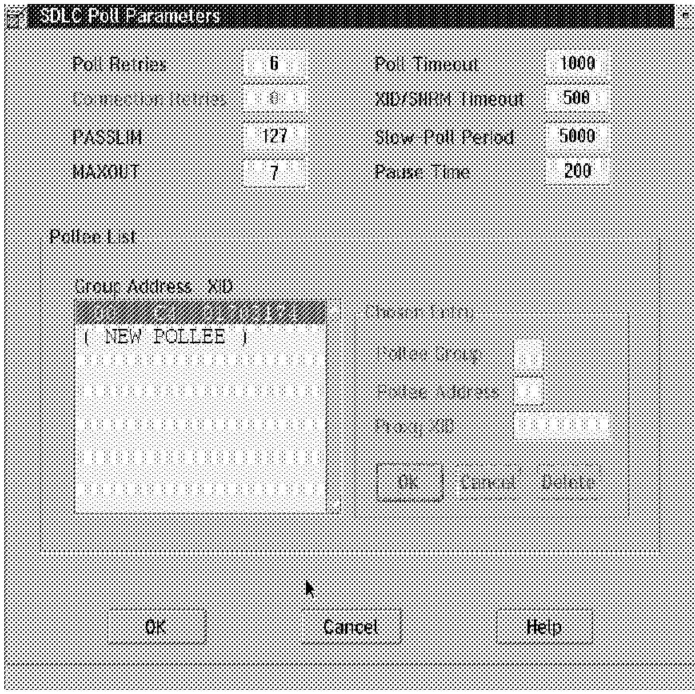


Figure 37. SDLC Port Parameters Panel. The parameters we used are somewhat arbitrary, but are common for remote SDLC 3174 control units. Double-click on NEW POLLEE to activate the Chosen Entry section of the panel. This will add a new line to the Pollee List subwindow.

- Frame Separation: 1

4. Click on *OK* to return to the previous panel, and then click on *Set Poll Parms* to obtain the display shown in Figure 37. 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 on *NEW POLLEE*. This will activate the Chosen Entry section. In this section, set:

   - Pollee Group: 00
   - Pollee Address: C4
   - Proxy XID: 01703174

   Click *OK* and then click on another *OK* again to exit from this panel.

5. 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=E20
```

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=E20,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 MVS 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 SYS1.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 remote 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.”[117]

[117] 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 upper-case letters. The system should respond with ENTER USERID -, and you would enter a TSO userid and continue normal operation.

12.6 How to Use the P/390 SNA Monitor

{ZP06} 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 subwindows 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 subwindow. 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.

12.7 How to Set Up NJE

{ZP10} This setup will enable two P/390 MVS 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.

1. 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.
```

2. 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 (in an OS/2) window:

   ```
   C> 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
   ```

   Another way to obtain the MAC address is to use the P/390 configurator option “Update LT Sessions” (F7). The MAC address is shown at the bottom of the panel produced by this function.

   You will need the MAC addresses of both systems later, when defining VTAM parameters.

3. Re-boot OS/2. OS/2 console messages should indicate that LAN connection is effective. Note that the same LAN adapter can be used for NetBios connections and for TCP/IP.[118]

4. 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.)

5. 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
   ```

6. Edit a number of SYS1.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.

[118] But only for one TCP/IP, which might be OS/2 TCP/IP or OS/390 TCP/IP, but not both.

```
RAID: SYS1.VTAMLST(PATH):

          PATH DESTSA=6,ER0=(6,1),VR0=0

P390: SYS1.VTAMLST(PATH):

          PATH DESTSA=5,ER0=(5,1),VR0=0
```

7. The CDRM member defines a VTAM to VTAM connection.

```
P390 and RAID: SYS1.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
```

8. 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: SYS1.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: SYS1.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: SYS1.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
```

9. The TYPE=XCA member defines the LAN station (the emulated 3172) to VTAM, and defines connections (LINEs and PUs) using the 3172.

RAID: SYS1.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: SYS1.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)
```

10. The ATCSTRxx member contains global VTAM parameters for a system. It is where the local subarea and SSCPNAME is defined.

RAID: SYS1.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: SYS1.VTAMLST(ATCSTR00)

```
      (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)
```

11. The ATCCONxx member defines which VTAMLST members should be activated automatically when VTAM starts.

RAID: SYS1.VTAMLST(ATCCON00)

```
      add A05LAN,PATH,CDRM to the list
```

P390: SYS1.VTAMLST(ATCCON00)

```
      add A06LAN,PATH,CDRM to the list
```

12. 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, etc.)
d net,majnodes
d net,stations
d net,e,id=xxxx       for more detail
```

13. On both, copy SYS1.PARMLIB(JES2PARM) to SYS1.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.

14. Edit SYS1.PARMLIB(JES2PARM) to include the following lines. We placed these lines near the beginning, where there is a comment about NJE.

```
RAID: SYS1.PARMLIB(JES2PARM)

     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: SYS1.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
```

15. 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.

16. 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.

Chapter 13. TCP/IP

{ZMOO} This chapter addresses TCP/IP setup and customization within the context of the OS/390 AD CD-ROM V2R5 and V2R6 systems, and a closely-related preloaded system available for IBM Application StarterPak systems. The AD CD-ROM is intended for use with P/390, R/390, and the IBM Integrated Server 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 Redbooks:

- *OS/390 eNetwork Communications Server V2R5 TCP/IP Implementation Guide Volume 1 - Configuration and Routing* (IBM order number SG24-5227)
- *OS/390 eNetwork Communications Server V2R5 TCP/IP Volume 2 - OpenEdition Applications* (IBM order number SG24-5228)
- *OS/390 eNetwork Communications Server V2R5 TCP/IP Volume 3 - MVS Applications* (IBM order number SG24-5229)

The systems based on P/390 adapters have servers underlying the S/390 processors. A P/390 system is built by adding a P/390 adapter to a PC Server, for example. 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, 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 for each of the platforms described. 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:

1. Understanding and selecting IP addresses. (Discussed in 13.1, "How to Assign IP Addresses" on page 224.)
2. Installing hardware, software, and customization of the underlying server. The details are different for each of the hardware platforms involved. (Discussed in several sections beginning with 14.2, "Hardware and Server Setup for TCP/IP" on page 256.)
3. Setup for TCP/IP on the server (OS/2 or AIX). (See 14.3, "How to Prepare a P/390 for TCP/IP" on page 257 and following sections.)
4. Basic OS/390 TCP/IP setup. This is common for all platforms, and is limited to making tn3270 and ftp[119] operational. (See 13.2, "How to Provide Basic OS/390 TCP/IP Customization" on page 225.)

[119] Basic utility functions, such as *ping* and *netstat* are also included.

5. A discussion of additional customization for OS/390 TCP/IP. This would be common for all platforms, and will be will be presented at an overview level.

13.1 How to Assign IP Addresses

{ZM99} 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:

- 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 R/390 or Integrated Server installation will be similar. An Application StarterPak is different because there is no underlying OS/2 or AIX; it might have only one LAN interface for OS/390 TCP/IP.
- Where there is more than one interface (typical P/390, R/390, Integrated Server), we will assume both 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.

13.2 How to Provide Basic OS/390 TCP/IP Customization

{ZM01} The setup for an underlying server (P/390, R/390, IBM Integrated Server) 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 Domino Go 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.

After completing your server setup (or setting the desired OSA-2 protocols, in the case of the Application StarterPak), you should IPL OS/390. Your first step should be to verify the state of the Missing Interrupt Handler for the emulated 3172 addresses and verify that the interfaces are online.

13.2.1 Missing Interrupt Handler

{ZM01A} You need to disable the OS/390 Missing Interrupt Handler (MIH) for the addresses associated with the emulated 3172 used by TCP/IP. This is done by adding member IECIOSxx to SYS1.PARMLIB.[120] 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).[121] 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.

120 The member name used is normally IECIOS00, but can have any suffix; the name follows normal PARMLIB rules.

121 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.

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  O
0E21 CTC  O
```

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.[122]

13.2.2 RACF and TCP/IP

The AD CD-ROM should already have the necessary RACF[123] 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.

13.2.3 Procedures and Data Sets

{ZM10C} 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 eNetwork Communications Server: IP Configuration* (IBM order SC31-8513, or later document). For our purposes, we will assume you are using the AD CD-ROM system and the data sets/files it provides.

---

[122] 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."

[123] A security product other than RACF may be used. In such cases, you must adapt TCP/IP security requirements to your security product.

The starting point for TCP/IP customization is the JCL for its started procedure. In the OS/390 V2R6 AD CD-ROM, this is in SYS1.PROCLIB(TCPIP):

```
//TCPIP PROC PARMS='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 SYS1.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.[124] The data set names associated with these DD statements have changed from release to release. The names shown above were used for the OS/390 V2R5 AD CD-ROM system. 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.[125]

Assuming the TCPIP started task described here, your initial OS/390 TCP/IP customization involves a few lines in the PROFILE data set.

13.2.4 TCP/IP Profile

{ZM01D} The complete PROFILE data set, as used in the author's V2R5 system is listed here. Comment lines have been removed for brevity:

```
ARPAGE 20
DATASETPREFIX TCPIP
TELNETPARMS
    PORT 23
    INACTIVE 600
    TIMEMARK 600
    SCANINTERVAL 120
    SMFINIT STD
    SMFTERM STD
    WLMCLUSTERNAME TN3270E ENDWLMCLUSTERNAME
ENDTELNETPARMS
TCPCONFIG        RESTRICTLOWPORTS
UDPCONFIG        RESTRICTLOWPORTS
AUTOLOG 5
    FTPD JOBNAME FTPD1   ; FTP Server
    PORTMAP              ; Portmap Server
ENDAUTOLOG
PORT
     7 UDP MISCSERV            ; Miscellaneous Server
```

124 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.

125 This statement assumes that the started task name is TCPIP. This should remain standard, starting with V2R5, 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.

```
       7 TCP MISCSERV
       9 UDP MISCSERV
       9 TCP MISCSERV
      19 UDP MISCSERV
      19 TCP MISCSERV
      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
     3000 TCP CICSTCP             ; CICS Socket
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
; BSDROUTINGPARMS false
;    ETH1      1500      0     255.255.255.0   0
; ENDBSDROUTINGPARMS
TRANSLATE
ITRACE OFF
ASSORTEDPARMS
   FWD
  RESTRICTLOWPORTS
ENDASSORTEDPARMS
BEGINVTAM
  TELNETDEVICE 3278-3-E NSX32703 ; 32 line screen
  TELNETDEVICE 3279-3-E NSX32703 ; 32 line screen
  TELNETDEVICE 3278-4-E NSX32704 ; 48 line screen
  TELNETDEVICE 3279-4-E NSX32704 ; 48 line screen
  TELNETDEVICE 3278-5-E NSX32705 ; 132 column screen
  TELNETDEVICE 3279-5-E NSX32705 ; 132 column screen
  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
  ENDDEFAULTLUS
  LUSESSIONPEND
  DEFAULTAPPL TSO
  LINEMODEAPPL TSO
  ALLOWAPPL TSO* DISCONNECTABLE
  ALLOWAPPL *
ENDVTAM
START LCS1
```

13.2.5 Basic Adapter Customization

{ZM01E} 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 R/390 or Application StarterPak) with a single Ethernet adapter that will use S/390 addresses E20 and E21. The adapter is connected via the LCS3172 device manager (P/390, R/390) or is an OSA-2 port (Application StarterPac).

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 S/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.[126] The third parameter is the LAN adapter number; this is the number from the Current Configuration window in MPTS.[127] The fourth parameter is a device name -- the same name you used in a DEVICE statement earlier.

[126] 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.)

[127] For an R/390, use the number corresponding to the AIX adapter name; ent3 would be 3, for example. For a StarterPak, use 0 or 1, corresponding to the port on the ENTR adapter.

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.[128]).

13.2.6 Static Routing Customization

{ZM01F} The GATEWAY statement(s) provide static routing. You might later remove these statements if you want dynamic routing. For your initial customization, we suggest you use static routing. The format of the gateway statement is unusual and is best explained with examples. We have added a comment line that helps understand 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.[129] 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.[130]

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.[131] To subnet a class B address to a class C subnet, the mask would be 0.0.255.0.

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)
```

[128] You can have multiple IP addresses with a single interface, but this is beyond the scope of this document

[129] 127.0.0.1 is automatically used as the loopback address; do not define it here.

[130] See an earlier discussion about appropriate packet sizes for Ethernet. We suggest 1492 as a starting size.

[131] Subnet masks of 255 are commonly used. You might use other values provided you produce a valid net/host address definition.

```
              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[132] are to be sent to the system at 9.12.17.50. 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.[133] 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.

132 In this example, routes for networks 9.12.17 and 193.1.2 are specified in previous statements.

133 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.

13.2.7 Other PROFILE Customization

{ZM01G} 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 20 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.[134]
- 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.[135]
- 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 eNetwork Communications Server: IO Configuration*

[134] OMVS, OE, OpenEdition, and UNIX System Services are all synonyms, as used here.

[135] 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.

(IBM order number SC31-8513). We strongly suggest you obtain a copy of this manual.

13.2.8 Other Basic Customization

{ZM01Z} 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).

13.2.9 The DATA Data Set and /etc/resolv.conf

{ZM01H} Our TCP/IP started task JCL, shown in 13.2.3, "Procedures and Data Sets" on page 226, 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:

```
keep 10.
         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
         ALWAYSWTO YES
```

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.[136] Our /etc/resolv.conf looked like this:

136 If it exists. If the /etc/resolv 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.

```
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.) *You should keep your DATA data set and your /etc/resolv.conf file synchronized.*[137]

In both cases, 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.

13.3 FTP Customization

{ZM02} 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
```

Note that the SYSFTPD DD statement is commented out. This causes a search for several default names.[138] One default name, assuming that DatasetPrefix (in the DATA data set) is "TCPIP") is TCPIP.FTP.DATA and this data set exists in the AD CD-ROM system. It appears as follows (with many comments removed):

[137] This dual structure is a result of earlier implementations of several different TCP/IP "stacks" and several sockets programming interfaces.

[138] We cannot think of any particular reason why the developers chose to do this instead of coding the DSNAME.

```
;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 halftrack blocking), PRIMARY (change to something larger than one track) and INACTIVE (change to a larger timeout number, such as 1200) parameters. The possible parameters are explained in the *IP Configuration* manual (SC31-8513).

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* 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.

13.3.1 inetd Customization

{ZM03} 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.[139]

```
//INETD  PROC
//INETD  EXEC  PGM=INETD,REGION=4096K,TIME=NOLIMIT,
//             PARM='POSIX(ON) ALL31(ON)/ /etc/inetd.conf'
//* Note the required lower case name above
//SYSPRINT DD SYSOUT=*
//SYSERR   DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSIN    DD DUMMY
```

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/otelnetd otelnetd -lm
logon    stream  tcp  nowait  OMVSKERN  /usr/sbin/rlogind rlogind -m
```

(The presence of the rlogin daemon in /etc/inetd.conf does not mean that rlogin is completely installed.) 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.

[139] Early shipments of the V2R5 AD CD-ROM had a minor customization error that prevented starting inetd from the rc file.

13.3.2 Local Host Name Customization

{ZM04} If you do not use a network 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 DATASETPREFIX 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 was 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=SCPMV5 and UNIT=3380. 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* manual contains information about /etc/hosts and the HOSTS data set in the chapter *Configuring the Site Table*. The TSO command TESTSITE is provided to try the results of MAKESITE.

13.3.3 TCP/IP Services Definitions

{ZM05} 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 V2R5 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          nicname
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         uucpd           # uucp daemon
remotefs        556/tcp         rfs_server rfs  # Brunhoff remote filesystem
telnet          623/tcp
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
```

The same system had the following data set for TCPIP.ETC.SERVICES. The contents are in lower case, 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          nicname
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
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
#     Kerberos services
klogin          543/tcp           # Kerberos aythenticated rlogin
kerberos        750/udp      kdc  # Kerberos aythentication--udp
kerberos        750/tcp      kdc  # Kerberos aythentication--tcp
kerboros_master 751/udp           # Kerberos aythentication
kerberos_master 751/tcp           # Kerberos aythentication
passwd_server   752/udp           # Kerberos passwd server
userreg_server  753/tcp           # Kerberos userreg serrver
kpop            1109/tcp          # Pop with Kerberos
knetd         2053/tcp                # Kerberos de-multiplexor
kshell          544/tcp         cmd     # and remote shell
eklogin         2105/tcp                # Kerberos encrypted rlogin
krb_prop        754/tcp                 # Kerboros slpave propagation
erlogin         888/tcp                 # Login and environment passing
#       Kerberos sample server
sample        906/tcp                 # Kerberos sample app server
sample        906/udp                #for kerberos simple test
```

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. You may notice that /etc/services has a telnet service defined at port 623; we did not experiment with this and do not know how it relates to the INTCLIEN version of telnet/tn3270.

13.4 Other Setup and Customization

{ZM06} 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)
```

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:

```
//EZAZSSI PROC P=''
//STARTVT EXEC PGM=EZAZSSI,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. 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.

13.5 Advanced TCP/IP Customization

{ZM07} 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.

13.5.1 Dynamic Routing

{ZM08} To use dynamic routing, you *must* disable static routing by removing (or commenting out) *all* the GATEWAY statements in your PROFILE data set. You can then enable dynamic routing. In our examples, this might be:

```
; 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.

Note that, with the OS/390 releases discussed here, *you must also run the OE RouteD server on the same OS/390 system.* The setup for OE RouteD is described in Chapter 24 of the *IP Configuration* document. If you work with OE RouteD, remember to configure VARSUBNETTING, IGNOREREDIRECT, and DATAGRAMFWD in your PROFILE controls.[140]

13.5.2 How to Start Using the Network File System

{ZM09} OS/390 TCP/IP includes both an nfs client and nfs server. The OS/390 AD CD-ROM V2R6 release 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 MVSNFS.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.[141] The list of files and data sets in MVSNFS.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), that are started via VTAMAPPL. You may want to edit or remove these if you are not using nfs. 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

[140] These are discussed on page 238 of the *Volume 1: Configuration and Routing* (SG24-5227) Redbook.

[141] 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.

AIX clients, and had no problems. However, we understand that some nfs clients may not match the nfs protocol version used by the OS/390 nfs server.

You *must* read the nfs manuals listed above if you plan to use either client or server functions. If you do not plan to use these functions, we recommend removing both functions from SYS1.ADCD06.PARMLIB(VTAMAPPL).

13.6 How to Begin Diagnosing TCP/IP Problems

{ZM10} A structured approach to problem solving is important when dealing with TCP/IP problems on these 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 device manager.
- OS/390 TCP/IP is not working.

This set of symptoms means that the server TCP/IP is correctly configured -- at least for the loopback address.[142]

Do you have the most recent P/390 (or R/390) device managers?. A number of fixes and upgrades have appeared over the last few months. 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, AIX), does it work? Can you ping it? (This would be address 9.12.17.150 in our examples.) If not, you have problems at the server level. For a P/390, 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 or AIX TCP/IP definitions. For an R/390, use smit to verify that the adapter is available.

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:[143]

```
        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
        0E20 CTC  O
        0E21 CTC  O
```

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:

142 This statement assumes you are not using CM/2 for your local sessions on a P/390.

143 We will assume the unit addresses are E20 and E21 for the rest of this section.

```
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:

1. 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 Current Configuration window. It should have only IEEE802.2. Note the adapter number before the IEEE802.2 line.
2. Review C:\IBMCOM\LANTRAN.LOG, looking for any obvious error messages associated with this line. There should be messages indicating the adapter started.
3. 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.
4. For an R/390, use smit to verify that AIX TCP/IP is not using the adapter. The TCP/IP adapter status should be *down*.

Start the P/390 Configurator. Be certain the appropriate addresses are defined, as shown in 14.3.5, "Update the P/390 Configuration" on page 266. 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 TCPIP
   (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 your 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* or ping *from* a more remote system, the ping may fail due to routing problems. There must be defined routes from your system (using GATEWAY statements, for example) and to your system (and these must defined in the remote system and any intermediate routers).

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 LAN.*. 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, we suggest starting with 1492, not the default 1500. If nothing works, verify your DIX or IEEE802.3 selection. The steps are documented earlier in this chapter. 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. Your server TCP/IP (OS/2, AIX) 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, in the OS/2 TCP/IP configurator, and in smit.

Useful commands for TCP/IP debugging are:

1. **ping** from a TSO command line
2. **oping** from an OMVS command line
3. **oeping** from an OMVS command line (more options)
4. **tracerte** from a TSO command line
5. **otracert** from an OMVS command line
6. **netstat** from a TSO command line
7. **onetstat** from an OMVS command line
8. **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 otelnet (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.[144] To connect to otelnet, you need to specify a port number when calling 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 9.12.17.151  -p 1023     or
telnet 9.12.17.151  1023        or
telnet 9.12.17.151:1023
```

or something similar.

[144] 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.

13.7 How to Compile TCP/IP Applications

{ZM11} 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=SYS1.SCEERUN
//                DD DISP=SHR,DSN=SYS1.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.

13.8 How to Set FTP Data Set Parameters

{ZM20} 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 R6 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. We understand that some ftp clients are designed for *passive FTP*, in which all connections[145] are initiated from the client. This apparently satisfies SOCKS firewalls.

13.9 How to FTP From a Batch Job

{ZM21} 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 eNetwork Communications Server IP User's Guide* (GC31-8514).

13.10 How to Establish TN3270E Connections

{ZM22} 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
```

145 Multiple TCP/IP port connections are used during the course of an ftp session

```
  Exit
Exit.  The session will restart.
```

The LU name you specify must be defined in the BEGINVTAM parameters of the 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. (The R5 and R6 AD systems do not include second logmode names in their TCP/IP PROFILEs. You will need to modify the PROFILEs if you want to use tn3270E.)

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.

13.11 How to Have Several tn3270 Ports

{ZM23} 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 TELENETPARMS sections to your PROFILE for TCP/IP. For example,

```
...
TELNETPARMS
  PORT 23
  INACTIVE 600
ENDTELNETPARMS
TELNETPARMS
  PORT 2323
  ...
ENDTELNETPARMS
...
```

You can also have several BEGINVTAM ... ENDVTAM blocks. See the IP Configuration manual for V2R6 (SC31-8513-01) for more details.

Multiple telnet ports can be useful for a variety of purposes, especially since you can start and stop individual port functions while TCP/IP is operational.

13.12 How to FTP Partitioned Data Sets

{ZM24} MVS load modules are stored in partitioned data sets, or PDSs.[146] 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 dataset 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

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).

[146] This discussion ignores details that differ with PDSEs.

13.13 Why Some Browsers Do Not Work With FTP

{ZM25} 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. (Other ftp platforms often avoid this problem by using the chroot() function as part of their ftp server startup. OS/390 does not provide the chroot() function.)

13.14 How to Use Nonstandard 3270 Screen Size

{ZM26} 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 with AD R6, 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 Integrated Server. We address two forms of tn3270 connections here:

1. 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.
2. 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 it to a standard size.

This was all that was needed for LAN3274 connections, via OS/2 TCP/IP. It worked because the SYS1.VTAMLST member that defined the terminals contained DLOGMOD=DYNAMICB.[147] 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 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.

[147] For the AD R6 system, this is SYS1.VTAMLST(NSNA90X) and (NSNA70X).

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 manipulates 3270 controls, might not work.

13.15 How to Set Up a Test TCP/IP

{ZM27} *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:

1. 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.
2. The naming conventions used to find the various TCP/IP parameter and configuration files are complex. You should not disturb them without good cause.
3. 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:

1. 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.
2. Edit your new PROFILE2, making the desired changes.
3. 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.
4. 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.
5. Stop the active TCPIP. The MVS command is P TCPIP.
6. 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.

7. 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) ENTRYPOINT(EZBPFINI)
SUBFILESYSTYPE NAME(TCPIP) TYPE(INET) ENTRYPOINT(EZBPFINI)
SUBFILESYSTYPE NAME(TCPIP2) TYPE(INET) ENTRYPOINT(EZBPFINI)
SUBFILESYSTYPE NAME(TCPIP3) TYPE(INET) ENTRYPOINT(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. For the AD system it is BPXPRM00 in SYS1.ADCD06.PARMLIB.)

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.

Chapter 14. P/390, R/390, IBM Integrated Server

{ZX00} This chapter contains information unique to P/390-based systems. These articles do not apply to other, larger S/390 machines.

14.1 How to Verify P/390 Disk Integrity

{ZX05} The P/390 and Integrated Server 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:\MVS\TSOPAK.130.
- Using an OS/2 window, change to your P390 directory.
- Using this OS/2 window, issue the line command:

```
CKDCHECK D:\MVS\TSOPAK.130 /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 (3380) 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 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=3380
//SYSIN    DD  *
   LISTVTOC VOL=3380=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.

14.2 Hardware and Server Setup for TCP/IP

{ZX99} TCP/IP setup for a P/390, R/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 used by the MVS master console and can be used for local TSO, CICS, and so forth. The TCP/IP used by OS/2 (P/390 and Integrated Server) or AIX (R/390) 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 or AIX) **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 or AIX) and one for OS/390.[148] *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.[149] This is sufficient for supporting an MVS master console and several TSO windows on the attached server display.

148 The Application StarterPak is a native S/390 system and does not have an underlying server. However, the OSA adapters it uses for LAN connections can be controlled from the OS/2-based Support Element.

149 It might also use the serial ports for SLIP or PPP connections, but this is not addressed here.

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, R/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 Mpbs 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 and R/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.

14.3 How to Prepare a P/390 for TCP/IP

{ZX01} Current 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.[150]

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:

1. Installing the adapters
2. Configuring the adapters with MPTS
3. Configuring one or more adapters for OS/2 TCP/IP
4. A minor edit for CONFIG.SYS

[150] The Server 500 did not have an integrated Ethernet adapter. You need 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 support by OS/2 for a Server 500 can be used for a Server 500-based P/390 system.

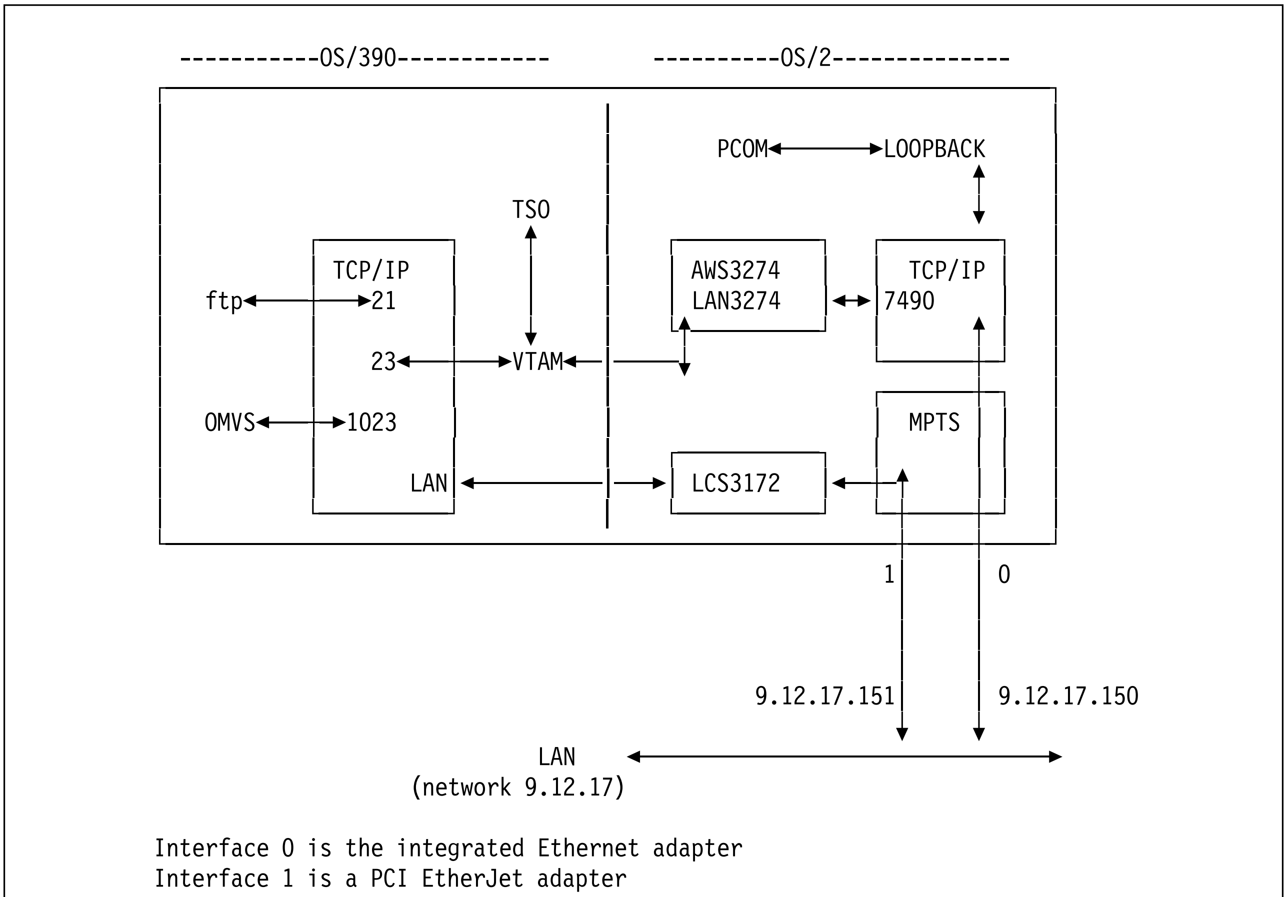


Figure 38. Goal for Basic P/390. This P/390 system has two LAN adapters. Adapter 0 is connected to OS2 TCP/IP and the LAN3274 device manager. tn3270 sessions to port 7490 at 9.12.17.150 will appear to be local 3270 displays to OS/390. tn3270 sessions to port 23 at 9.12.17.151 will connect to OS/390 TCP/IP 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. The PCOM sessions (on the OS/2 side) connect to OS/2 TCP/IP via the loopback address.

5. Update the P/390 configuration (DEVMAP)
6. Rebooting OS/2 and verifying the adapters started correctly
7. Configure PCOM (or another 3270 emulator)

Figure 38 provides a graphic view of our goal for setting up TCP/IP for a P/390 system. It involves both OS/2 TCP/IP and OS/390 TCP/IP.

14.3.1 Installing the Adapters

{ZX01B} 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.

14.3.2 Configuring Adapters With MPTS

{ZX01C} 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 39 on page 260. Note that there are three windows in this panel: Network Adapter, Protocols, and Current Configuration. The initial Current Configuration window will be empty. (Note that all three windows can scroll up and down, and may contain many more lines of data than shown here.)

The general operation of this panel is:

1. Select a Network Adapter that matches one of your adapters and ADD it to your Current Configuration.
2. Then select the protocol(s) that should be associated with this adapter and ADD each to the Current Configuration.
3. 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 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.)[151] 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.)

[151] We assume your diskette will have a later version of the driver than the base OS/2 system is likely to have.

```
LAPS Configuration

Network Adapter                            Protocols

3COM 3C503 EtherLink II Adapter      ↑     Netware NetBIOS Emulat ↑
3COM EtherLink III Family OS/2             IBM IEEE 802.2
Artisoft NodeRunner & AE-X Ether.          IBM OS/2 NETBIOS
IBM PCI Ethernet Adapter Driver            IBM TCP/IP
etc                                  ↓     ......                 ↓

 ADD   CHANGE   OTHER ADAPTER               ADD   OTHER PROTOCO

Current Configuration

AMD PCNet Family Ethernet Adapter                ↑      OK
   0 - IBM IEEE 802.2
   0 - IBM TCP/IP
IBM 100/10 EtherJet PCI Adapter                         CANCEL
   1 - IBM IEEE 802.2

                                                 ↓      HELP

 EDIT   REMOVE   CHANGE NUMBER
```

Figure 39. Primary MPTS Working Panel

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.

Now, select the first adapter you wish to configure (which should be the Integrated AMD Ethernet adapter if you have a Server 330); a single mouse click on the adapter name will select it. Then select the ADD button. This should result in a new line in the Current Configuration window, similar to the first line shown in the figure.

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.[152]

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**

[152] IEEE 802.2 is also selected for adapters intended for OS/390 SNA or OS/2 SNA.

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*, 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 39 on page 260 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.[153]

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               (sometimes 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.**

To change your MPTS 802.2 ethernet protocol to DIX:

1. In the Current Configuration window, double click (with the mouse) on 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
```

2. This should produce a large display (scrollable) titled *Parameters for IEEE 802.2*. This display contains lines such as:

[153] As an aside, note that either (or both) adapters can be used for SNA at the same time they are being used for TCP/IP. SNA connections can share adapters between OS/390 and OS/2 and can use adapters that are running TCP/IP at the same time.

```
* ..... (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**.

3. Do not change any other parameters in this panel unless you have a specific requirement and understand what you are changing.
4. Click OK to exit.
5. 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.) 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.[154]

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:

1. In the Current Configuration window, double click (with the mouse) on the title line (not a protocol line) for the token ring adapter.
2. 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).
3. If you wish, find the line named *Enable full-duplex* and change it from YES to NO.
4. 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

[154] 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.

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 for this question.

14.3.3 Configuring Selected Adapter(s) for OS/2 TCP/IP

{ZX01D} 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 40 on page 264.

Assuming you will use the TCP/IP interface for AWS3274,[155] you should first enable the loopback interface:

1. Select (single click) the LOOPBACK INTERFACE line on the Configure Network Interface display.
2. Select the ENABLE INTERFACE option.
3. 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 40) to the MPTS adapter numbers (as illustrated in Figure 39). 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
```

In this example, four LAN adapters are defined. The first two are Ethernet and the last two are token ring --- although the LAN types are irrelevant to the numbering scheme. The adapter numbers are 0 - 3.[156] The OS/2 TCP/IP interface numbers are assigned, starting with interface 0, only to adapters having the (OS/2) TCP/IP protocol associated with them.

[155] That is, assuming you are no longer using CM/2 for 3270 emulation on the P/390 display. Earlier releases of the P/390 support programs (the AWS3274 device manager, in particular) used CM/2 to provide several 3270 sessions for the MVS master console, and several TSO sessions.

[156] These numbers are used later by OS/390 TCP/IP and VTAM.

```
@---------------------------------------------------------
|     Configure Network Interface                         |
|                                                         | Network
|   LAN interface 0      [X]  ENABLE INTERFACE            | Routing
|   LAN interface 1                                       |
|   LAN interface 2                                       | Hostname
|   LAN interface 3      [ ]  AUTOMATIC (DHCP)            |
|   LAN interface 4                                       | General
|   LAN interface 5                                       |
|   LAN interface 6      [X]  MANUAL, USING:              | Security
|   LAN interface 7                                       |
|   loopback                                              | Servers
|                   IP   [ 9.12.17.150   ]                |
|                                                         | Printing
|                                                         |
|   [ADVANCED OPTIONS]  MASK [ 255.255.255.0 ]            | Mail
|                                                         |
|                                                         | etc
|_________________________________________________________|
```

Figure 40. Primary Configuration Display for OS/2 TCP/IP

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:

1. Select (single click) the LAN INTERFACE 0 line on the Configure Network Interface display.
2. Select the ENABLE INTERFACE option.
3. 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.
4. 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.[157] 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).

[157] There has been considerable discussion about the best Maximum Transmission Unit size for Ethernet on a P/390, resulting in a range of opinions and recommendations. The author has had no problems with the 1492 size.

The Configure Network Interface display has a number of “tabs” on the right-hand side. 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.

1. Select the Autostart tab. This will display the Configure Automatic Starting of Services screen.
2. 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.
3. Select **telnetd** in the left-hand window and select **Autostart service** and **inetd super server daemon** on the right-hand side.
4. Select **portmap** in the left-hand window and select only **Autostart service** on the right-hand side.
5. You should not need **ftp** or **tftp** unless you require these for your server (OS/2 or AIX) operation. (We used tftp to boot IBM Network Stations, but this was for our specific setup.)

Select and set the other tabs, as follows:

1. Routing. We normally make no entry here. However, you may want to configure IP Forwarding active on this screen.
2. Hostname. We normally make no entries here.
3. General. We normally make no entries here.
4. Security. You should enter a telnet password on this screen. We strongly recommend this.[158]
5. Servers. We normally make no entries here.
6. Printing. We normally make no entries here.
7. Mail. We normally make no entries here.
8. Sendmail. We normally make no entries here.
9. 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.

14.3.4 Correcting CONFIG.SYS

{ZX01E} 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. 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.*

---

158 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. Never versions of OS/2 TCP/IP can have multiple passwords and they are stored in an encrypted format. However, most P/390 owner use the older OS/2 TCP/IP that was preloaded or installed from a basic OS/2 CD-ROM.

If you 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[159] and you might want to place it last last in these statements.

14.3.5 Update the P/390 Configuration

{ZX01F} 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.[160] S/390 addresses associated with these device managers appear (to OS/390) to be locally-attached non-SNA 3270 terminals. If you are not using CM/2, these two device managers perform the same function. You will use AWS3274 to define 3270 sessions on the display that is directly attached to your P/390 system, and LAN3274 to define 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.

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 41 on page 267. 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 letter if your display; use the letters in your display. 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.)

---

[159] The author has never bothered to alter this placement, but others do.

[160] These device managers can use CM/2 instead. For this discussion, we assume you are not using CM/2.

```
       Active Devices: nn   Available disk space in Mybtes nnn.nnn
   ┌──────────────────────────────────────────────────────────────┐
   │ Addr Device  Label    Atype  Size     Mgr  FN/P              │
   │ >    >       >        >      >        >  >                   │
   │  00C  2540                             7   D:\RDR\*.JOB      │
   │  120  3380    HFSVOL   CKD    1770C    2   G:\OS390\HFSVOL.120│
   │  122  3380    SCPMV5   CKD    1770C    2   E:\OS390\SCPMV5.122│
   │  124  3380    WORK02   CKD    300C     2   E:\OS390\WORK02.124│
   │  700  3278             DSPY            3   Master Console    │
   │  701  3278             DSPY            3   Master Console    │
   │  900  3278             DSPY            4                     │
   │  901  3278             DSPY            4                     │
   │  902  3278             DSPY            4                     │
   │  A20  3390    OS39R6   CKD    3339C    2   F:\OS390\OS39R6.A20│
   │  A22  3390    OS3R6A   CKD    2226C    2   F:\OS390\OS3R6A.A22│
   │  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=AWSPCSRV O=CHAN370  P=AWSICE
```

Figure 41. P/390 Device Configuration Display

You should define several LAN3274 addresses starting at 900, 901, 902, and so forth. Examples of these are shown in Figure 41. 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.[161] 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-E2F and E40-E47 generated as CTCs, and you can use any even/odd pair in this range. For our examples we will use

[161] 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.

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 41.

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.[162] 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.

14.3.6 Rebooting OS/2 and Checking Results

{ZX01G} At this point, we suggest you shutdown 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 Mpbs, 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 example \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 network. 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

162 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.

configurator (newer releases)). The OS/2 telnet function can be used to remotely IPL OS/390, among other useful tasks.

14.4 Configuring PCOM

{ZX02} You need an 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.[163] There are many 3270 emulator products available from many vendors. One is IBM's *Personal Communications AS/400 and 3270*, 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 will 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.[164] 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:

1. Screen size 32x80 (this is optional, but very common)
2. Link Parameter - select CONFIGURE LINK
   a. Host name - 127.0.0.1 (or any other appropriate address)
   b. Select ADVANCED SETUP
      1) Change the port number to 7490 (or port 23 for direct OS/390 TCP/IP connections
      2) 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

163 Earlier P/390 systems used CM/2 for these functions.

164 Setup for this is described later.

sessions. We usually create three sessions for PCOM installed on the P/390 itself, and two sessions for PCOM installed on other personal computers.

Start one of the sessions. There is a pulldown 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.[165]

After starting one of the PCOM sessions, you can use the COMMUNICATIONS pulldown 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.

14.4.1 Summary and Next Step

{ZX01I} If you have followed the steps in this section, your P/390 should:

1. Have LAN adapter cards installed
2. Have MPTS configured to cover both OS/2 TCP/IP and OS/390 TCP/IP (using separate adapter cards, of course)
3. Have OS/2 TCP/IP configured and operational
4. 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.
5. 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 Chapter 13, “TCP/IP” on page 223.

14.5 IBM Integrated Server Setup

{ZX10} 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.

[165] Defining the square brackets as “apl ad” and “apl bd” (which results in the hex values ad and bd for the characters) worked well for us.

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.

14.6 R/390 Setup

{ZX20} There have been several base RS/6000 machines used for R/390 systems. The current base is the RS/6000 model F50, and we will assume this base for the following discussion. The setup goal is the same as for the P/390 system illustrated in Figure 38 on page 258, except that AIX is used instead of OS/2. Just as for a P/390, you will probably require two LAN adapters -- one for AIX TCP/IP and one for OS/390TCP/IP. Just as for a P/390, you could configure your server (AIX) TCP/IP to work without a LAN adapter (using only the loopback address). And, just as for a P/390, we strongly suggest you use two LAN adapters because it provides more flexibility in using your system.[166]

R/390 owners are often experienced UNIX users. If this is the case, we will assume that you can install and customize your LAN adapters for use under AIX. We suggest you activate all your LAN adapters under AIX TCP/IP and verify that they are working. This can be done with the *smit* menus.

Basic setup considerations include:

1. Do you have the AIX device drivers installed for your LAN adapters? The normal AIX installation process installs only the device drivers for the devices actually present. If you add another type of adapter, you will need to install the device drivers for it. (Adding another instance of an existing adapter type does not require additional drivers.) smit menus exist to assist you in installing additional drivers from the AIX distribution CD.
2. Configure and test your LAN adapters with AIX TCP/IP.
3. Use the current version of the R/390 support programs. At the time this was written, this was version 4.2.1.5. There have been substantial changes in this release. For example, it is no longer necessary to edit the IPL script to enable or disable various device managers or to detach LAN interfaces from AIX TCP/IP.
4. Use the most current fixes for LCS3172. These are available from the ftp site *p390.ibm.com*. There have been substantial improvements in performance in the latest versions.
5. Remember to make the DLPI modifications that are described in the LCS3172 DOC file. OS/390 TCP/IP will not work correctly without this step.
6. Ethernet connections default to the DIX format. See the device map sample below for an indication of how to change to IEEE802.3 format.
7. When using the R/390 configurator panel, press F12 to view the device manager parameters. You must set the parameter for LCS3172 to NOPARMS.

166 For example, if your OS/390 TCP/IP configuration is not correct, you will need a local 3270 connection (provided through AIX TCP/IP) in order to log onto TSO and correct the problem. (The loopback address can be used for this, and does not require an adapter.)

```
    Active Devices: nn   Available disk space in Mybtes nnn.nnn

Addr Device  Label    Atype  Size     Mgr  FN/P
>    >       >        >      >        >  >
 00C  2540                             7   /os390/*.job
 120  3380    HFSVOL   CKD    1770C    2   /os390/HFSVOL.120
 122  3380    SCPMV5   CKD    1770C    2   /os390/SCPMV5.122
 124  3380    WORK02   CKD    300C     2   /os390/WORK02.124
 700  3278             DSPY            3   Master Console
 701  3278             DSPY            3   Master Console
 900  3278             DSPY            4
 901  3278             DSPY            4
 902  3278             DSPY            4
 A20  3390    OS39R6   CKD    3339C    2   /os390/OS39R6.A20
 A22  3390    OS3R6A   CKD    2226C    2   /os390/OS3R6A.A22
 E20  3088                             A   /dev/ent1
 E21  3088                             A   /dev/ent1


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=AWSPCSRV O=CHAN370  P=AWSICE
```

Figure 42. R/390 Configurator. The contents are very similar to that of a P/390. Note the extra operand in the line for address E20 and for address E21.

8. We assume you will normally use two LAN adapters: one for AIX and one for OS/390. However, it is possible to have a viable system with only one adapter that is assigned to OS/390. You would use the loopback function (under AIX) to connect 3270 emulator sessions (on the RS/6000 display) that provide the MVS master console and at least one TSO session.

9. There is no AIX equivalent for the MPTS and CONFIG.SYS functions of OS/2. AIX TCP/IP setup is through smit and is a completely different process than OS/2 TCP/IP setup. Otherwise, some of the secondary comments under 14.3, "How to Prepare a P/390 for TCP/IP" on page 257 may be useful.

Figure 42 contains an example of a device map for an R/390 system. Note the lines for addresses E20 and E21. The FN/P field must contain the AIX name of the LAN adapter to be used for this connection. *Starting with R/390 release 4.2.1.5* the S/390 startup process (when you click IPL) will automatically detach this LAN adapter from AIX TCP/IP, making it available for OS/390 TCP/IP.

As shown, the LAN connection (assuming it is Ethernet) will be in DIX format. If you modify the FN/P operand to be "/dev/ent1,802.3," it would be used in IEEE802.3 format.

Use the LAN number (ent0, ent1, ent2, tok0, tok1, and so forth) as the third operand (a single digit) in the LINK statement in the OS/390 TCP/IP PROFILE data set. This is described in 13.2.5, "Basic Adapter Customization" on page 229.

14.7 Application StarterPak Setup

{ZX30} 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 Mpbs) 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.

14.7.1.1 Connections To the ENTR OSA-2 Adapter

Figure 43 on page 274 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.

14.7.2 Resetting the Ethernet vs Token Ring Selection

{ZX30C} 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:

1. Remove any connections from each of the six connectors on the ENTR adapter.
2. 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.
3. Use the SE tools to enable a reset function for the OSA adapter. This is described below.
4. 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):

1. A2 - Not Operational. This will be on when the card is first reset; it should not be on during normal operation.

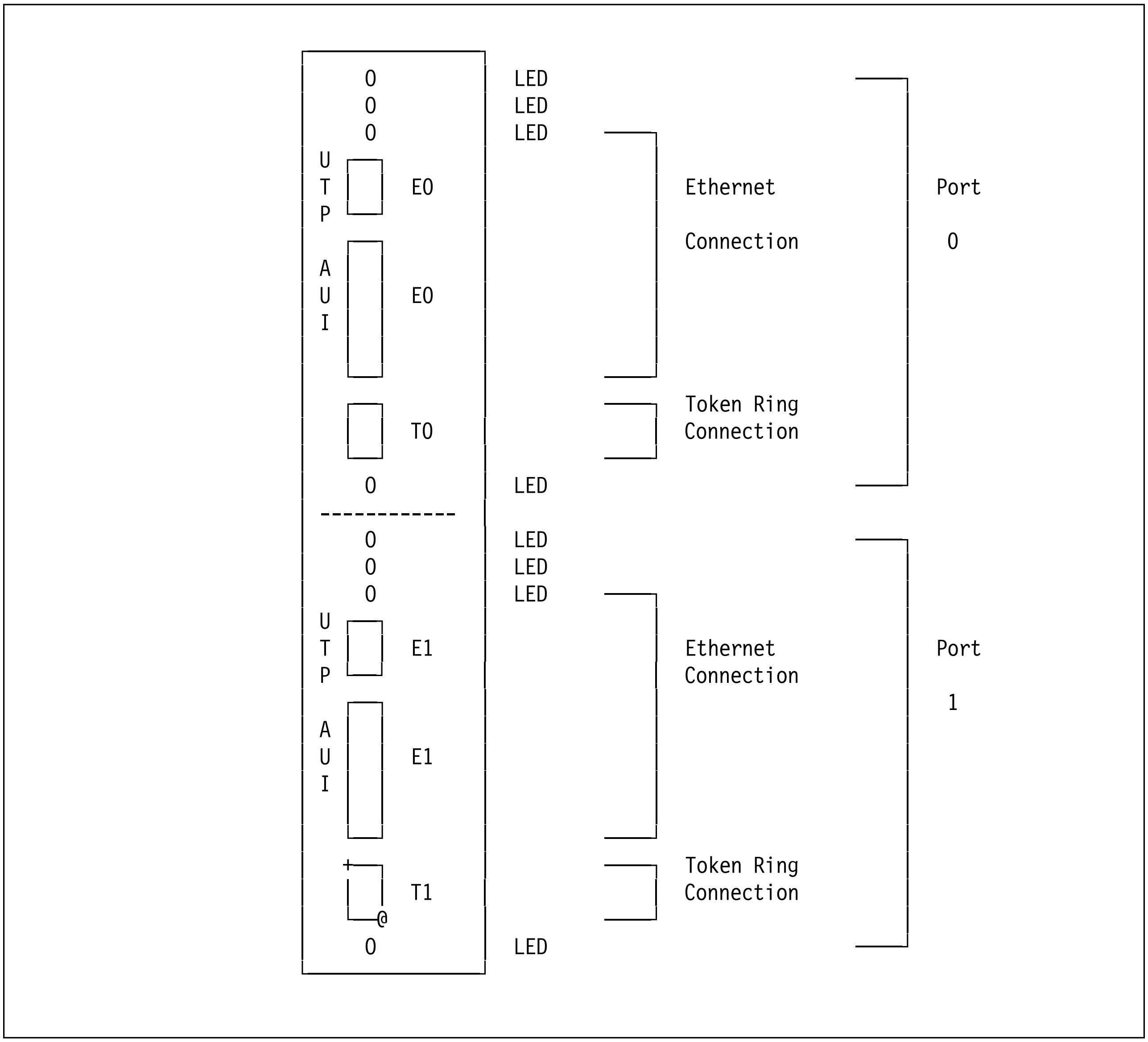


Figure 43. OSA-2 Ethernet / Token Ring Adapter Connections. Each ENTR adapter, as seen from one end of an Application StarterPak (with the end cover open) has six connectors and a number of LEDs.

2. B2 - Operational. This light should flash during normal operation of the adapter.
3. C2 - Ethernet port not operational on LAN.
4. D2 - Token ring port not operational on LAN.

There are a great many 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:

1. Select the **Groups** option on the left-hand side of the display.
2. Double click on the **CPC** icon.

3. 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.
4. This should produce a small window offering a choice of CPs or CHPIDs. Click on the **CHPIDs** option.
5. 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.
6. On the right-hand size of the SE display, find the **CHPID Operations** display.
7. 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.
8. You should receive a window with a single choice: **OSA Specific Advanced Facilities**. Select this option and click OK.
9. 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
```

10. You can try the various options to learn more about your OSA port.
11. Before exiting, select **Enable autosense on next reset**, and click **apply**.
12. Exit from the OSA and CHPID panels on the left-hand side of the screen.
13. Be certain you have your desired LAN connections for the ENTR ports.
14. Perform a POR, using this option in one of the right-hand displays on the SE console.

14.7.3 Finding Unit Addresses of OSA Interfaces

{ZX30D} The Application StarterPak uses IOCDS files to associate channels, control units, and devices with the unit addresses 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 unit 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 has OS/390 working (implying that a valid IOCDS is being used), you can use the SE console as follows:

1. On the right-hand side select the **CPC Configuration** display. On the left-hand side, select **Groups**.
2. 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.
3. 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.
4. Select the **View** option from the action line of the I/O Configuration window; then select **Channel Path Configuration** from the drop-down menu.
5. 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.[167] Use the mouse to select the line corresponding to your OSA chpid. Then select **View** in the action line and select **Device Information** from the drop-down menu.

6. 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.

7. Exit from the SE display screens.

If you have access to the IOCDS used for your system[168] 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
```

[167] This implies it is located in the right-most slot of the available slots for OSA adapters.

[168] Access is typically from the HCD utility, under TSO.

```
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,   *
      UNIT=3270
```

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 the redbook *Open Systems Adapter 2 Implementation Guide*, IBM order number 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.)

14.8 How to Display OS/2 Cache Statistics

{ZX40} If you are using Warp Server Advanced with your P/390, you can display cache statistics, with a dynamic update. In an OS/2 window, issue this command:

```
CACHE386 /S:D
```

There should be no spaces in the "/S:D" string. This will display statistics until you enter Ctrl-C to stop it.

14.9 How to Use the CPU ID of a P/390

{ZX42} 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):

```
0        8                          32             48          63
-----------------------------------------------------------------
| vers   | serial                   | model          |  zeros    |
-----------------------------------------------------------------
```

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 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 (usually virtual memory) characteristics.

The version numbers are not intended as a stable programming interface. They may change on future cards, without notice. Also, note that some of the fields appear to change when running under VM.

14.10 How to Use Two SCSI3480 Drives

{ZX44} It is possible to use two SCSI tape drives as 3480 (or as two 3420s, for that matter) drives. In essence, the process uses two copies of the SCSI3480 program under different names.

The following steps are involved:

1. Add the following line to CONFIG.SYS:

   ```
   DEVICE=D:\P390\AWS3480.SYS xx,yy
   ```

   where xx is the SCSI address of the tape drive. This statement can be added after a similar statement for the SCSI3480.SYS module. Be certain your two SCSI tape drives have different SCSI addresses. The yy parameter is "00" if you have only one SCSI adapter; otherwise it is the number of the adapter.
2. Use the P/390 Configurator to define one tape drive corresponding to SCSI3480 and one drive corresponding to AWS3480.
3. If your configurator display does not include AWS3480 in the list of device managers at the bottom of the screen, then

   a. Edit D:\P390\AWSCTYPE.MGR. Add this line at the end of the file:

      ```
      AWS3480 3480
      ```

      or use AWS3420 3420 if you require a second SCSI 3420 drive.

   b. Save and restart the P/390 configurator. The new device manager, AWS3480, should be listed at the bottom of the screen, with an associated code. For example, manager "P" might correspond to AWS3480. Define a device using this code:

      ```
      >561  >3480  >        >          >      > P >
      ```

      You can use any 390 address for the drive, but MVS should have an appropriate 3480 defined at that address.

   c. You must reboot to use the new CONFIG.SYS information. The second SCSI3480 drive should be available for use.

If you need more than two SCSI3480 devices, you must obtain the ISITAPE device manager, which is available from an independent vendor. It contains installation instructions and will emulate multiple 3420, 3480, 3490, and 3490E drives, and is used in place of SCSI3480 and AWS3480.

14.11 How to Perform AWSTAPE Mounts

{ZX46} Tapes emulated with the AWSTAPE device manager can be used as labelled[169] 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 "remove" 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).

[169] There was a known problem in earlier releases of AWSTAPE, involving multi-file standard-label tapes. Label processing used a read backwards command in this case, and the P/390 did not emulate this correctly. Label handling in current releases of AWSTAPE should be correct.

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.[170] 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.[171] 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.

170 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.

171 If you rework your configuration, we suggest that you might want to specify OFFLINE=YES for your tape drives.

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).

14.12 How to Install an R/390

{ZX48} This chapter is not intended as an introduction to R/390 systems. You can use the redbook *P/390 and R/390 with OS/390: An Introduction* (IBM order number SG24-4847) for a general introduction to R/390 systems. The following text briefly describes the general steps needed to install the software (AIX, R/390 programs, and OS/390) for an R/390 system.

R/390 systems are configured and provided by several vendors, and the specific configurations vary from case to case. In general, there is a wider range of "typical" R/390 configurations than there is for P/390 systems. For this reason, we cannot provide a finely detailed set of installation instructions. R/390 variations include the use of SSA disks or SCSI disks, with either RAID or non-RAID adapters.

The disk configuration is significant. The *standard* installation process for the the OS/390 AD CD-ROM[172] assumes that a sufficient number of free disk drives are available. In this case, "free" means the disks are not part of an existing volume group and "sufficient" means that about 12 GB of disk space is available on the free drives.

If your disk arrangement does not match that assumed by the standard installation script, you can manually UNZIP the OS/390 volumes on the CD-ROMs and control the placement of the resulting files. This manual installation process is described later.

14.12.1 AIX Installation

{ZX48B} At the time we did this work, AIX 4.3.2 was the current AIX level. We installed AIX 4.3.2 from CD-ROM, using very basic installation options. This was done by loading the AIX CD-ROM, turning the system key to Service (if your system has a key), applying power to the system, and following the instructions on the screen. The goal is to install a standard AIX system; no unusual requirements exist for the R/390.

After the AIX CD-ROM installation process starts, you are given several choices by selecting the Change/Show Installation Settings and Install option. In this panel, we elected to do a *New and Complete Overwrite* installation (as opposed to a Migration installation). In this same panel, we selected the disk we wanted to use for AIX (hdisk0) and elected to install the tcb (Trusted Computing Base). R/390 operation does not depend on any of these choices, and you may do something different. Many users perform a Migration installation (sometimes known as a Preservation installation) instead of the New and Complete Overwrite; we know of no problems in this area.

The installation process takes about 30 minutes once it started.

[172] This is a set of CD-ROMs available to IBM's S/390 Partners-in-Development organization. In addition to OS/390, these CDs contain an extensive collection of other program products useful to developers. The "AD" qualifier means "Application Development" and is often used as a shorthand notation for this particular CD-ROM set.

The R/390 needs a number of additional programs from the AIX CD-ROM. There are:

1. The bos.dosutil bundle of programs.
2. The bos.dlc bundle of programs
3. Any additional drivers for your specific printers
4. Any additional drivers for ISA adapters you might have

We used smit to install these:

```
smit
  Software Installation and Maintenance
    Install and Update Software
      Install/Update Selectable Software (Custom Install)
        Install Software Products at Latest Level
          Install New Software Products at Latest Level
            Input device  - /dev/cd0   (or press F4)
              SOFTWARE to install  (press F4 for list)
              (It takes some time to produce the list)

Scroll through the list, and use F7 to mark the components you
want to install.

              ENTER to begin installation
              SCROLL to check the SUCCESS messages
```

You must reboot AIX before this LAN software is used.

Using the **df** command, We observed that our /usr file system was almost full, and knew that the P/390 support programs (and other products we might want to install) would go into this file system. An AIX file system is in a logical volume. If you expand the file system, the logical volume is automatically expanded (provided there is room in the volume group containing the logical volume). To expand a file system:

```
smit
  System Storage Management
    File System
      Add / Change / Show / Delete File System
        Journaled File System
          Change / Show Characteristics of a Journaled File System
            (Select /usr from list shown)
              SIZE (in 512-byte blocks) = nnnnnnn
                (overtype the size field with a larger number)
                  ENTER
```

For example, the SIZE shown for /usr was 434176 512-byte blocks. We changed this to 510000 blocks, which added about 40 MB to /usr. The system automatically added a number of LPs (logical partitions) to /usr to match the requested size. We also added 32000 blocks (16 MB) to the root file system, and the same amount to the /home file system.

You should configure a minimal TCP/IP environment, defining a host name and a LAN TCP/IP address. The default R/390 IPL script will use these in order to start local tn3270 sessions. This script uses the *hostname* command to obtain whatever name you select for your system name. (It is possible to use the R/390 without a LAN adapter associated with AIX TCP/IP, but this involves modification of the IPL script to use the loopback address.) *After you install the R/390* *programs, read the LCS3172.DOC file that will be in /usr/bin/r390.* It describes changes you may need to make to your AIX TCP/IP environment.

We added another user ID at this point. This is not required, since all the P/390 support functions and operation are under the *root* user ID.

```
smit
  Security and Users
    Users
      Add a User
        User NAME = OGDEN
        (no other parameters needed.  Press ENTER)
      Change a User's Password
        User NAME = OGDEN
          (enter password)
```

If you intend to use *hcon* for 3270 emulation, you will need to install, configure, and add users for it. In general, R/390 users are *assumed* to be using hcon. It is built into the IPL script that is used to start an OS/390 operating system on the R/390. Moreover, this default IBM-provided IPL script assumes that you have two hcon sessions, named “a” and “b,” defined for AIX userid *root*. If your hcon installation does not provide exactly this set of user and session definitions, you need to change hcon or the ipl390 shell script (after you install it from the R/390 diskettes).

At this point we shut down AIX, turned off the power, and verified that the P/390 adapter was installed. When AIX was restarted, the CDE (full-screen X Windows) logon screen appeared. There is an option to use a standard command-line logon instead, and we used this option to complete the installation process. (You can install and operate the R/390 under CDE if you like; we are not aware of any restrictions in this area, but we prefer to use a single command-line login and then start xinit manually.

14.12.2 R/390 Support Programs Installation

{ZX48C} If you have an older version of the R/390 support programs installed, you must remove these before installing a new version. The path for removing an older version is:

```
smit
  Software Installation and Maintenance
    Maintain Installed Software
      Remove Software Products
        SOFTWARE name                      (r390.obj)
        PREVIEW only?                      (NO)
```

(Later versions of smit provide an *install-remove* option that can combine these two steps.)

The R/390 support programs, on multiple diskettes, are installed with smit. You should log in as root and do the following:

```
smit
  Software Installation and Maintenance
    Install/Update Selective Software (Custom Install)
      Install Software Products at Latest Level
        Install New Software Products at Latest Level
          INPUT device/directory for software        (/dev/fd0)
          SOFTWARE to Install                        (all_licensed)
          PREVIEW only?                              (NO)
```

```
Use all the default values shown
```

The installation process should take about five minutes. Check the messages after it finishes. There should be *successful* messages for both the root and user portions.

The P/390 support programs are installed in /usr/lpp/r390/bin, which is linked to /usr/bin/r390. That is, the files are actually installed in directory /usr/lpp/r390/bin, but can also be accessed in directory /usr/bin/r390. This second directory path name is shorter, and is normally used for references.

You need to add this directory to your PATH. Edit /etc/environment and add /usr/bin/r390 to the end of the PATH definition:

```
PATH=/usr/bin:/etc:/usr/sbin:.../usr/bin/X11:/sbin:/usr/bin/r390
```

There is one global environmental variable that must be defined. If necessary, first change the permissions for /etc/profile so you can edit it:

```
chmod 755 /etc/profile
```

and then edit /etc/profile and add:

```
CONFIG_FILE=/mvs/os390/devmap.mvs
PS1='$PWD # '
export CONFIG_FILE PS1
```

at some appropriate place, using the directory and file name you will use for your system. (Our use of these names is described in the next section.) You are not required to change the PS1 prompt, as shown here; however we find usage easier if the AIX prompt line displays the current directory. (If you have a more sophisticated prompt command, this is the time to install it.)

The P/390 licensed code and diagnostics should be installed next. This is the same diskette used for P/390 systems, and the label may indicate P/390 instead of R/390. It is the *IBM PC Server S/390 Advanced Diagnostics and Option Diskette*, which is separate from your other R/390 diskettes. To install, insert the diskette in the reader and use the smitty fastpath `smitty p390`.

This uses a series of smit panels and options that are unique to the R/390 (and also relatively new; early R/390 users may not be aware of these smit panels). There are options to install the licensed code, to install ARTIC modules (for the PCI version of the R/390 S/390 channel adapters), to run R/390 diagnostics, and so forth.

If you plan to use a SCSI tape drive to emulate a 3420/3480 drive for MVS, you should be certain the drive is *available* in AIX. Note that SCSI tape drives are usually narrow (8-bit devices) and experience has shown that these are best attached through a narrow SCSI adapter. That is, even though cables are available to connect a narrow device (a tape drive, in this case) to a *wide*SCSI adapter, we recommend not using this method of attachment.

You should also set the defined blocksize of the drive to zero, which means that variable- length records may be used. This may be done by:

```
smit
  Devices
    Tape Drives
      Change/Show Characteristics of a Tape Drive
        (use F4 to select your drive; e.g., rmt0)
        BLOCK size (0=variable length)              (0)
        BLOCK SIZE for variable length input        (0)
        NO EXTERNAL TAPE LABELS
```

Read /usr/bin/r390/LCS3172.DOC. If necessary, make the changes it describes for your LAN drivers used with TCP/IP.

14.12.3 OS/390 AD CD-ROM Installation

{ZX48D} The following description applies generally to S/390 Application Development CD-ROM systems. See Chapter 2, "OS/390 AD CD-ROM Status" on page 3 for information about the current AD release.

A shell script named **insos390** is provided on the first CD to install the CD-ROMs. This shell script assumes you have sufficient free disks (not in any volume group), and it installs all the distributed files on the CD-ROMs, including a number of emulated tape files containing migration jobs and a large amount of (potential) PTF service. For the initial installation of a CD-ROM system, probably done before your R/390 system was delivered to you, this installation shell script is completely appropriate.

The installation script provided with the AD CD-ROMs performs all the following steps for you. The steps are documented here in case you need to intervene, restart, or modify the process. It is unlikely you would want to exactly duplicate these steps, but understanding the general process may be helpful. (The following material was written several releases ago, and the specific steps may vary somewhat between releases.)

These instructions create a separate volume group and file system to hold the various emulated 3380/3390 volumes used by OS/390, plus a few miscellaneous files. This is not required. You can create the files containing the emulated volumes in any JFS file system and volume group. We made a separate volume group and file system for ease of management.

1. Log into AIX as *root*. All of the steps assume you are working as root.
2. You will need to write large files, and you must change the *ulimit* parameter to permit this. Edit /etc/security/limits and add the line (if you have not already done this):

   ```
   root:
       fsize=4194303
   ```

   to the stanza for root. This permits you to write files up to 2GB. (The number is the maximum number of 512-byte blocks.) We found we needed to log out and log in again to have this new size take effect. Issue the *ulimit* command to verify that your maximum file size is at least 4000000; if it is not, the following installation steps will fail.
3. Our system had five disks that were unused so far. We will use these to make a large volume group to hold the emulated S/390 volumes. We decided to use the name *mvs* as the volume group name. The AIX command:

```
mkvg -d 10 -y mvs hdisk1 hdisk2 hdisk3 hdisk4 hdisk5
```

will create the volume group. If your individual disk drives are larger than about 4.0 GB, or if you have a RAID array that makes a single large logical drive, you will need another operand to force the physical partition (PP) size to larger than the default value of 4 MB. A sample command would be mkvg -d 10 -y mvs -s 8 hdisk1 hdisk2 etc. This would cause all physical partitions to be 8 MB, and is suitable for a drive up to 8 GB. The operand -s 64 would be used with a RAID array up to 64 GB. Most documentation assumes that the PP size (and the logical partition (LP) size) is 4 MB. If you must use a larger size than 4 MB, take care reading instructions and examples that assume a 4 MB PP/LP size.[173]

4. Use the command

   ```
   varyonvg mvs
   ```

   to make the volume group accessible.

5. Make a mount point for a new file system in the root directory:

   ```
   cd /
   mkdir mvs
   ```

   creates a directory named mvs, and we will mount our new file system over this directory. (There is nothing special about the name mvs; we wanted something that would clearly indicate R/390 files and selected this name. At this point we are using the name *mvs* for a volume group, a directory, and (in the next step) for a file system; these multiple usages do not conflict.

6. We now want to make a file system that uses all (or almost all) the space in the volume group. We used this command:

   ```
   crfs -v jfs -m /mvs -g mvs -A yes -p rw -a size=21900000
   ```

   This will take several minutes to create a file system of size 21900000 * 512 or 11.21 GB. (This leaves a small amount of free space in our volume group. We were uncertain whether this would be needed for the jfs log file.) You should adjust the size parameter to whatever fits in your mvs volume group, of course. The file system will be mounted automatically when AIX is started.

   You can determine exactly how much space is available in your new volume group by using the following smit path:

   ```
   smit
     System Storage Management
       Logical Volume Manager
         List Contents of a Volume Group
           VOLUME GROUP name     (mvs)

           Look for the line "Total PPs."  After the number of
           PPs, the MB size of the volume group is reported.
   ```

   You can multiply this MB number by 2048 to obtain the value for the size=nnnn parameter in the **crfs** command.

7. Use the command:

   ```
   lsfs
   ```

173 If this discussion is slightly confusing, you may need to review the AIX Logical Volume Manager (LVM) documentation.

to determine the logical volume name that was created for the new file system. In our case, it was /dev/lv00.

8. Mount the file system:

```
mount /dev/lv00 /mvs
```

9. Make a directory named os390 in the mvs file system:

```
cd /mvs
mkdir os390
```

Please note that nothing in the R390 design requires the use of a file system named mvs, or a subdirectory named os390. However, these elements are used by the standard OS/390 AD CD-ROM installation process. We suggest you use the same so that discussions based on the standard process will match your system.

10. Now we need to mount the first CD-ROM volume. We had already determined (by listing /dev) that the name of the drive was /dev/cd0. We first created a mount point for it:

```
cd /
mkdir /cdrom
```

This could be any name, but *cdrom* is widely used for this purpose. We the inserted the CD-ROM in the reader, and used this command:

```
mount -v cdrfs -r /dev/cd0 /cdrom
```

11. Use the *ls* command to explore the first CD-ROM. The files we want to move are in the subdirectory os390.

```
ls /cdrom
ls /cdrom/os390
```

12. Unzip the required files from the first CD-ROM, and place them in /mvs/os390:

```
/usr/bin/r390/unzip -j /cdrom/os390/os39r6.zip -d /mvs/os390
```

This will unzip the emulated OS39R6 volume, and will take at least 30 minutes. This zip file will inflate into two files, but the R/390 will handle these two as a single logical file. (The emulated 3390 volume is larger than 2GB; the 2GB file size limit in OS/2 and AIX required that emulated volumes larger than 2GB be implemented as multiple AIX or OS/2 files.)

We also need the devmap.mvs file from this CD-ROM, and it is not in ZIP format.

```
cp /cdrom/aix/devmap.mvs  /mvs/os390/devmap.mvs
```

Proceed to unzip all the volumes in the same way. For example:

```
/usr/bin/r390/unzip -j /cdrom/os390/os3r6a.zip -d /mvs/os390
```

13. Check your /mvs/os390 directory and determine if the file names are all lower case. AIX is case sensitive, but OS/2 is not. The creation process for the CD-ROMs sometimes produces files with upper-case names after they are unzipped.

```
cd /mvs/os390
ls
     devmap.mvs OS39R2_1.A80 OS39R2_2.A80 OS39H2.124
     SCPMV5.122 OS39RB.A82 P390DX.123 OS39D2_1.A81 OS39D2_2.A81
```

You can use files with upper case names, but most UNIX people are happier with lower case names. To change file names, do the following:

```
cd /mvs/os390
mv OS39R6_1.A80  os39r6_1.a80
mv OS39R6_2.A80  os39r6_2.a80
etc
```

If you elect to use lower case names, be certain you change all the letters. If you elected not to load some volumes your list will not have these files, of course.

14. Issue the commands:

```
cd /usr/bin/r390
awsprof
```

to verify that the current profile is /mvs/os390/devmap.mvs. If it is not, issue the command:

```
export CONFIG_FILE=/mvs/os390/devmap.mvs
```

14.12.3.1 R/390 Configuration Panel Updates

Before IPLing OS/390, you need to verify that the R/390 configuration file, the *device map* or *devmap*, is correct. If you followed the instructions above, you copied a devmap.mvs file from CD-ROM, and are pointing to it with the environment variable CONFIG_FILE. You cannot edit a devmap with a normal text editor. You can change it only with the R/390 configurator program. If you attempt to alter it with an editor, it will not be usable.

1. Start X windows, if you are not already running it. (You must be logged into AIX as *root* to use the R/390 subsystem.) The following instructions assume you are using basic X windows instead of CDE. You can use CDE, but you may need to modify these instructions slightly. Start the window with the R/390 Main Menu:

```
xinit              (to start X windows)
cd /usr/bin/r390   (always start r390 from here)
r390 &             (in the aixterm window)
```

2. Click on the P/390 Configuration icon. This should produce an initial P/390 Configurator panel that asks for a password. By default, there is no password; press Enter. The next menu should be the FUNCTIONS menu. We will work with F4 (Update System Environment) and F2 (Update System Devices) in this menu.

3. Press F4 to Update System Environment. This should produce a panel something like this:

```
Customer Name        >
Local Area NetID     >SYS1
System IPL Address   >A80
System Load Parm     >0122CS
System Mode          >ESA

System CPUID         >000001
Cache Size           >0
Expanded Store Size  >0
Character Font Name  >
```

You should enter a short name in the Customer Name field. This name is included in R/390 traces, and has no other function. The NetID is not relevant. The IPL address and Load Parm should be as shown, assuming you are using the OS/390 R2 AD CD-ROM system. Leave the other fields as shown. If you made any changes, press Enter and then F10 to return to the FUNCTIONS menu.

4. Press F2 to go to the Update System Devices panels. These panels define all the S/390 devices that will be seen by OS/390, and define the AIX files used to emulate 3380 and 3390 volumes. Your goal is to have a list that looks something like this:

```
    Addr Device Label  Atype Size   Mgr   FN/P
   >    >      >      >     >      >   >
 @  010  3278
    00C  2540                       6   /mvs/jcl*
    00E  1403          PRTR         5   /mvs/printer.00e
 @  063  3278
    122  3380   SCPMV5 CKD   1770C  2   /mvs/os390/scpmv5.122
    123  3380   P390DX CKD   177CC  2   /mvs/os390/p390dx.123
    124  3380   OS39H2 CKD    478C  2   /mvs/os390/os39h2.124
    560  3480                       J   /dev/rmt0
    580  3420B
    700  3278          DSPY         3   OS/390 Master Console
    701  3278          DSPY         3   Local TSO 1
    702  3278          DSPY         3   Local TSO 2
    900  3278          DSPY         3   Local TSO 3
    901  3278          DSPY         3   Local TSO 4
    A80  3390  OS39R2  CKD   3339C  2   /mvs/os390/os39r2_1.a80
    A81  3390  OS39D2  CKD   3339C  2   /mvs/os390/os39d2_1.a81
    A82  3390  OS39RB  CKD    500C  2   /mvs/os390/os39rb.a82
 @  E20  3088                       8
 @  E21  3088                       8
 @  E40  3088                       7
```

(The names shown here are several releases old, but the general format is the same as for current releases of the AD system.) Some lines in this panel will be shown in a very dark purple color; these lines are indicated by an “at symbol” (@) above. These devices are *turned off* and can be ignored. You can toggle a device on and off by moving the cursor to the appropriate line and pressing Cntl-F3 and Cntl-F5. OS/390 will not see a device that is turned off.

You can overtype parameters in the FN/P fields by moving the cursor to the field, changing it, and then pressing Enter. Make the changes indicated above, using the AIX file names you created earlier. The os39r2_1.a80 file will automatically be linked with the os39r2_2.a80 file; you do not specify this second file name.

You will have more devices than indicated in the list here. You can *turn off* the additional devices, or simply leave them. If they reference files that you did not create, there will be a few error messages when you start the IPL process, but you can usually ignore these.

You can create more devices by moving the cursor to the line with the greater-than symbols (>), entering appropriate fields, and pressing Enter.

An important note: The character in the Mgr column corresponds to one of the device managers listed at the bottom of the panel (and not shown in the example here). The character associated with a given device manager may change in different releases of the R/390 support programs. The correct device manager, not a given character, is the essential element. For example, 3380/3390 emulation is provided by the AWSCKD device manager.

AWSCKD is usually specified by Mgr "2," but this could change in future releases.

The configuration example shown above contains an entry for a card reader at address 00C. We strongly suggest you define this device as shown. It permits you to "submit" MVS jobs by copying the JCL (in ASCII) to the AIX file named in this definition. *Do not define a name consisting solely of "*" or "*.*," regardless of the descriptions in the R/390 DOC files.* Using the example above, a file copied to /mvs/jclxxxx (where xxxx can be anything) can be sent to the MVS card reader. See a more complete R/390 description for more information about the emulated card reader.

Changes to the Update System Devices panel(s) are not saved until you press F10 to exit from this panel, and then press F6 to exit from the Configuration function.

5. Press F10 to exit from the Update System Devices panel, and then press F6 to save all changes and exit from the configuration functions. This final F6 will cause your changes to be written to the current devmap, as identified by the CONFIG_FILE environmental variable.

14.12.4 IPL OS/390

{ZX48E} The default IPL process creates two windows containing 3270 emulators. Using a devmap similar to the one shown above, the first 3270 emulator will be associated with OS/390 address 700 and will become the MVS master console. The second 3270 window will be associated with OS/390 address 701 and can be used for TSO.

The IPL process will first attempt to use hcon for these 3270 windows. If hcon is not present, it will attempt to use X3270. If neither of these program products is installed in your AIX system, it will use a limited-purpose 3270 emulator named xant. The xant program is included on the R/390 support diskettes.

1. Click on the IPL icon. After a while, you should see two 3270 windows and the S/390 activity display (showing the PSW and a busy-state indicator). Within a minute or so, you should see the SPECIFY SYSTEM PARAMETERS message in one of the 3270 windows, and you can continue the IPL process in the usual way.

   There are noticeable delays during the IPL process, so be patient. The delays are to allow various processes to initialize themselves before the actual S/390 IPL function is started.

2. The default ENTER key for 3270 sessions (under both X windows and CDE) is the carriage-return key (the NL key). This is disconcerting to 3270 users (especially those using ISPF panels). This was the result of particular 3270 emulator startup options. See the redbook referenced earlier for a discussion of one of the default emulators.

3. The default 3270 sessions are restricted to 24-line screens and do not respond as expected to resizing. Resizing, by dragging the corner of a window, produces more blank space around the 3270 data lines, but does not make the font larger. Again, see the *P/390 and R/390 with OS/390: An Introduction* redbook for a brief discussion of tn3270 emulators.

4. No Send and Receive commands are available with the default xant 3270 emulator.

14.13 How Many Token Ring Adapters Do I Need?

{ZX50} 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:

1. LCS3172
2. LAN3172
3. OS/2 TCP/IP
4. OS/2 SNA (usually via CM/2)
5. LCS3172 and LAN3172
6. LCS3172 and OS/2 SNA
7. LAN3172 and OS/2 TCP/IP
8. LAN3172 and OS/2 SNA (if different SAPs) are used
9. OS/2 TCP/IP and OS/2 SNA
10. LCS3172 and LAN3172 and OS/2 SNA (different SAPs)
11. 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).

14.14 How Many Ethernet Adapters Do I Need?

{ZX51} 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                 (sometimes 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:

1. LCS3172 (with MPTS 802.2 set to either DIX or IEEE802.3. You probably want DIX)
2. LAN3172 (you probably want MPTS 802.2 set to 802.3, although some SNA systems can use DIX)[174]
3. 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.)
4. OS/2 SNA (probably CM/2). (You could set MPTS 802.2 to either 802.3 or DIX; you probably would use 802.3.)
5. 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.)
6. 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.)

[174] OS/2 and Windows NT (but not Windows 95) support SNA via DIX FrameTypes.

7. 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.)

8. 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.)

9. 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 this these are the default settings.)

10. 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.)

11. 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.

14.15 How to Display Square Brackets With PCOM

{ZX52} 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 pulldown 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 lefthand 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′.

14.16 How to Edit a PDS from OS/2

{ZX53} OS/390 volumes (for a P/390 based system) are emulated CKD devices. The emulator runs unser 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.[175] 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.

175 And it may change in the future. You should not produce any OS/2 or AIX software that depends on the internal format of the emulated CKD files.

14.17 OS/2 Does Not See My CD-ROM Drive

{ZX54} 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 sevice 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.)

14.18 How to Disconnect a Token Ring

{ZX55} 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.)

14.19 Should I Use HPFS386 For My P/390?

{ZX56} During a recent period 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.[176] 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:

1. 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.)
2. Use LAN Server Advanced Fixpack IP08508 (IP_8508) or later. (IP08506 is a prereq for IP08508).
3. 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.)
4. 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.

[176] AIX, for the R/390, can use large amounts of RISC memory as a disk cache and this has been shown to provide significant performance enhancement for the R/390 with some workloads.

14.20 How to Determine the Level of Your P/390 Programs

{ZX57} 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.

Internal IBM users can obtain the complete set of P/390 diskettes (seven 1.44 diskettes) in image form with the VM command:

```
TOOLS SENDTO PKVMWFB1 TOOLS P390 GET P390 PACKAGE
```

14.21 How to Remove PCOM′s Printer Startup

{ZX58} IBM′s Personal Communications S/400 and 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 \PROGRA1\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 [lt] 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.

14.22 How to Use PC Memory As S/390 Expanded Storage

{ZX59} 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:

1. 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.

2. Go to the F4 option of the P/390 configurator. Set the Expended 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.)

3. 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 32MB for expanded storage.

14.23 How to Install A Second EtherJet Adapter

{ZX60} 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.[177]

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[178] 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.)

177 The adapter we installed was 10/100 EtherJet PCI Adapter with Wake on LAN. We ignored the Wake on LAN feature.

178 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.

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).

14.24 Adjusting LAN Buffer Sizes

{ZX61} As distributed, LAN3172 cannot recieve a token ring SNA frame larger than 4096 bytes. You can alter this by:

1. Go to the P/390 configurator panel
2. Go to the F2 (devices) screen
3. Press F12 to go to the screen containing additional startup parameters for the device managers.
4. In the LAN3172 line, enter the parameter /RB=2048.
5. Exit (saving the changes) from the configurator and restart the P/390 subsystem.
6. You should be able to handle frame sizes up to 16000 bytes.

You must be at a recent level of P/390 software to do this. Note that this change is needed only in unusual situations.

A few installations may encounter “checksum” errors when using LCS3172 (device manager) and OS/390 TCP/IP with the integrated ethernet adapter in the PC Server 330 (and 325) systems. If this happens, try this:

1. Go to the P/390 configurator panel
2. Go to the F2 (devices) screen
3. Press F12 to go to the screen containing additional startup parameters for the device managers.
4. In the LCS3172 line, enter the parameter /RBEN 256.
5. Exit (saving the changes) from the configurator and restart the P/390 subsystem.

If you use a “watch window” in OS/2 that displays LCS3172 information. You should see lines like this:

```
LANA ADDRESS RECV FRAME 00000000 LOST/LOW 00000000 LOST/OUT 00000000
4000749001C8 XMIT FRAME 00000000 LOST/ERR 00000000 RCVBUF 00F8:00F8
```

The RCVBUF numbers should be in the range 00F4 - 00FC if your modification worked.

You can avoid stopping and restarting the P/390 subsystem (and reIPLing MVS) when working with the LAN adapter parameters. The following sequence usually works:

```
AWSSTAT /L                  see which managers are started
AWSSTAT LCS3172             stop LCS3172
AWSSTAT /L                  verify that it stopped
   change parameters
AWSSTART LCS3172.EXE n      start with normal parameters
AWSSTAT /L                  verify it started
```

The AWSSTAT command, followed by the name of a device manager, stops that device manager. You cannot stop AWSCKD, for example, because MVS will almost instantly complain about it. The LAN device managers can usually be stopped. The AWSSTART command will restart the device manager. The n in the example indicates that no OS/2 window should be associated with the program.

Chapter 15. Understanding Mainframe Printing

{ZY00} 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:

- Was a simple character record, with a maximum of 120 or 132 characters [179].
- Had a *carriage control tape* that could be used to skip to the top of the next page or to skip to any of several other positions on the page that were indicated by the current carriage control tape.[180] The different positions on the carriage control tape were 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.[181]
- 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

[179] The extra 12 character positions for the printer was an option on early 1403 printers. To this day, many utilities create 120 column output because the original, base 1403 printer had 120 columns

[180] An operator could change the carriage control tape, and different jobs might call for different carriage control tapes.

[181] JES2 contains an paramter to skip over page perforations if a print stream contains no controls of its own.

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 world.

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.

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.

15.1.1 AFP Data Streams

{ZY02} 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 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 segment pages into multiple records.

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 system. 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. Some of the IBM documentation is shipped in this format.

15.1.2 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 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 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 garbage than anything else on your printed page.

15.1.3 What Are AFP Resources?

{ZY03} AFP uses special resources to provide the advanced functions. These resources are:

- Formatting resources
  - A Form Definition is a resource mostly controlling 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
  - A 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 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.

    What you can specify in a page definition:
    - Positioning of the line data on a page, lines may be placed as whole entities, or the line may be split into fields and they then be placed. This includes also 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 so called 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 page where to skip with channel 1 you get more than one original application program output page to printed on one page.
    - Which font is used for printing; similarly to the positioning, you can specify it 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.[182]

- 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 some text (not only images). Fonts are called by overlays, 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 reserved for a page. The structure of AFP fonts will be discussed later.

An example of JCL to print a listing 4-up and duplex and using GT20 characters.

```
//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
/*
```

15.1.4 How Do I Create AFP Data Streams?

{ZY04} 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”) is a common source of AFP printing. These products are text formatters that, based on the coding inside the source text and profile definitions outside of the text file, creates composed pages as output.

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

[182] 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.

from what it is in OS/390. In 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 nowadays.

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) of it, and transferred it 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  501DA5000EFDEECCDFFFFFF
                            A063FF00021468574FFFFFF
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,

15.1.5 What Else Do I Get with AFP?

{ZY05} 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.

15.1.6 Fonts

{ZY06} 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:

- SYS1.FONTLIB library contains the most common fonts to be used for replacing traditional line printing.

With OS/390 the following AFP fonts are supplied:

- 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.

- 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.

15.1.7 How to Deal with Different Printer Resolutions

{ZY07} 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.

15.1.8 How to Consider NLS Printing

{ZY08} 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. See 15.3, “How to Print with a PC Laser Printer” on page 316.

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.

15.1.9 How to Print a File In OS/390

{ZY20} 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, it is, 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 a way or another. When you issue a print command, ISPF will put 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.

15.1.10 Printing from CICS

{ZY21} 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 shut down, the data set will be 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 printed 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 printed then by PSF/MVS.

15.1.11 How to Control Line Count

{ZY22} 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.

15.1.12 Controlling Printing

{ZY23} 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.

15.2 A Small SNA AFP Printer for P/390

{ZY24} If you have a 3174 control unit[183] available, either local or SNA-attached, and one of the older "small" AFP printers, such as a 3812, 3816, or 4028 unit, you can easily attach these to your P/390. Note that the P/390 device managers, used to emulate S/390 devices on PCs, do not provide AFP emulation. You need either a real APF printer or PSF/2, which does emulate AFP printing using a PC printer. Of course, there are newer AFP printers available. Some of them can be attached to 3174 units using coax attachment and some of them may be attached through token-ring.

As the most common case might be attaching a coax attached printer, either old or new, we give examples only how to specify them in VTAM definitions and PSF startup. If you want to attach a token-ring or channel attached printer directly to the MVS system, you may find useful information in the redbook *Printing with MVS on the IBM PC Server System/390*, SG24-4612.

[183] This is normally used for 3270 terminals, which are attached via coax to this control unit.

A remote 3174 may be attached using a token-ring attachment or an SDLC line. The following example shows how to set up such printers. In the example, proper coding for a VTAM logmode table entry is shown. You may check the most recent information in the PSF/MVS System Programmer's Guide.

To make the VTAM logmode operable, you have to assemble the logmode table with the entries you need. There is a sample JCL in the library IPO1.INSTLIB. The name of the member is VTMASM. The sample job assumes that your source is in the library IPO1.SAMPLIB. The job assembles the logmode table into the SYS1.VTAMLIB library.

15.2.1.1 Local Channel Attached 3174

This portion of the example assumes you have a channel-attached SNA 3174 connected to a S/370 Channel Emulator/A adapter on the P/390.

- VTAM Application Node

  For each AFP printer that is attached through VTAM, you must define an application that is referred to in the PSF start procedure

```
PSFAPPL2 VBUILD TYPE=APPL
P390PSF1 APPL  AUTH=ACQ,EAS=1,SONSCIP=YES
```

- VTAM Major Local Node

  The definition for a local 3174 SNA unit could be:

```
SNA080  VBUILD TYPE=LOCAL
SNA080PU PU CUADDR=080,MAXBFRU=20,PUTYPE=2,ISTATUS=ACTIVE

SNA08003 LU LOCADDR=3,MODETAB=MIBM3812,DLOGMOD=IBM3812C
```

- VTAM Logmode

```
MIBM3812 MODETAB
IBM3812C MODEENT LOGMODE=IBM3812C,                                   +
               FMPROF=X'03',                                         +
               TSPROF=X'03',                                         +
               PRIPROT=X'B1',                                        +
               SECPROT=X'B0',                                        +
               COMPROT=X'7080',                                      +
               RUSIZES=X'85C7',                                      +
               PSNDPAC=X'02',                                        +
               SRCVPAC=X'02',                                        +
               SSNDPAC=X'00',                                        +
               PSERVIC=X'014000010000000001000000'
         MODEEND
         END
```

- PSF Procedure

```
//PSF3812  PROC
//**************** THE PSF SNA WRITER PROCEDURE ***************
//*
//*
//STEP01   EXEC PGM=APSPPIEP,REGION=1750K
//JOBHDR OUTPUT PAGEDEF=V06483,     /* JOB SEPARATOR PAGEDEF   */
//       FORMDEF=A10110,CHARS=GT15  /* JOB SEPARATOR FORMDEF   */
//JOBTLR OUTPUT PAGEDEF=V06483,     /* JOB SEPARATOR PAGEDEF   */
//       FORMDEF=A10110,CHARS=GT15  /* JOB SEPARATOR FORMDEF   */
//DSHDR  OUTPUT PAGEDEF=V06483,     /* DS  SEPARATOR PAGEDEF   */
//       FORMDEF=A10110,CHARS=GT15  /* DS  SEPARATOR FORMDEF   */
//MSGDS  OUTPUT PAGEDEF=A06462,     /* MESSAGE DATASET PAGEDEF */
```

```
//       FORMDEF=A10110,CHARS=GT15  /* MESSAGE DATASET FORMDEF */
//FONT01   DD   DSN=SYS1.FONTLIBB,  /* SYSTEM FONTS           */
//         DISP=SHR
//PSEG01   DD   DSN=SYS1.PSEGLIB,   /* SYSTEM PAGE SEGMENTS   */
//         DISP=SHR
//OLAY01   DD   DSN=SYS1.OVERLIB,   /* SYSTEM MEDIUM OVERLAYS  */
//         DISP=SHR                 /*                        */
//PDEF01   DD   DSN=SYS1.PDEFLIB,   /* SYSTEM PAGEDEFS        */
//         DISP=SHR
//FDEF01   DD   DSN=SYS1.FDEFLIB,   /* SYSTEM FORMDEFS        */
//         DISP=SHR
//* ***********************************************************/
//*             PRINTDEV                                      */
//* ***********************************************************/
//PRT7     CNTL
//PRT7     PRINTDEV FONTDD=*.FONT01, /* FONT     LIBRARY DD    */
//          OVLYDD=*.OLAY01,         /* OVERLAY  LIBRARY DD    */
//          PSEGDD=*.PSEG01,         /* SEGMENT  LIBRARY DD    */
//          PDEFDD=*.PDEF01,         /* PAGEDEF  LIBRARY DD    */
//          FDEFDD=*.FDEF01,         /* FORMDEF  LIBRARY DD    */
//          JOBHDR=*.JOBHDR,         /* JOB HEADER SEPARATOR   */
//*                                  /*   OUTPUT               */
//          JOBTRLR=*.JOBTLR,        /* JOB TRAILER SEPARATOR  */
//*                                  /*   OUTPUT               */
//          DSHDR=*.DSHDR,           /* DATA SET HEADER        */
//*                                  /*   SEPARATOR            */
//          MESSAGE=*.MSGDS,         /* MESSAGE DATA SET OUTPUT*/
//          PAGEDEF=V06483,          /* DEVICE PAGEDEF DEFAULT */
//          FORMDEF=A10110,          /* DEVICE FORMDEF DEFAULT */
//          CHARS=(GT12,             /* DEVICE                 */
//          GT10,GT15,GT18),         /* DEFAULT FONT SET       */
//          PIMSG=YES,               /* ACCUMULATE DATA SET    */
//*                                  /*   MESSAGES             */
//          DATACK=UNBLOCK,          /* REPORT ALL DATA-CHECK  */
//*                                  /*   ERRORS               */
//          TRACE=NO,                /* CREATE INTERNAL TRACE  */
//          FAILURE=WCONNECT,        /* PSF ACTION ON PRINTER  */
//*                                  /*   FAILURE              */
//          TIMEOUT=REDRIVE,         /* PSF ACTION ON TIMEOUT  */
//          DISCINTV=1800,           /*DISCONNECT INTERVAL IN  */
//*                                  /*   SECONDS              */
//          APPLID=P390PSF1,         /*VTAM APPLICATION PGM    */
//*                                  /*   NODE NAME            */
//*                                  /*IS IN THE PU/LU DEFINTIONS*/
//          MGMTMODE=IMMED,          /*PRINTER MANAGEMENT MODE */
//          LUNAME=SNA08003          /* VTAM LOGICAL UNIT NAME */
//PRT7      ENDCNTL
```

If you are using a 240 pel printer, the font libary should be SYS1.FONTLIBB as in the JCL above. If you use a 300 pel printer, then the font library should be SYS1.FONT300.

15.2.1.2 Remote Token-Ring Attached 3174

If the 3174 is attached to a token ring, you could use the following definitions:

- VTAM Application Node

```
PSFAPPL2 VBUILD TYPE=APPL
P390PSF1 APPL  AUTH=ACQ,EAS=1,SONSCIP=YES
```

- VTAM Major Node

```
SL3172MN VBUILD TYPE=XCA
SL3172PU PORT  CUADDR=E20,                                          C
               DELAY=0,                                             C
               ADAPNO=0,                                            C
               MEDIUM=RING,                                         C
               SAPADDR=4,                                           C
               TIMER=60
*
GRP2172  GROUP DIAL=YES,DYNPU=YES,ANSWER=ON,ISTATUS=ACTIVE,CALL=INOUT
SLL2     LINE
SLP2     PU
SLL3     LINE
SLP3     PU

3174RTR VBUILD TYPE=SWNET
SLP3174R PU CUADDR=080,MAXBFRU=20,PUTYPE=2,ISTATUS=ACTIVE
SNA08004 LU LOCADDR=3,MODETAB=MIBM3812,DLOGMOD=IBM3812T
```

- VTAM Logmode

```
MIBM3812 MODETAB
IBM3812T MODEENT LOGMODE=IBM3812T,                                  +
               FMPROF=X'03',                                        +
               TSPROF=X'03',                                        +
               PRIPROT=X'B1',                                       +
               SECPROT=X'B0',                                       +
               COMPROT=X'7080',                                     +
               RUSIZES=X'8787',                                     +
               PSNDPAC=X'10',                                       +
               SRCVPAC=X'10',                                       +
               SSNDPAC=X'00',                                       +
               PSERVIC=X'014000010000000001000000'
         MODEEND
         END
```

- PSF Startup is exactly similar to the one above, the only thing you have to change is the LUNAME and possibly the APPLID.

15.2.1.3 Remote SDLC Line Attached 3174

Setup for an SDLC connection is almost the same. See 12.5, "How to Connect a Simple SDLC 3174" on page 212 for more details about SDLC 3174 usage.

- VTAM Application Node

  See the example above.

- VTAM Major Node See the example in above. You just add the following LU below the PU

```
SNA08004 LU LOCADDR=6,MODETAB=MIBM3812,DLOGMOD=IBM3812R
```

- VTAM Logmode

```
MIBM3812 MODETAB
IBM3812R MODEENT LOGMODE=IBM3812R,                                  +
               FMPROF=X'03',                                        +
               TSPROF=X'03',                                        +
               PRIPROT=X'B1',                                       +
               SECPROT=X'B0',                                       +
               COMPROT=X'7080',                                     +
               RUSIZES=X'8585',                                     +
```

```
          PSNDPAC=X'03',                                    +
          SRCVPAC=X'03',                                    +
          SSNDPAC=X'00',                                    +
          PSERVIC=X'014000010000000001000000'
      MODEEND
      END
```

- PSF Startup
- PSF Startup is exactly similar to the one above, the only thing you have to change is the LUNAME and possibly the APPLID.

15.2.2 What to Do When Something Goes Wrong?

{ZY25} Typical errors with printing are, for example, problems with national language characters, printing off of 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 then 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 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 is exceeding 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.

15.2.3 Handling AFP Errors and Error Messages

{ZY26} In the PSF procedure for each printer, you can specify keyword parameters PIMSG and DATACK. 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 DATACK=UNBLOCK, all data checks, whether they are caused by invalid characters or exceeding the boundaries, are reported to PSF. If you set DATACK=BLKCHAR, only errors caused by exceeding the boundaries are reported, invalid character errors are blocked. If you set DATACK=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, DATACK=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.

15.3 How to Print with a PC Laser Printer

{ZY30} *This section is an update to the information in* P/390 and R/390 with OS/390: An Introduction *(IBM order number SG24-4847). It includes some of the text from the original document, with new information added.*

"Printed" output, typically from JES2, can be handled by the AWS2821[184] 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.

This description is primarily for the P/390. Differences for the R/390 are noted at the end.

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

[184] This device manager is named for the IBM 2821 control unit, which was part of the original S/360 series. The 2821 controlled card readers, punches, and printers.

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 a 2821, 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\PRTCTL.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\PRT.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 was recently contributed to one of the P390 FORUMs, and 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.[185]

15.4 R/390 Printing

{ZY31} Recent releases of the R/390 support programs have greatly expanded the printing functions. These are now at least equivalent to the P/390 printing capabilities. We strongly suggest you read the DOC files with the most recent releases of the R/390 support diskettes for more information.

One key difference must be stressed: the P/390 device manager (AWS2821) detects the end of a job′s output by either (1) recognizing an EOJ string, or (2) receiving no more output from the S/390 for at least 10 seconds. The R/390 equivalent program uses only EOJ string recognition, although it can check for many different (user defined) EOJ strings.

The different EOJ strings can be used to direct output to different AIX queues.

[185] 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.

Appendix A. Special Notices

This publication is intended to help users of the OS/390 AD CD-ROM systems better understand their systems The information in this publication is not intended as the specification of any programming interfaces that are provided by OS/390 and any associated products.

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 the International Business Machines Corporation in the United States and/or other countries:

ACF/VTAM	AD/Cycle
Advanced Function Printing	AFP
AIX	AnyNet
APPN	AS/400
BatchPipes	BookManager
BookMaster	C++/MVS
C/MVS	C/370

CBIPO	CBPDO
CICS	CICS/ESA
CICSPlex	COBOL/370
C/370	DATABASE 2
DB2	DB2 Connect
DB2 Extenders	DB2 Client Application Enablers
DB2 OLAP Server	DB2 Universal Database
DFDSM	DFSMS
DFSMS/MVS	DFSMSdfp
DFSMSdss	DFSMShsm
DFSMSrmm	DFSORT
DRDA	ECKD
eNetwork	Enterprise System/3090
Enterprise System/4381	Enterprise System/9000
Enterprise System/9370	Enterprise Systems Architecture/370
Enterprise Systems Architecture/390	ES/3090
ES/4381	ES/9000
ES/9370	ESA/370
ESA/390	ESCON XDF
ESCON	EtherJet
EtherStreamer	FFST
FFST/2	GDDM
Hiperbatch	Hiperspace
IBM	IBMLink
IMS	IMS/ESA
IP PrintWay	Magstar
Micro Channel	MQ
MQSeries	MVS/DFP
MVS/ESA	MVS/SP
MVS/XA	Net.Data
Netfinity	NetRexx
NetSpool	NetView
NETWORK STATION	Nways
OpenEdition	Operating System/2
OS/2	OS/390
Parallel Sysplex	PR/SM
Print Services Facility	PrintManager
PrintWay	Processor Resource/Systems Manager
PS/2	PSF
QMF	RACF
RAMAC	RETAIN
RISC System/6000	RMF
S/370	S/390
SOM	SupportPac
Sysplex Timer	System Object Model
System/360	System/370
System/390	SystemPac
VisualAge	VisualGen
VisualInfo	VisualLift
VTAM	WebExplorer

The following product names are trademarks of Tivoli Systems, Inc.:

Tivoli	NetView

The following product names are trademarks of Lotus Development Corporation:

Lotus	Ami Pro
Freelance	Freelance Graphics
Lotus Notes	SmartSuite
Domino	eSuite
Lotus Go Webserver	Notes

The following terms are trademarks of other companies:

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

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

Other company, product, and service names may be trademarks or service marks of others.

Appendix B. Related Publications

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

B.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see "How to Get ITSO Redbooks" on page 327.

- *P/390 (and R/390) OS/390: New User's Cookbook*, SG24-4757
- *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 Implementation Guide Volume 2 - OpenEdition Applications*, SG24-5228
- *OS/390 eNetwork Communications Server V2R5: TCP/IP Implementation Guide Volume 3 - MVS Applications*, SG24-5229
- *P/390 and R/390 with OS/390: An Introduction*, SG24-4847
- *Connectivity on a PC Server System/390*, SG24-4624
- *Printing with MVS on the IBM PC Server System/390*, SG24-4612

B.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates two to four times a year at significant savings.

CD-ROM Title	Collection Kit Number
System/390 Redbooks Collection	SK2T-2177
Networking and Systems Management Redbooks Collection	SK2T-6022
Transaction Processing and Data Management Redbook	SK2T-8038
Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
AS/400 Redbooks Collection	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SK2T-8041
RS/6000 Redbooks Collection (PDF Format)	SK2T-8043
Application Development Redbooks Collection	SK2T-8037

B.3 Other Publications

These publications are also relevant as further information sources:

- *OS/390 eNetwork Communications Server: IP Configuration Version 2 Release 5*, SC31-8513
- *OS/390 Network File System Customization and Operation*, SC26-7253
- *OS/390 Network File System User's Guide*, SC26-7254
- *IBM PC Server 500 System/390 Installation, Configuration, and User's Guide for MVS/ESA*, SA22-7210
- *IBM PC Server 500 System/390 Messages and Codes* , SA22-7227

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** http://www.redbooks.ibm.com/

  Search for, view, download or order hardcopy/CD-ROMs redbooks from the redbooks Web site. Also read redpieces and 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.

- **E-mail Orders**

  Send orders via e-mail including information from the redbook fax order form to:

In United States:	e-mail address: usib6fpl@ibmmail.com
Outside North America:	Contact information is in the "How to Order" section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl/

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the "How to Order" section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl/

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the "How to Order" section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl/

This information was current at the time of publication, but is continually subject to change. The latest information for customers may be found at http://www.redbooks.ibm.com/ and for IBM employees at http://w3.itso.ibm.com/.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may also view redbook, residency and workshop announcements at http://inews.ibm.com/.

IBM Redbook Fax Order Form

Please send me the following:

Title	**Order Number**	**Quantity**

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number

- Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Index

Special Characters

Numerics

A

E

F

G

H

I

J

R

S

T

ITSO Redbook Evaluation

P/390, R/390, S/390 Integrated Server: OS/390 New User's Cookbook
SG24-4757-01

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.ibm.com
- Fax this form to: USA International Access Code 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?
__Customer   __Business Partner   __Solution Developer   __IBM employee
__None of the above

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction ____________

Please answer the following questions:

Was this redbook published in time for your needs?   Yes____ No____

If no, please explain:
____

____

____

____

What other redbooks would you like to see published?
____

____

____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)
____

____

____

____

____

SG24-4757-01
Printed in the U.S.A.

P/390, R/390, S/390 Integrated Server: OS/390 New User's Cookbook

SG24-4757-01