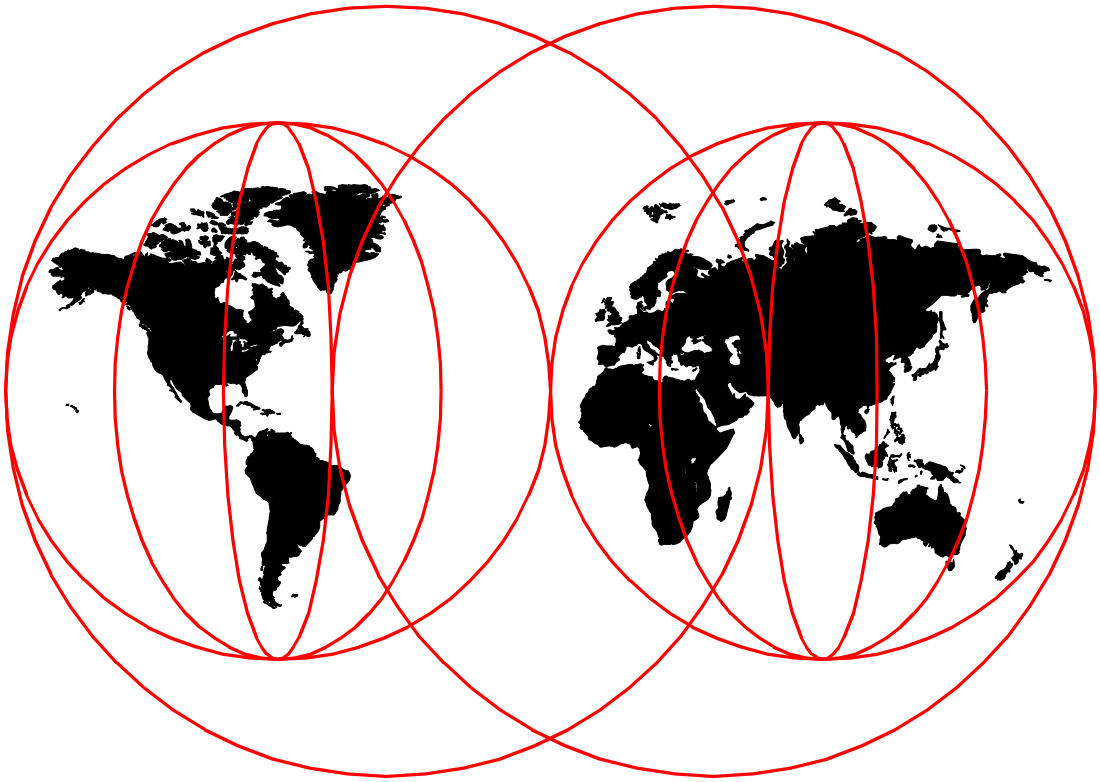
Experiences with Migrating Oracle to OS/390

Kathryn Arrell, Alain Atge, Lou Doran, Dennis Dutcavich
Michel Gayraud, Steve Hwung, Ramona Townsend



International Technical Support Organization

www.redbooks.ibm.com

SG24-4981-00

International Technical Support Organization

SG24-4981-00

Experiences with Migrating Oracle to OS/390

June 2000

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix E, “Special notices” on page 115.

First Edition (June 2000)

This edition applies to the Oracle8 (and Oracle 8*i*) Database Server with Oracle Applications Release 11 running on OS/390 V2R6.

The document created or updated on 28.06.2000.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

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

Contents

Figures

Preface

This redbook helps you to do the following:

- Upgrade an Oracle database from Oracle7 on S/390 to Oracle8 on S/390 using the Oracle Migration utility
- Migrate an Oracle database from a UNIX platform to OS/390 using the Oracle Export/Import utilities
- Upgrade Oracle Applications from 10.7 NCA to Release 11 NCA
- Install Oracle8*i* Release 8.1.5 on OS/390

This redbook will be useful to those migrating a database to Oracle on OS/390 for the first time.

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Kathryn Arrell is an ERP Specialist at the International Technical Support Organization, Poughkeepsie Center.

Alain Atge is a S/390 Specialist with the EMEA New Technology Center in Montpellier. He works in the ERP Business Segment and provides technical presales support for Oracle Applications on OS/390.

Louis Doran is an ERP Solutions Specialist with the North American ERP Competency Center in Philadelphia. Before joining the ERP Group, Lou worked in Network Marketing and has a background in large systems.

Dennis Dutcavich is a Software Engineer with the S/390 division. He works in the ERP Business Segment and provides technical marketing support for Oracle Applications on OS/390.

Michel Gayraud is a Specialist System Engineer in France. He has worked at IBM since 1982. He has eight years of experience as a marketing representative in large MVS accounts and three years as a networking specialist in the IBM Networking division. He currently works in the Oracle Joint Solution Center in Montpellier.

Steve Hwung is a Database Specialist with IBM Global Services on assignment to the IBM/Oracle Competency Center in Redwood Shores.

Ramona Townsend is a Database Specialist working in S/390 Technical Support.

Thanks to the following people for their invaluable contributions to this project:

Bob Haimowitz
Richard Conway
Vasilis Karras
International Technical Support Organization, Poughkeepsie Center

Tom Holiday
IBM Global Services

Susan MacKinnon
Oracle UK

Jim Gillespie
ORAMAIN Consulting Services, LLC

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 “IBM Redbooks review” on page 123 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. Overview

This redbook describes the following:

- How to upgrade an Oracle database from Oracle7 on S/390 to Oracle8 on S/390 using the Oracle Migration utility
- How to migrate an Oracle database from a UNIX platform to OS/390 using the Oracle Export/Import utilities
- How to upgrade Oracle Applications from 10.7 NCA to Release 11 NCA using adaimgr
- How to migrate an Oracle database that supports applications written either by customers or by application providers that support their applications on OS/390

1.1 Methods of migrating the Oracle database

Oracle provides three methods to migrate the Oracle database:

- Oracle Migration utility. This utility migrates previous levels of the Oracle database to Oracle8. Using this utility is the quickest way to migrate Oracle. However, you must migrate the entire database as it exists when you perform the migration. The Migration utility only works if you are migrating the database on the same platform. You cannot use this method to change platforms. Not all previous levels of the database can be migrated using the Migration utility. Check the *Oracle8 for OS/390 Migration Guide* to determine whether your database can be migrated using this utility.

  Note

  The Migration utility does not allow you to change the character set that was used when the database was created. This can be a consideration if you are moving Oracle ERP Applications from Oracle7 to Oracle8, since the character set is changed from WEBCDIC37C to WEBCDIC1047.

  If you do want to change character sets, you must use Export/Import utilities or the COPY method.

- Export/Import utilities. The Oracle database can be migrated using the Export and Import utilities that are provided with the database. Normal considerations that apply to any export/import apply here as well. This method can be used to migrate specific parts of the database.
- Copying data. This method involves using the SQL*Plus COPY command, the SQL CREATE TABLE command using FROM or AS SELECT FROM clauses, or SQL INSERT INTO command. This method can be accomplished by using database links.

The advantage of using the Oracle Migration utility is in its simplicity and speed. All the necessary planning, testing, and practice migrations should still take place, but this method will prove to be the quickest and requires the least skill.

The Export and Import utilities will allow you to move parts of the database. Greater DBA skills are required, and the time needed to do an import may make this method impractical.

Copying data offers a more granular migration. Specific objects can be migrated, as with exports and imports. Additionally, specific rows within tables can be migrated. This method requires very specific knowledge of your database schemas and SQL or SQL*Plus. This method can also take a long time to execute.

For most installations, you may find that using the Migration utility is most practical. You should spend the time needed to understand all three methods so that you can determine which one best fits your needs. Then, you should plan for and practice the migration after insuring that you have a current backup of the database you will migrate. If you make a mistake during the migration, it is very likely that the current copy of the database will be lost forever. Make sure you have a backup.

We have provided examples of experiences using the Migration utility and Export/Import utilities. We chose not to use the COPY method and we are not aware of any customers that have chosen this method.

Part 1. Upgrading the Oracle Database on OS/390

This section covers our experiences for planning and upgrading the Oracle Database on OS/390 using the Oracle Migration utility.

Chapter 2. Using the Oracle Migration utility on OS/390

If your Oracle7 database is already on S/390, you can use the Oracle Migration utility to migrate the database to Oracle8. We followed the directions using two Oracle documents, the *Oracle8 for OS/390 Migration Guide* and Chapter 11 of *Oracle8 for OS/390 Installation Guide*. The first document provides the overall procedure (and is targeted at UNIX); the second document provides the specific information needed to run the Migration utility on OS/390.

The OS/390 document provides pre-migration and migration checklists. The most important tasks listed are to plan and rehearse the migration. During the process, we made several simple mistakes that required us to restart the process from the beginning.

For test purposes, we selected the Oracle Applications Release 10.7 SmartClient Global Demo Database to migrate. It is about 10 GB of data and it contains approximately 9,000 database objects, other than tables, such as stored procedures, triggers, views, functions, and so forth. The actual migration process, executing the procedures needed to migrate the database, took less than three hours to complete.

2.1 Migration overview

As we went through the migration process, it became evident that we needed to better understand the bigger picture—particularly what actually had to be done to prepare a new Oracle8 database for the migration from Oracle7.

We had two choices of how to build an empty Oracle8 database. The *Oracle8 for OS/390 Migration Guide* suggests you build the new database using the install process as if this was the first Oracle8 database being installed on the system. Since we already had Oracle8 installed, we chose to create a secondary instance using the installation panels. Either way, the overall process is the same.

Note

You do not create the database instance until later.

The “big picture” steps of the actual migration process are as follows:

1. Create the Oracle libraries for an Oracle8 database. You follow all the steps to install and create an instance except creating the database. This will be done much later. This is a very important step. If you create the database, the migration will fail and the Oracle7 database will no longer function and will need to be recreated.
2. Run the Migration utility with `CHECK_ONLY=TRUE`. This will determine if you have enough space in the System tablespace for the new data dictionary.
3. Run the Migration utility without the `CHECK_ONLY=TRUE` parameter. This only creates a conversion file that is used during the actual migration to Oracle8.
4. Create new control files, a SNAPCF data set, and ensure the INITORA member has the correct parameters installed. The SNAPCF data set is used by the Recovery Manager during database recovery.
5. Start the new Oracle8 database. This creates the new database instance.
6. Stop the database.

7. Start the database and run the post-installation jobs.

At this point, we applied Oracle patch sets to the new database.

2.2 Migration steps

The following sections contain the details of the procedures and the problems we encountered when we migrated the Global Demo Database from Oracle7 to Oracle8.

2.2.1 Pre-migration

The *Oracle8 for OS/390 Migration Guide* and the *Oracle8 for OS/390 Installation Guide* provide a comprehensive list of tasks to complete prior to migrating a database, which includes making a copy of the current database in case the migration goes awry.

Consider this additional step. Prior to migration, determine what invalid objects exist in the old database. When you migrate, all the objects, such as views, procedures, triggers, packages, and so forth, will be invalid and will need to be recompiled. You should have a list of invalid objects before you start to facilitate any diagnostics.

We created the script in Figure 1 to determine the invalid objects in the database and ran it from Windows using SQL*Plus. This could be done from TSO as well.

```
-- determine the invalid objects in the data base

COLUMN owner FORMAT a20
COLUMN object_name FORMAT a35

SPOOL INVLD_OBJS.lst
SELECT  object_name, owner, object_type
FROM    dba_objects
WHERE   status='INVALID'
ORDER BY object_type;

SPOOL OFF
```

Figure 1. Script to find invalid database objects

We were unaware that database objects would be invalidated as a result of the migration. This fact doesn't seem to be documented. When we ended up with over 9000 invalid objects, we thought we did something wrong and then performed another migration. We had the same results. This time we recompiled the invalid objects.

2.2.2 Migration

The next step makes an assumption that you have chosen a way to create the new Oracle8 database. The choices, based on the following chapters of the *Oracle8 for OS/390 Installation Guide*, are as follows:

- To create a database from scratch using either Chapter 4, "Installation Without SMP," or Chapter 5, "Installation With SMP/e"
- Or using Chapter 10, "Installing Multiple Oracle Subsystems"

We chose the last method. It is simple and quick.

During the migration process, we had to refer back and forth between the *Oracle8 for OS/390 Migration Guide* and the *Oracle8 for OS/390 Installation Guide*. This can get confusing at times. We recommend that you start with Chapter 3 in the *Oracle8 Migration Guide*.

Follow these steps:

1. Read through Chapter 3 to the “Prepare the Version 7 Source Database for Migration” section. Perform the steps listed in that section. The purpose of these steps is to insure that there is no user with the name MIGRATE and to determine if there are any changes that need to be performed with the rollback segments.
   - At this point, we moved to the *Oracle8 for OS/390 Installation Guide*. There is a section in this document called “Review Version 8 Migration Utility Command Line Options.” We recommend that you bookmark this section because later in the process, the list of options will be needed for the JCL used to run the Migration utility.
2. Go to the *Oracle8 for OS/390 Installation Guide*, Chapter 11. The process to migrate the database starts in step 2 of the migration steps. Read and familiarize yourself with the previous steps in this chapter. You can use the method suggested in step 1, or you can install a secondary instance using Chapter 10 in this book. We chose the latter.

In Chapter 11, step 2.1 “Using the Migration utility” states that you should not run the installation job ORPIJE00. If you chose the secondary instance method, the job will be ORSIJE00, where the S indicates this is for a secondary instance.

In either case, this job (ORSIJE00) will allocate all the required VSAM files, such as the control files, redo log files, and so on. *It is very important not to run this job at this point in the sequence, as the redo and the database files from the earlier version are re-used and the control files will be allocated later.*

The instructions have the user examine the ORPIJFxx jobs and run those that will not interfere with the installation. If you used the secondary instance method, there will only be one Fxx job, ORSIJF00. This job (ORSIJF00) will create the following members where xxxx is the instance name:

- **MPxxxx** is the MPMPARM member.
- **OIxxxx** is the InitOra for this instance.
- **STRTxxxx** is the member used with the Oracle start proc to start the database.
- **BLDGxxxx** is the member used to create the database the first time the instance is started.
- **ORAxxxx** is the start proc.

If a normal install is chosen, you will see some or all of the following, depending on the options selected during the installation process. The members we had from the original installation were:

- **ORPIJF01** - Set up the Oracle parmlib
- **ORPIJF05-F08** - Receive the members into the user’s parmlib

We ran the ORSIJF00 job to create the parmlib members. *It is very important to not start the new Oracle8 database at this point as you first have to check if you need additional space in the system tablespace.* We used this JCL to check the requirements of the system tablespace to insure we had enough space allocated. See Figure 2 on page 9.

```
//*  NOTE:   1. //CONTROL1 - V7 CONTROL FILE
//*          1. //CONTROL2 - V7 CONTROL FILE
//*          2. //INITORA  - V7 INITORA FILE
//*          3. //MIGRATE  - V7 -> V8 MIGRATION SQL SCRIPT
//*          4. //CNVFILE  - V7 -> V8 MIGRATION OUTPUT
//*                        ( TO BE USED BY V8 STARTUP JCL )
//*****************************************************************
//*  STEP1 :
//*   Run V7 to V8 migrate sql
//*   DUMP V7 CONTROL FILE to /DD/CNVFILE
//*****************************************************************
//MIGRATE  EXEC PGM=MIG,PARM='++/DD/SYSIN'
//STEPLIB   DD  DISP=SHR,DSN=MPMN.ORA804.AUTHLOAD
//ORA@GDDB  DD  DUMMY
//*
//CONTROL1  DD  DISP=SHR,DSN=GDDB.CONTROL1
//CONTROL2  DD  DISP=SHR,DSN=GDDB.CONTROL2
//INITORA   DD  DISP=SHR,DSN=MPM2.ORACLE.PARMLIB(INITGDDB)
//MIGRATE   DD  DISP=SHR,DSN=MPMN.ORA804.SQL(MIGRATE)
//*
//CNVFILE   DD  DSN=GDDB.CNVFILE,DISP=(,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(4096,(12,6)),DCB=(RECFM=FB,BLKSIZE=4096),
// VOL=SER=ORACL9
//*DB1       DD  //* Include those database and log datasets
//*DB2       DD  //* referenced with DD name by Oracle. Refer to
//*LOG1      DD  //* your Oracle7 startup JCL.
//*
//SYSTEM1  DD DSN=GDDB.SYSTEM1,DISP=SHR
//DATA1    DD DSN=GDDB.DATA1,DISP=SHR
//DATA2    DD DSN=GDDB.DATA2,DISP=SHR
//DATA3    DD DSN=GDDB.DATA3,DISP=SHR
//DATA4    DD DSN=GDDB.DATA4,DISP=SHR
//DATA5    DD DSN=GDDB.DATA5,DISP=SHR
//INDEX1   DD DSN=GDDB.INDEX1,DISP=SHR
//INDEX2   DD DSN=GDDB.INDEX2,DISP=SHR
//INDEX3   DD DSN=GDDB.INDEX3,DISP=SHR
//INDEX4   DD DSN=GDDB.INDEX4,DISP=SHR
//INDEX5   DD DSN=GDDB.INDEX5,DISP=SHR
//RBS      DD DSN=GDDB.RBS,DISP=SHR
//TEMP     DD DSN=GDDB.TEMP,DISP=SHR
//OFFICE   DD DSN=GDDB.OFFICE,DISP=SHR
//CONTROL1 DD DSN=GDDB.CONTROL1,DISP=SHR
//CONTROL2 DD DSN=GDDB.CONTROL2,DISP=SHR
//LOG1     DD DSN=GDDB.LOG1,DISP=SHR
//LOG2     DD DSN=GDDB.LOG2,DISP=SHR
//LOG3     DD DSN=GDDB.LOG3,DISP=SHR
//SPOOL     DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//SYSERR    DD  SYSOUT=*
//      PEND
//    EXEC PGM=IEFBR14
//DD1 DD DSN=GDDB.CNVFILE,DISP=(OLD,DELETE)
//EXEC MIGRATE
//SYSIN     DD  *
 CHECK_ONLY=TRUE
 DBNAME=GB107
NEW_DBNAME=GDB8 MULTIPLIER=20
```

Figure 2. ORSIJF00 job

Insure that you use the parameter `CHECK_ONLY=TRUE` so that the job only calculates space requirements and does not perform the actual migration. Information on CHECK_ONLY can be found in the *Oracle8 for OS/390 Migration Guide* under

"System Considerations and Requirements, Space Requirements." See Figure 3 for the sample output from running the Migration utility in CHECK_ONLY. The space estimates will be found in the output of the job.

```
*******************************LINES DELETED*************************************
#^ connect (migrate, migrate)
#^ create_cluster_ts()
#^ connect (internal)
#^ space_out(
estimated space requirement for V8 version of V7 catalog objects is 113951 block
estimated space requirement for new V8 catalog objects is 237 blocks
estimated space requirement for total V8 catalog is 115034 blocks
free space found in system tablespace is 75797 blocks
insufficient space for new dictionaries, 471179264 bytes needed, 310464512 found
) v8 catalog space requirement:  471179264
free space found:  310464512
******************************** BOTTOM OF DATA ********************************
```

Figure 3. Output from Migration utility with CHECK_ONLY=TRUE

As a result of running the Migration utility in CHECK_ONLY=TRUE mode, we added another data set to the system tablespace of the new database. Figure 4 on page 11 shows a sample of the JCL we used.

```
//ADDSYS5 JOB (0000,OR),'ORACLE INSTALL',CLASS=A,
//           MSGCLASS=X,PRTY=15,MSGLEVEL=(1,1),NOTIFY=DUTCH
//*  JOB DESCRIPTION: ALLOCATE VSAM FILE AND RUN CCF ONIT THEN     *
//*                   ADD TO DATABASE.                             *
//*-----------------------------------------------------------------*
//*
//STEPAMS1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  /*-------------------------------------------------------*/
  /*                                                       */
  /*       DELETE THE CLUSTER                              */
/*--------------------------------------------------------*/
  DELETE (GDDB.SYSTEM5 -
        CLUSTER PURGE
  SET MAXCC=0    /* REMOVE AFTER FIRST TIME EXECUTION */
  SET LASTCC=0   /* REMOVE AFTER FIRST TIME EXECUTION */
/*--------------------------------------------------------*/
/*                                                        */
/*       DEFINE THE DB (ORACLE DATA BASE) CLUSTERS.       */
/*                                                        */
/*--------------------------------------------------------*/
  DEFINE CLUSTER                                    -
    (                                               -
     NAME(GDDB.SYSTEM5)                             -
         VOLUMES(ORACLA)                            -
         CONTROLINTERVALSIZE(4096)                  -
         MEGABYTES(100)                             -
         RECORDSIZE(4089 4089)                      -
         NONSPANNED                                 -
UNIQUE                                      -
           NONINDEXED                                 -
           SHR(3 3)                                   -
      )                                               -
      DATA                                            -
      (                                               -
      NAME(GDDB.SYSTEM5.DATA) -
      )
 /*
 //ORACCF EXEC PGM=CCF,REGION=3072K,
 //       PARM='/DSN/GDDB.SYSTEM5/'
 //STEPLIB DD DSN=MPM2.ORACLE.CMDLOAD,DISP=SHR
 //SYSOUT DD SYSOUT=*,DCB=(LRECL=132,BLKSIZE=1320,RECFM=VB)
 //SYSERR DD SYSOUT=*,DCB=(LRECL=132,BLKSIZE=1320,RECFM=VB)
 //SYSIN DD DUMMY
 //ORACATLG EXEC ORADBAV7,INDEX=MPM2,LIBV=ORACLE,REGION=6M
 //ORA@GDDB DD DUMMY      <=== ORA@SSN (ORACLE SUBSYSTEM NAME).
 //ORA$FNA  DD DSN=MPM2.ORACLE.PARMLIB(SERVFNA),DISP=SHR
//SQL      DD DISP=SHR,DSN=MPM2.ORACLE.SQL
//SYSIN    DD *
SET ECHO ON
CONNECT INTERNAL
ALTER TABLESPACE SYSTEM
ADD DATAFILE '/DSN/GDDB.SYSTEM5';
//SYSPRINT DD SYSOUT=*
//*
```

Figure 4. JCL to add new system table

The data set was created with VSAM, and we ran the Oracle CCF utility after the cluster was created. The additional data set created, GDDB.DBF.SYSTEM5 in our case, was added to the system tablespace in the Oracle7 database.

Then, we had to add the following four statements to the Migration utility JCL:

```
//SYSTEM2  DD DSN=GDDB.SYSTEM2,DISP=SHR
//SYSTEM3  DD DSN=GDDB.SYSTEM3,DISP=SHR
//SYSTEM4  DD DSN=GDDB.SYSTEM4,DISP=SHR
//SYSTEM5  DD DSN=GDDB.SYSTEM5,DISP=SHR
```

By this point, we had done the following:

- Performed all the pre-migration tasks.
- Created a new database without initializing it.
- Ran the Migration utility in check mode and as a result, added a new datafile to the Oracle7 system tablespace.

The next steps in the procedure are to start the Oracle7 database and run the Migration utility. We ran this without problems. It took approximately 30 minutes to run. We used the same JCL that we used for the space check, except that the CHECK_MODE=TRUE was removed. This must run to a condition code 0.

2.2.3 Creating the new database

There are several sets of files to build before the database instance can be created.

2.2.3.1 Creating the control files

At this point, the Migration utility has been run to create the convert file that Oracle needs to actually create the Oracle8 database. New data sets must be created for the control files in the new Oracle8 database. The control files are normally created by running the ORSIJE00 installation job when the engine is created. However, when using the Migration utility, this installation job is not run, so we have to create the control files separately.

Note: The migration chapter in the *Oracle8 for OS/390 Installation Guide* says to create two control files. This should be considered a minimum. Most DBAs will create three control files, and each control file should be on a separate physical volume for recovery reasons.

The INITORA for the new database should have the data set names for the new control files created. Figure 5 on page 13 is a partial copy of our new INITORA.

```
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
EDIT       MPMN.ORA804.PARMLIB2(OIGDB8) - 01.07            Columns 00001 00072
Command ===>
****** ***************************** Top of Data ******************************
000001 CONTROL_FILES = "/DSN/GDB8.DBF.CONTROL1"
000002 CONTROL_FILES = "/DSN/GDB8.DBF.CONTROL2"
000003 DB_FILE_MULTIBLOCK_READ_COUNT = 16
000004 DB_BLOCK_BUFFERS = 3200
000005 DB_BLOCK_SIZE = 4096
000006 DB_FILES = 256
000007 DB_NAME = GB107
000008 NLS_LANGUAGE = "AMERICAN"
000009 NLS_TERRITORY = "AMERICA"
*********************************LINES   DELETED****************************
000011 NLS_DATE_FORMAT = DD-MON-RR
000012 NLS_NUMERIC_CHARACTERS = ".,"
000013 OPTIMIZER_MODE = RULE
```

Figure 5. INITORA sample

2.2.3.2 Creating the SNAPCF file

In addition to the control files, you will also need to create a SNAPCF data set. This is not documented very well in the manuals. A new feature in Oracle8 is the Recovery Manager. In short, the Recovery Manager will keep track of the archived redo logs and catalog these data sets (within Oracle) and recall them in the proper sequence to recover the database. During recovery, Oracle will create new database files and try to recover the database in the new files it creates. If it fails, the original corrupted files still exist and other recovery methods can be tried.

In order to recover the database, Oracle creates new VSAM data sets and runs CCF against them. Oracle uses the VSAM MODEL parameter to accomplish this. To do this, it needs a data set to model. The SNAPCF data set provides that function. There are two DD statements for this:

```
//SNAPC       DD   DSN=GDB8.DBF.SNAPCF
//ORA$VSAM    DD   DSN=GBD8.DBF.SNAPCF, DISP=SHR
```

Figure 6 on page 14 shows the JCL we used to create the SNAPCF data set. We did not run CCF against this data set, since CCF will be run against the data set that is modeled from this data set.

```
EDIT       DUTCH.JCL.CNTL(SNAPF) - 01.03                 Columns 00001 00072
Command ===>                                               Scroll ===> CSR
****** ***************************** Top of Data ******************************
000001 //SNAPF1  JOB (0000,OR),'ORACLE INSTALL',CLASS=A,
000002 //           MSGCLASS=X,PRTY=15,MSGLEVEL=(1,1),NOTIFY=DUTCH
000003 //*
000004 //*
000005 //*
000006 //* LIBRARY ===> MPMN.ORA804.INSTLIB
000007 //* MEMBER  ===> ORPIJE00
000008 //* AUTHOR  ===> ORACLE CORPORATION
000021 //*--------------------------------------------------------------------*
000022 //*  JOB DESCRIPTION: DELETE AND ALLOCATE THE  ORACLE/MVS DATABASE,*
000023 //*                   CONTROL, AND REDO LOG VSAM CLUSTERS.         *
000024 //*--------------------------------------------------------------------*
000025 //*
000026 //STEPAMS1 EXEC PGM=IDCAMS
000027 //SYSPRINT DD SYSOUT=*
000028 //SYSIN    DD *
000029 //*-------------------------------------------------------*/
000030 //*                                                       */
000031 //*       DELETE THE ORACLE DB, CONTROL, AND LOG          */
000032 //*       CLUSTERS.                                       */
000033 //*                                                       */
000034 //*-------------------------------------------------------*/
000035    DELETE (GDB8.DBF.SNAPCF) -
000036           CLUSTER PURGE
000037    SET MAXCC=0    /* REMOVE AFTER FIRST TIME EXECUTION */
000038    SET LASTCC=0   /* REMOVE AFTER FIRST TIME EXECUTION */
000039 //*-------------------------------------------------------*/
000040 //*                                                       */
000041 //*       DEFINE THE DB (ORACLE DATA BASE) CLUSTERS.      */
000042 //*                                                       */
000043 //*-------------------------------------------------------*/
000044    DEFINE CLUSTER                                      -
000045      (                                                 -
000046      NAME(GDB8.DBF.SNAPCF)                             -
000047           VOLUMES(ORACL4)                              -
000048           CONTROLINTERVALSIZE(4096)                    -
000049           RECORDS(1100 500)                            -
000050           RECORDSIZE(4089 4089)                        -
000051           NONSPANNED                                   -
000052           UNIQUE                                       -
000053           NONINDEXED                                   -
000054           SHR(3 3)                                     -
000055      )                                                 -
000056      DATA                                              -
000057      (                                                 -
000058      NAME(GDB8.DBF.SNAPCF.DATA) -
000059      )
000060 //*-------------------------------------------------------*/
000061 //*                                                       */
000062 //*       LIST THE CATALOG CLUSTER ENTRY                  */
000063 //*                                                       */
000064 //*-------------------------------------------------------*/
000065    LISTC   ENT(GDB8.DBF.SNAPCF) ALL
000066 /*
```

Figure 6. Sample JCL to create SNAPCF file

2.2.3.3 Changes to the JCL to start Oracle

Next, we added the following two lines to the start proc used to start the new database:

```
//CNVFILE     DD     DSN=GDDB.DBF.CNVFILE,DISP=SHR
//MIGRATE     DD     DSN=MPMN.ORA804.SQL(MIGRATE),DISP=SHR
```

The first data set, GDDB.DBF.CNVFILE, is the data set created as a result of running the Migration utility. The second DD just points to the location of the migration script. See Figure 9 on page 17 for a sample of the JCL we used.

Next, we started the database to create the new Oracle8 instance. We used the following command to start the database:

```
START ORAGDB8.GDB8,START=MIGSTART
```

Follow the steps in the migration chapter in the *Oracle8 for OS/390 Installation Guide* to complete the migration. We did the following:

1. Started the database to execute the migration. After this step, you will no longer be able to run the Oracle kernel against the database.
2. Brought the database down gracefully from svrmgrl.
3. Removed the DD statements in the start proc needed for the migration.
4. Did a normal start of the database and ran the final jobs to complete the installation, including the INSTLIB jobs:
   - ORSIJI00 - Help files
   - ORSIJH00 - Data dictionary
   - ORSIJJ00 - SCOTT/TIGER and his demo tables.

2.2.3.4 Final steps

At this point, the Oracle7 database has been completely migrated. As part of the migration, all the views, procedures, packages, and so forth have been invalidated. We recompiled all the invalid objects (over 9000) and were left with about 135 objects that would not compile.These objects seem to have dependencies on Oracle7 or the newer Release 11.0 applications.

For example, this is the script we use to correct the invalid procedures that were identified using the query shown in Figure 1 on page 6:

```
SPOOL altprscr.sql

SELECT 'ALTER PROCEDURE '||owner||'.'||object_name||' COMPILE;'
FROM dba_objects
WHERE status='INVALID' and object_type='PROCEDURE';

SPOOL OFF
```

Figure 7. Script to correct invalid objects

The output of this script is shown in Figure 8 on page 16.

```
ALTER PROCEDURE APPS_APPDEMO.BOM_PR_GET_PROFILE COMPILE;
ALTER PROCEDURE APPS_APPDEMO.CATALOG_ITEMS COMPILE;
ALTER PROCEDURE APPS_APPDEMO.CSTPALAR COMPILE;
ALTER PROCEDURE APPS_APPDEMO.CSTPAPOR COMPILE;
ALTER PROCEDURE APPS_APPDEMO.CSTPAWAC COMPILE;
ALTER PROCEDURE APPS_APPDEMO.CSTPOPOA COMPILE;
ALTER PROCEDURE APPS_APPDEMO.CSTPSHRK COMPILE;
ALTER PROCEDURE APPS_APPDEMO.CSTPSMOH COMPILE;
ALTER PROCEDURE APPS_APPDEMO.CSTPSMTL COMPILE;
ALTER PROCEDURE APPS_APPDEMO.CSTPSPSR COMPILE;
ALTER PROCEDURE APPS_APPDEMO.CSTPULMC COMPILE;
ALTER PROCEDURE APPS_APPDEMO.MRP_AUTO_REDUCE_MPS COMPILE;
ALTER PROCEDURE APPS_APPDEMO.MRP_CALCULATE_INVENTORY_TURNS COMPILE;
ALTER PROCEDURE APPS_APPDEMO.MRP_CHECK_ORDER_MODIFIERS COMPILE;
ALTER PROCEDURE APPS_APPDEMO.MRP_MAINTAIN_REP_PERIODS COMPILE;
ALTER PROCEDURE APPS_APPDEMO.MRP_PLANNER_WORKBENCH_LOAD COMPILE;
ALTER PROCEDURE APPS_APPDEMO.MRP_RELEASE_PLAN COMPILE;
ALTER PROCEDURE APPS_APPDEMO.MRP_UPDATE_MRP_COLS COMPILE;
ALTER PROCEDURE APPS_APPDEMO.MRP_USER_DEFINED_SNAPSHOT_TASK COMPILE;
ALTER PROCEDURE APPS_APPDEMO.MRP_VALIDATE_FIELD_RANGE COMPILE;
ALTER PROCEDURE APPS_APPDEMO.MRP_WEEKS_MONTHS COMPILE;

'ALTERPROCEDURE'||OWNER||'.'||OBJECT_NAME||'COMPILE;'
--------------------------------------------------------------------------------
ALTER PROCEDURE APPS_APPDEMO.WIP_DISCRETE_JOBS_PROC COMPILE;
ALTER PROCEDURE OWA.KK COMPILE;
ALTER PROCEDURE OWA.TEST_OWA COMPILE;
ALTER PROCEDURE WWW_DBA.KK COMPILE;

25 rows selected.

Input truncated to 9 characters
```

Figure 8. Output of scripts correcting invalid objects

If this was an ERP database, the next steps would encompass running adaimgr from the Support Processor to upgrade the applications. If this was not an ERP database, all the invalid objects should be investigated. As mentioned earlier in this chapter, we recommend that you determine all the invalid objects in the database prior to doing the migration.

We were able to run several queries against the existing ERP tables and against the Scott demo tables. This was only a quick verification. If this were an actual production database, more extensive tests should be run. See Figure 9 on page 17.

```
****** **************************** Top of Data ******************************
000001 //ORAGDB8  PROC SYSOUT='SYSOUT=X', JES SYSOUT CLASS.
000002 //           INDEX='MPMN',
000003 //*                           NONVSAM LIBRARY HLINDEX.
000004 //           LIBV='ORA804',    ORACLE/INST VERSION.
000005 //           SSN=GDB8,         SSN NAME.
000006 //           START=STRTGDB8    SVRMGRL INSTANCE STARTUP PARMS.
000007 //*
000008 //*-------------------------------------------------------------------*
000009 //*             ORACLE CORPORATION                                    *
000010 //*             CORPORATE PRODUCT SUPPORT                             *
000011 //*             500 ORACLE PARKWAY                                    *
000012 //*             REDWOOD SHORES, CALIFORNIA U.S.A. 94065               *
000013 //*                                                                   *
000014 //*                              ATTENTION: MVS PRODUCT SUPPORT       *
000015 //*                                         650/506-7000              *
000016 //*                                                                   *
000017 //*  ORACLE SERVER LICENSED MATERIALS - PROPERTY OF ORACLE CORP.       *
000018 //*  CONTAINS RESTRICTED MATERIALS OF ORACLE CORP.                    *
000019 //*  REFER TO COPYRIGHT INSTRUCTIONS DOCUMENT NUMBER XXXX-XXXX.       *
000020 //*  (C) COPYRIGHT ORACLE CORP. 1986                                  *
000021 //*-------------------------------------------------------------------*
000022 //*
000023 //IEFPROC  EXEC PGM=ORACLE,PARM='>/OPER/',REGION=0M,
000024 //            DYNAMNBR=40,
000025 //            TIME=1440
000026 //STEPLIB  DD DSN=&INDEX..&LIBV..V80432.AUTHLOAD,DISP=SHR
000027 //         DD DSN=&INDEX..&LIBV..V80431.AUTHLOAD,DISP=SHR
000028 //         DD DSN=&INDEX..&LIBV..AUTHLOAD,DISP=SHR
000029 //SYSOUT   DD &SYSOUT,DCB=(LRECL=133,BLKSIZE=137,RECFM=VB,BUFNO=1)
000030 //SYSERR   DD SYSOUT=*,DCB=(LRECL=133,BLKSIZE=137,RECFM=VB,BUFNO=1)
000031 //ORA$LIB  DD DSN=&INDEX..&LIBV..V80432.AUTHLOAD,DISP=SHR
000032 //         DD DSN=&INDEX..&LIBV..V80431.AUTHLOAD,DISP=SHR
000033 //         DD DSN=&INDEX..&LIBV..AUTHLOAD,DISP=SHR
000034 //SQLBSQ   DD DSN=&INDEX..&LIBV..SQL(SQLBSQ),
000035 //            DISP=SHR
000036 //SYSPRINT DD DSN=&INDEX..&LIBV..GDB8ALRT,DISP=SHR
000037 //INITORA  DD DSN=&INDEX..&LIBV..PARMLIB2(OI&SSN),
000038 //            DISP=SHR
000039 //STARTUP  DD DSN=&INDEX..&LIBV..PARMLIB2(&START),
000040 //            DISP=SHR
000041 //SYSIN    DD DSN=&INDEX..&LIBV..PARMLIB2(MPGDB8),
000042 //            DISP=SHR
000043 //         DD DSN=&INDEX..&LIBV..PARMLIB(SQLDBA),
000044 //            DISP=SHR
000045 //         DD DSN=&INDEX..&LIBV..PARMLIB2(NETGDB8),
000046 //            DISP=SHR
000047 //MPMINS   DD DSN=MPMN.ORA804.PARMLIB2(TNSGDB8),
000048 //            DISP=SHR
000049 //SNAPCF   DD DSN=GDB8.DBF.SNAPCF,
000050 //            DISP=SHR
000051 //*
000052 //* THE FOLLOWING TWO LINES ARE FOR THE DB CONVERSION FROM 7 TO 8
000053 //*CNVFILE  DD DSN=GDDB.DBF.CNVFILE,DISP=SHR
000054 //*MIGRATE  DD DSN=MPMN.ORA804.SQL(MIGRATE),DISP=SHR
000055 //*
000056 //*    ORA$VSAM used by RMAN for dynamic allocation of DB files
000057 //ORA$VSAM DD DSN=GDB8.DBF.SNAPCF,
000058 //            DISP=SHR
000059 //*
```

Figure 9. Sample Oracle start proc

Chapter 3. Planning for use of the Migration utility

This chapter covers the key points to consider in your planning process.

3.1 High-level planning

The high-level overview planning should include the following steps.

3.1.1 Review changes and new features

The changes for Oracle 8*i* are:

- AUTHLOAD must be a PDSE.
- You must use /DSN/ instead of /DD/ for VSAM.
- Archiv files can only be to disk.

Note: IBM no longer requires PDSEs on SMS-managed volumes. See WSC Flash 10007.

The new Oracle features for OS/390 include:

- Recovery Manager - including proxy copy.
- UTL_FILE and UTL_HTTP
  Note: The UTL_FILE package lets your PL/SQL programs read and write OS text files. This must be an HFS on OS/390.
- USS utilities
- SQL*Plus, PL/SQL wrap, Intelligent Agent, NLS customized character sets
- OS auditing to SMF
- DBVERIFY

3.1.2 Select a migration method

You must decide: will you use the Migration utility or the Export/Import utilities?

3.1.3 Assess the resource requirements

Do you have the appropriate hardware and staffing resources available?

3.1.4 Plan the pre-migration steps

The pre-migration steps should be:

a. Install Oracle8.
b. Install the MIG utility in AUTHLOAD.
c. Execute a minimal test against the new database.
d. Prepare the Oracle7 database for migration.
e. Shut down Oracle7 cleanly, using SHUTDOWN NORMAL or IMMEDIATE.
f. Do a standalone backup.

3.1.4.1 Pre-migration checklist

The pre-migration steps should include:

- Check that the procedural option is installed (< 7.3 only).
- Check the status of datafiles and tablespaces.
- Check the SYSTEM rollback segment are sized correctly for.
- Check that there is no USER or ROLE that is named “migrate”.

- Check that there are no uncommitted or pending in doubt transactions.
- Check free space in SYSTEM TABLESPACE.
- Check the database character set.

  The Migration utility must use the same character set

- Disable replication by running this procedure:
  `dbms_repcat.suspend_master_activity`
  - Set the initialization parameter as follows:
    `JOB_QUEUE_PROCESSES = 0`
- Refer to Oracle8 Replication documentation for interoperability considerations.

3.1.4.2 Migration steps - phase 1

a. Shut down the Oracle7 database cleanly using SHUTDOWN NORMAL or IMMEDIATE.
b. Perform a cold backup using DF/DSS, FDR, or your standard method.
c. Start up Oracle7, ensuring that no USERS are active.
d. Use the RESTRICT option.
e. Stop the SQL*Net master task.

3.1.4.3 Migration steps - phase II

After completing the phase I steps, do the following:

a. Run the Migration utility using the sample JCL in SRCLIB(MIGJCL).
b. Add DD statements for /DD/ references.
c. If the database character set is not WE8EBCDIC1047, add:

```
ORA$ENV -NLS_LANG='AMERICAN_AMERICA.WE8EBCDIC37C'
```

d. If migrating from < 7.3.3, add:

```
ORA$ENV - CONNSTR='Z:ssn'
```

e. Check the output from migrate utility.

3.1.4.4 Migration steps - phase III

After completing the phase II steps, do the following:

a. Define new control files to be larger than v7.
b. Update the Oracle initialization parameters.
c. Set COMPATIBLE=8.0.0 or higher.
d. Set CONTROL_FILES to point to new datasets.
e. Delete all obsolete parameters.
f. Change all renamed parameters.
g. Add the following to the V8 startup JCL:

```
//CNVFILE DD...
//MIGRATE DD...
```

h. Start the Oracle8 database with the migration parameters which are in
i.
j. the member MIGSTART in PARMLIB:

```
CONNECT INTERNAL
SET ECHO ON
STARTUP PFILE=/DD/INITORA NOMOUNT
ALTER DATABASE CONVERT;
ALTER DATABASE OPEN RESETLOGS;
```

3.1.5 Prepare a backout plan

A backout plan is needed in case you encounter problems with the migration process and have to restore your system to its former status.

3.1.6 Prepare a test plan

You should design the following types of tests:

- Minimal testing
- Migration testing
- Functional testing
- RDBMS, network, applications testing
- Integration testing
- Performance testing
- Volume/load stress testing

3.1.7 Plan the post-migration steps

You should plan to perform the following tasks as part of the post-migration steps:

a. Check the joblog and alert log.
b. Run the database conversion SQL scripts.
c. Complete the migration.
d. Drop the migrate user.
e. Create system catalog views and packages.
f. Update INSTLIB(ORPIJH00) to run the following:
   - 8.0.4: CAT8000 + CAT8003 (+ CATREPM)
   - 8.0.6: CAT8000 + U0800020 (+ CATREPM)
   - 8.1.5: U0703040 (+ CATREP + R0703040)
g. Recompile invalidated PL/SQL objects:
   - 8.0.4: SQL script available in MetaLink
   - 8.0.6 or 8.1.5: SQL script UTLRP
h. Perform SHUTDOWN NORMAL or IMMEDIATE.
i. Perform a cold backup.

Part 2. Migrating an Oracle database from UNIX to OS/390

This section covers the steps to move an Oracle database from UNIX to OS/390 using the Export/Import utilities.

Chapter 4. Moving an Oracle database from UNIX or NT to S/390

This chapter discusses the points to consider when planning a move of an Oracle database from UNIX or NT to OS/390. Some of the reasons customers consider migrating to OS/390 are:

- Scalability
- Reliability and availability
- Security
- Ease of systems management on S/390
- Ability to use existing in-house skills
- Consolidation of databases onto a single platform

The Oracle documentation needed is:

- *Oracle for OS/390 Users' Guide*
- *Oracle for OS/390 System Administration Guide*
- *Oracle for OS/390 Installation Guide*
- Generic Oracle Documentation CD

Note: This documentation, including the CD, must match the level of database you will be installing on OS/390.

4.1 Planning the move from UNIX or NT to OS/390

When you are changing platforms with the Oracle database, the database objects and schema are moved to the new database. This includes not only tables and table data but stored procedures, triggers, views, packages, etc. Use the Oracle Migration utility to perform a release upgrade of an Oracle database that is already installed on OS/390.

We strongly suggest that running a proof of concept be your first step, using an application with a moderate amount of data, less than 50 Mbytes. The transaction workload selected should represent the typical workload characteristics of the production database, including OLTP, batch and reports.

The following sections detail the process to migrate the database.

4.1.1 Planning phase

The planning phase is very important. During these steps all requisite needs to move the database will be determined, including all infrastructure needs (hardware, software, network) as well as the process to install the database on S/390 and move the data from UNIX to OS/390. You should go through the following steps:

- Run a proof of concept.
- Document the objectives for moving Oracle to S/390. Determine the reasons for moving the database to OS/390, and insure that the reasons are valid and a successful move can be executed.
- Identify a team of people that will be responsible for the success of the move:
  - Sponsoring executive manager - Identify a senior manager that will support the move and help resolve situations within the I/ T organization

- Database administration manager - Identify the person that will supply DBA skills in support of the project.
- Oracle and S/390 team - List the team members from the customer, IBM, and Oracle that will be responsible for performing the move and supporting the database during the proof of concept and during production.

- Analyze the applications that are being moved.

  It is very important to understand what is being moved and all of the components that make up the application. This is especially important for application servers and Web servers. Identify what can and cannot be moved to S/390. List and investigate any appropriate alternatives. Time needs to be spent on batch processing.

  If ERP, decisions must be made about how to handle the concurrent managers. If non-ERP applications, you must determine where batch updates come from and how will they be submitted to the database on S/390. Potential problems such as ASCII/EBCDIC sort sequence differences, the use of Oracle exits, and typical porting concerns for porting applications from UNIX to USS need to be identified here.

  - Type of application (ERP/CRM/SCM or RYO)
  - Middle tier
  - Client Software (Oracle Forms, Powerbuilder, VisualBasic, Oracle Reports, etc.)
  - Web-based
    - Web server
    - HTTP Server
  - Batch applications and batch windows
  - Service Level Agreements in place

- Document the characteristics of the current UNIX/NT system hosting Oracle.

  A capacity plan will need to be developed. If this is ERP, the IBM account team should use the resources available, such as Tech Line or the Competency Center, to provide a sizing based on the platform currently hosting the Oracle database.

  - Identify the number of Oracle instances considered for the total move. Are all databases being moved, including development, test, QA, etc.?
  - Do a capacity plan for the S/390 based on all instances, not just production.

- Verify the network requirements for the move to OS/390

  It is critical that the network characteristics be identified, for two reasons: primarily, this needs to be done for the production database to insure there are no latencies that would effect performance; additionally, the network bandwidth is important when moving large amounts of data from UNIX/NT to the S/390

  - TCP/IP requirements
    - IBM TCP/IP
    - OEM TCP/IP - insure the OEM TCP/IP is supported by Oracle.
  - LU6.2
  - Level of TCP/IP on OS/390
    - If using IBM TCP/IP, the customer should use HPNS for best performance.

    - Network bandwidth - develop a diagram with all the network connections. Look for areas that lack connectivity or are potential bottlenecks.

- Analyze the Oracle database to be used for the proof of concept.

  It is possible that the level of Oracle on UNIX may not be the same as the level available on OS/390. It is extremely important to identify any problems that could inhibit or delay the move because of this.

  - Release levels - will the move necessitate moving to a lower level or Oracle compared to the instance currently running on UNIX/NT?
  - Size of database in bytes - identify the size of the database to be moved for the proof of concept and for production. Moving data is a slow process. If the database is very large, additional planning may be necessary.
  - Projected growth - this is needed to correctly size datafiles.
  - Planned projects - are there any new projects underway that should be considered, or which may have an effect on this database?
  - Number of concurrent and logged on users - this is needed for capacity issues.
  - Is batch run at the same time as OLTP? This will affect the number of MIPs needed.

- Determine skill requirements.

  The Oracle database management skills are transportable to OS/390. The same approach is used to monitor and maintain the database. However, some skills will need to be built for the OS/390 operating system. Identify the people involved in supporting the database, their tasks, and what OS/390 education will be needed, as follows:

  - DBA - should understand the OS/390 platform, TSO, and possibly knowledge of JCL.
  - Application administration - there should be little impact in this area.
  - System administration - the OS/390 administrators will need to understand tasks such as installation, console commands, creating VSAM foilsman back up and recovery. However, this is highly dependent on how the customer divides these tasks in his organization.

- Determine the S/390 environment and resources. Given what has been established to this point, determine the S/390 requirements, for example:

  - LPAR
  - Security
  - Sysplex
  - Other workloads
  - S/390 model (MIPs available vs. needed)
    - Memory requirements
  - Other resources
    - DASD
    - OSA/Communications

- Develop new plans for the following processes on OS/390. When the database and application is moved to OS/390, the following processes will change. Identify new processes and the people that will be responsible for making changes to the processes.

  - Backup and recovery - Determine best structure for tablespaces to be able to perform online backups. Implement Oracle's Recovery Manager.

- Disaster Recovery - If a disaster recovery plan is in place, how is it affected? If a standby database is being used, determine how it will be configured.
- Change Control - Identify any new change control procedures needed.

- Select tools needed to support Oracle and the application on OS/390.

  All the tools currently being used by the customer to support Oracle may work in the OS/390 environment:

  - System monitor
  - Database monitor
  - Network monitor
- Printing

  If this is ERP, determine if the printing continues to be from the Concurrent Managers, or whether it will move to OS/390. If not ERP, determine the printing requirements.

- Develop a test plan.

  It is important to establish a plan and a set of objectives to insure that success can be determined and the benefits can be measurable. Develop a timeframe in which the tests will be conducted and schedule time to analyze the outputs from the tests.

  - Objectives for functional test:

    Determine who will make this decision and what will be a successful test.

  - Objectives for performance test:

    It is worth the effort to spend additional time here to make sure the proper goals are determined, including database growth, transaction rates, and batch processing.

4.1.2 Migration plan and proof of concept

1. Prepare the environment:
   - S/390
   - Network
   - DASD
2. Install Oracle on OS/390, and insure that all the features needed are selected.
3. Create an instance similar to the instance that is being moved.
4. Export/Unload data from the UNIX/NT instance.
5. Move data to S/390.
6. Import/Load data into S/390.
7. Make all appropriate changes to the client software.
8. Make all appropriate changes to the middle-tier software:
   1. Application server
   2. Web server
   3. HTTP server
9. Test
   1. Functional test
   2. Performance test

- Develop an incident tracking process. Identify the process to resolve problems that arise during the move and testing of the application.

- Test the functionality of the application. Obtain acceptance that the functions run correctly on the OS/390.

- Test the printing processes.
- Stress-test the database on the new platform. It's possible that tuning issues may arise after the move to OS/390. Hardware and memory may have been installed on the UNIX system to resolve performance issues that were related to tuning.

  The areas of most concern are as follows (listed in the order of where most problems occur): is the SQL properly constructed; are there proper indexes; is analyze being done if the optimizer is being used; initora parameters.

4.1.3 Migrate the production databases

- Ensure that the most current patch sets and fixes are applied to the database on OS/390.
- Migrate the database structure. Use the Export and Import utilities to move only the schema (ROWS=N).
- Migrate the data to Oracle on OS/390. Once the schema has been installed in the database on OS/390, you can move the contents of the database. This takes some planning, especially if this is a large database. The objective is to create as many imports as possible, create flat files from the UNIX database, and use SQL*Loader to load the database on OS/390, or use some combination of both. This can be a lengthy process and good planning is an important part of this process.
- Post-migration tasks:
  - Database performance monitoring and tuning

    Tuning the database is a continuing process. The processes developed during the test can be altered to support the database in production.

  - Database administration procedures

    Insure that sufficient skills are available to maintain the database.

  - Database backup and recovery procedures

    These processes should have been checked out in during the proof of concept. Insure that these processes continue to be followed. Lost data is an absolute catastrophe.

4.2 Project planning

If the proof of concept and subsequent move are of a significant size, a project manager should be put in place to insure that all processes are done, and accomplished according schedule. Success will only be achieved when the project is supported by management, and has a detailed plan, and the progress is tracked.

It is very important to track the success for an agreed-to period of time after the move to production. A cross-functional team should be identified to track performance and response times. Conceivably, this team could exist for an indefinite time to insure all end-user requirements are met. The group should consist of a person each from the DBA team, networking, systems, and end-user communities.

4.3 Managing performance expectations on S/390

As previously mentioned, in addition to planning for the sizing of the database on S/390, you should also consider planning for performance management on the migrated database.

Some points to consider are:

- It is advisable to use at least a G5 S/390 processors preferably a G6 or G7, especially if you are migrating from an NT or UNIX platform that has a high Mhz rating.
- Use the latest version of OS/390 (particularly TCP/IP) to provide the best network communications performance. Use Oracle 8.0 plus patches as well.
- Using RVA or ESS enables to you have your data files striped without having to manage the layout. If you are using 3390s, you should spend some time designing the data layout to avoid I/O contention. This can be accomplished by manually *striping* the files (creating several database files that are associated with a tablespace, and placing them on different physical volumes).
- Database parameters must be continually monitored and tuned as needed to maximize performance.
- Poorly designed SQL can result in full table scans or processing that gives the user poor response time. A process must be put in place to:
  - Review the use of indexes
  - Track long-running queries with the plan of optimizing the SQL

Our experience has shown that there are no platform-specific performance problems with Oracle on OS/390. The use of current technology and standard performance management polices are necessary, though, to optimize the throughput.

Chapter 5. Using the Export/Import utilities

This chapter describes the steps to export an Oracle database on AIX and then import it to an Oracle database on OS/390. This involves creating a database on OS/390. The steps to create a database on OS/390 are described in the redbook *Oracle Applications 11 for OS/390 Installation Guide*, SG24-4980 and in *Oracle8 for OS/390 Installation Guide*, A68790. We do not repeat that material here.

For this project, we moved the AIX Oracle Vision database to OS/390 using the Oracle Export and Import utilities.

The steps to perform this process are:

1. Analyze what has to be moved.
2. Export the database.
3. Prepare the Oracle database on OS/390.
4. Import the database.
5. Review the import log and handle any problems.
6. Verify that the new database is ready.

Some steps in the migration process may be executed in a variety of ways. We cover the different alternatives that you may choose from, such as:

- Running the export or import interactively or, alternatively, using a parameter file
- Running the Import utility on OS/390 or, alternatively, from the AIX platform using the Oracle TWO_TASK variable to point to the OS/390 Oracle database instance
- Using different parameter options for the Export and Import utilities

Note

Be aware that the Export/Import utilities only run in native OS/390. They do not run in OS/390 UNIX System Services.

If you are going to export to a lower level of Oracle (that is from Oracle 8.0.5 to 8.0.4), use the parameter Direct = Y or you will run into problems with the import.

5.1 Analyzing the database

We needed to analyze the existing database so that we could size the datafiles we needed to create on OS/390.

5.1.1 Querying the database

We checked that the Oracle database on AIX was running and that the ORACLE_SID and other variables were pointing to the database we wanted to export. We then checked the number of tables and packages that were in the AIX database.

5.1.1.1 Checking the number of tables in the database

We used the following SQL commands to query the database to determine the size of the source database; see Figure 10. With this information, we were able to determine the size of the VSAM files we had to create on OS/390.

```
SQL> Select TABLESPACE_NAME, FILE_NAME, BYTES from dba_data_files;

TABLESPACE_NAME FILE_NAME                                          BYTES
--------------- ------------------------------------------- ------------
TEMP            /ora805/app/oracle/product/8.0.5/dbf/temp1vd11.dbf 209,715,200
RBS             /ora805/app/oracle/product/8.0.5/dbf/rbs1vd11.dbf 209,715,200
SYSTEM          /ora805/app/oracle/product/8.0.5/dbf/sys3vd11.dbf 419,430,400
SYSTEM          /ora805/app/oracle/product/8.0.5/dbf/sys2vd11.dbf 419,430,400
SYSTEM          /ora805/app/oracle/product/8.0.5/dbf/sys1vd11.dbf 419,430,400
USER_IDX        /ora805/app/oracle/product/8.0.5/dbf/idx2vd11.dbf 524,288,000
USER_IDX        /ora805/app/oracle/product/8.0.5/dbf/idx1vd11.dbf 524,288,000
USER_DATA       /ora805/app/oracle/product/8.0.5/dbf/dat4vd11.dbf 524,288,000
USER_DATA       /ora805/app/oracle/product/8.0.5/dbf/dat3vd11.dbf 524,288,000
USER_DATA       /ora805/app/oracle/product/8.0.5/dbf/dat2vd11.dbf 524,288,000
USER_DATA       /ora805/app/oracle/product/8.0.5/dbf/dat1vd11.dbf 524,288,000
11 rows selected.
```

Figure 10. SQL to list the size of the tablespaces

5.1.1.2 Information about tablespaces

To determine the default storage clause for each tablespace, we entered the following query:

```
SQL> select tablespace_name,initial_extent,next_extent,min_extents,max_extents
  2  pct_increase
  3  from dba_tablespaces
  4 column tablespace_name format a10 ;
```

The result of the query is shown in Figure 11.

```
TABLESPACE         INITIAL_EXTENT NEXT_EXTENT MIN_EXTENTS PCT_INCREASE
---------- -------------- ----------- ----------- ------------
SYSTEM              16384       16384           1          505
TEMP              1048576     1048576           1           50
USER_DATA          262144      524288           1           50
USER_IDX           262144      524288           1           50
RBS               1048576     1048576           1          100
```

Figure 11. SQL to show the extents used

Use the `describe` function in SQL*Plus as shown in Figure 12 on page 33 to show the columns in the dba_data_files table.

```
SQL> desc dba_data_files
 Name                           Null?    Type
 ------------------------------ -------- ----
 FILE_NAME                               VARCHAR2(257)
 FILE_ID                        NOT NULL NUMBER
 TABLESPACE_NAME                NOT NULL VARCHAR2(30)
 BYTES                                   NUMBER
 BLOCKS                         NOT NULL NUMBER
 STATUS                                  VARCHAR2(9)
 RELATIVE_FNO                            NUMBER
 AUTOEXTENSIBLE                          VARCHAR2(3)
 MAXBYTES                                NUMBER
 MAXBLOCKS                               NUMBER
 INCREMENT_BY                            NUMBER
```

Figure 12. SQL to list the fields in dba_data_files

The query shown in Figure 13 on page 34 describes the fields in all_all_tables.

```
SQL>  desc all_all_tables
 Name                            Null?    Type
 ------------------------------- -------- ----
 OWNER                                    VARCHAR2(30)
 TABLE_NAME                               VARCHAR2(30)
 TABLESPACE_NAME                          VARCHAR2(30)
 CLUSTER_NAME                             VARCHAR2(30)
 IOT_NAME                                 VARCHAR2(30)
 PCT_FREE                                 NUMBER
 PCT_USED                                 NUMBER
 INI_TRANS                                NUMBER
 MAX_TRANS                                NUMBER
 INITIAL_EXTENT                           NUMBER
 NEXT_EXTENT                              NUMBER
 MIN_EXTENTS                              NUMBER
 MAX_EXTENTS                              NUMBER
 PCT_INCREASE                             NUMBER
 FREELISTS                                NUMBER
 FREELIST_GROUPS                          NUMBER
 LOGGING                                  VARCHAR2(3)
 BACKED_UP                                VARCHAR2(1)
 NUM_ROWS                                 NUMBER
 BLOCKS                                   NUMBER
 EMPTY_BLOCKS                             NUMBER
 AVG_SPACE                                NUMBER
 CHAIN_CNT                                NUMBER
 AVG_ROW_LEN                              NUMBER
 AVG_SPACE_FREELIST_BLOCKS                NUMBER
 NUM_FREELIST_BLOCKS                      NUMBER
 DEGREE                                   VARCHAR2(10)
 INSTANCES                                VARCHAR2(10)
 CACHE                                    VARCHAR2(5)
 TABLE_LOCK                               VARCHAR2(8)
 SAMPLE_SIZE                              NUMBER
 LAST_ANALYZED                            DATE
 PARTITIONED                              VARCHAR2(3)
 IOT_TYPE                                 VARCHAR2(12)
 TABLE_TYPE_OWNER                         VARCHAR2(30)
 TABLE_TYPE                               VARCHAR2(30)
 TEMPORARY                                VARCHAR2(1)
 NESTED                                   VARCHAR2(3)
 BUFFER_POOL                              VARCHAR2(7)
```

Figure 13. SQL to list the names of fields in all_all_tables

The query shown in Figure 14 on page 35 describes the dba_rollback_segs table.

```
SQL>  describe dba_rollback_segs
 Name                           Null?    Type
 ------------------------------ -------- ----
 SEGMENT_NAME                   NOT NULL VARCHAR2(30)
 OWNER                                   VARCHAR2(6)
 TABLESPACE_NAME                NOT NULL VARCHAR2(30)
 SEGMENT_ID                     NOT NULL NUMBER
 FILE_ID                        NOT NULL NUMBER
 BLOCK_ID                       NOT NULL NUMBER
 INITIAL_EXTENT                          NUMBER
 NEXT_EXTENT                             NUMBER
 MIN_EXTENTS                    NOT NULL NUMBER
 MAX_EXTENTS                    NOT NULL NUMBER
 PCT_INCREASE                   NOT NULL NUMBER
 STATUS                                  VARCHAR2(16)
 INSTANCE_NUM                            VARCHAR2(40)
 RELATIVE_FNO                   NOT NULL NUMBER
```

Figure 14. SQL to list the names of fields in the dba_rollback_segs

The query shown in Figure 15 lists the rollback segments and the tablespace names associated with the RBS.

```
SQL> column segment_name format a15
SQL> column tablespace_name format a15
SQL> select segment_name,tablespace_name
  2  from dba_rollback_segs;

SEGMENT_NAME    TABLESPACE_NAME
--------------- ---------------
SYSTEM          SYSTEM
RBS1            RBS
RBS2            RBS
RBS3            RBS
RBS4            RBS
```

Figure 15. SQL to list the names of the existing rollback segments

This information will be used to size the OS/390 data sets in 5.3.2, “Preparing the files on S/390” on page 38.

5.1.1.3 Checking the number of packages

Query your database to see how many packages (which are stored procedures packaged together) exist on it. It may be faster to drop the packages and then recompile them after the import.

To query your database (this will give you system-wide objects):

```
SQL>Select object_type and object_name from dba_objects, where
object_type='PACKAGE'.
```

Or use user_object for your schema only.

The Oracle ERP application has over 9000 packages.

5.2 Exporting the data from the AIX Oracle database

The Oracle Export utility of Oracle was used to export the Vision database. The export could be done interactively, or by issuing a command with a parameter file. We used a parameter file first. Running the export interactively is described in 5.7, “Alternative approaches” on page 45.

Using a parameter file
We set up the following parameter file (see Figure 16):

/home/oracle/scripts/expvidb.par

```
USERID=system/manager
BUFFER=40960
FILE=/ora805/expvidb.dmp
COMPRESS=n
GRANTS=y
INDEXES=y
FULL=y
ROWS=y
CONSTRAINTS=y
RECORDLENGTH=4096
STATISTICS=none
DIRECT=y
LOG=/ora805/expvidb.log.
```

Figure 16. Export parameter file

Note

The first time we ran the Export utility, we used `DIRECT=y`. The DIRECT=y option does not export tables that have column types of REF, LOB, BFILE, or OBJECT.

The message at the end of the export log did not indicate there were any errors, but in examining the file, we found the following message, so we reran the export using `DIRECT=n` as follows:

```
EXP-00067- cannot export tables which contain object or LOB data.
```

5.3 Running the Export utility

To run the Export utility, we issued the following command:

```
exp parfile=/home/oracle/script/expvidb.par
```

The time needed for exporting will vary depending upon the size of your database and the hardware. In our case, the export we executed ran for one hour and the resulting file was 732 MB.

In order to export the database while it is active, you must use the export parameter `CONSISTENT=y`. This places a much higher overhead on the rollback segments for the rdbms. This provides data integrity in an active database by ensuring a read-consistent view.

The output log shown in Figure 17 was 247 KB long.

```
Connected to: Oracle8 Enterprise Edition Release 8.0.5.1.0 - Production
PL/SQL Release 8.0.5.1.0 - Production
Export done in WE8ISO8859P1 character set and WE8ISO8859P1 NCHAR character set


About to export the entire database ...
. exporting tablespace definitions
. exporting profiles
. exporting user definitions
. exporting roles
. exporting resource costs
. exporting rollback segment definitions
. exporting database links
. exporting sequence numbers
. exporting directory aliases
. exporting foreign function library names
. exporting object type definitions
. exporting cluster definitions
. about to export SYSTEM's tables via Direct Path ...
. . exporting table                  BROWSER_DOCS_          0 rows exported
. . exporting table              BROWSER_DOCS_SQL_          0 rows exported
. . exporting table                BROWSER_GRANTS_          0 rows exported.



..............many lines not shown



. . exporting table               EUL$VBH_IT_LINKS          0 rows exported
. . exporting table                   EUL$VERSIONS          1 rows exported
. about to export PROJMFG's tables via Direct Path ...
. about to export REPADMIN's tables via Direct Path ...
. exporting referential integrity constraints
. exporting posttables actions
. exporting synonyms
. exporting views
. exporting stored procedures
. exporting triggers
. exporting snapshots
. exporting snapshot logs
. exporting job queues
. exporting refresh groups and children
. exporting user history table
. exporting default and system auditing options
Export terminated successfully without warnings.
```

Figure 17. Export log

5.3.1 An alternative method of running an export

The export command may be run interactively without preparing the export parameter file. The sequence of commands is shown in Figure 18 on page 38.

```
exp
Export:  Release 8.0.5.1.0 - Production on Fri Jul 16: 10:50:42 1999
(c) Copyright 1998 Oracle Corporation.
All rights reserved.
Username: system
Password:
Connected to: Oracle8 Enterprise Edition Release 8.0.5.1.0 - Production PL/SQL
Release 8.0.5.1.0 - Production
Enter array fetch buffer size: 4096 >30000

Export file: ./expvidb.dmp >
 (1) E(ntire), (2)U(sers), or (3)T(ables):
  (2)U > E
Export grants (yes/no): yes > y
Export table data (yes/no): yes > y
Compress extents (yes/no): yes > n

About to export......
  exporting snapshots
  exporting snapshot logs
  exporting database links    etc.


Many lines not shown here
```

Figure 18. Running the Export utility interactively

5.3.2 Preparing the files on S/390

In order to import to the S/390, you must set up the S/390 Oracle instance to receive the data from the export, as follows:

1. Create the Oracle database instance on OS/390. This is described in Appendix C, “Installing the Oracle8i Database on OS/390” on page 83.
2. Define the additional tablespaces that will be required.

This can be done using JCL, or you can issue the commands using SVRMGRL. The JCL will allocate the VSAM files, run the Oracle CCF utility to format the VSAM files for Oracle’s use, and then the Oracle `alter` command will add the tablespaces to the Oracle database.

5.3.2.1 Creating tablespaces

We created datafiles for:

- Additional system space
- USER_DATA
- USER_IDX
- Rollback segment
- TEMP

The first DDL adds a third datafile to the system tablespace. The second and third DDLs create the tablespaces for the data (USER_DATA) and the indexes (USER_IDX). The fourth DDL creates the rollback tablespace. The fifth DDL creates the temporary tablespace that Oracle will use to do tasks such as sorting. This TEMP area is never permanent, but you do not want Oracle to run out of temporary space, either.

The create statements we used are shown in Figure 19.

```
alter tablespace SYSTEM add '/dsn/MPMD.DBF.SYSTEM.DB3;

create tablespace USER_DATA datafile '/DSN/MPMD.DBF.UDATA' size 2000M
default storage (initial 100M next 100M minextents 2 maxextents 250 pctincrease0)
online;

create tablespace USER_IDX datafile '/DSN/MPMD.DBF.USERIDX' size 750M
default storage (initial 50M next 50M minextents 2 maxextents 250pctincrease 0)
online;

create tablespace RBS datafile '/DSN/MPMD.DBF.RBS' size 450M
default storage (initial 50M next 50M minextents 2 maxextents 250 pctincrease 0)
online;

create tablespace TEMP datafile ' /DSN/MPMD.DBF.TEMP' size 500M
default storage (initial 10M next 10M minextents 2 maxextents 250 pctincrease 0)
online;

alter tablespace SYSTEM add '/dsn/MPMD.DBF.SYSTEM.DB3
```

Figure 19. Create tablespace commands

5.3.3 Creating rollback segments

We created rollback segments R1, R2, R3, R4, and R_BIG. When we were running long jobs, such as adaimgr and exports, we altered the four small rollback segments offline and used only R_BIG.

Figure 20 on page 40 shows a sample of the JCL we used.

```
//ORATOMB JOB (0000,OR),'ORACLE INSTALL',CLASS=A,
//           MSGCLASS=X,PRTY=15,MSGLEVEL=(1,1),NOTIFY=ORACLE2
//*
//*   ALLOCATE ROLLBACKSEGMENTS
//*
//STEPAMS1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
    DELETE (MPMD.DBF.RBS1) -
           CLUSTER PURGE
    SET MAXCC=0    /* REMOVE AFTER FIRST TIME EXECUTION */
    SET LASTCC=0   /* REMOVE AFTER FIRST TIME EXECUTION */
  /*-------------------------------------------------*/
    DEFINE CLUSTER                                  -
      (NAME(MPMD.DBF.RBS1)                 -
           VOLUMES(ORACL1)                          -
           CONTROLINTERVALSIZE(4096)                -
           MEGABYTES(500 200)                        -
           RECORDSIZE(4089 4089)                    -
           NONSPANNED                               -
           UNIQUE                                   -
           NONINDEXED                               -
           SHR(3 3))                                -
      DATA                                          -
     (NAME(MPMD.DBF.RBS1.DATA))
//ORACCF EXEC PGM=CCF,REGION=3072K,
//      PARM='/DSN/MPMD.DBF.RBS1'
//STEPLIB DD DSN=MPMD.ORA804.V80432.CMDLOAD,DISP=SHR
//        DD DSN=MPMD.ORA804.V80431.CMDLOAD,DISP=SHR
//        DD DSN=MPMD.ORA804.CMDLOAD,DISP=SHR
//SYSOUT DD SYSOUT=*,DCB=(LRECL=132,BLKSIZE=1320,RECFM=VB)
//SYSERR DD SYSOUT=*,DCB=(LRECL=132,BLKSIZE=1320,RECFM=VB)
//SYSIN DD DUMMY
//ORACATLG EXEC ORADBATH,REGION=6M
//ORA@MPMD DD DUMMY      <=== ORA@SSN (ORACLE SUBSYSTEM NAME).
//ORA$FNA  DD DSN=MPMD.ORA804.PARMLIB(SERVFNA),DISP=SHR
//SQL      DD DISP=SHR,DSN=MPMD.ORA804.SQL
//SYSIN    DD *
  SET ECHO ON
CONNECT INTERNAL
  CREATE TABLESPACE RBS DATAFILE '/DSN/MPMD.DBF.RBS1/'
  CREATE ROLLBACK SEGMENT R1 TABLESPACE RBS
     STORAGE(INITIAL 5M NEXT 5M MINEXTENTS 2 MAXEXTENTS 249 OPTIMAL 20M)
  CREATE ROLLBACK SEGMENT R2 TABLESPACE RBS
     STORAGE(INITIAL 5M NEXT 5M MINEXTENTS 2 MAXEXTENTS 249 OPTIMAL 20M)
  CREATE ROLLBACK SEGMENT R3 TABLESPACE RBS
     STORAGE(INITIAL 5M NEXT 5M MINEXTENTS 2 MAXEXTENTS 249 OPTIMAL 20M)
  CREATE ROLLBACK SEGMENT R4 TABLESPACE RBS
     STORAGE(INITIAL 5M NEXT 5M MINEXTENTS 2 MAXEXTENTS 249 OPTIMAL 20M)
  CREATE ROLLBACK SEGMENT R_BIG TABLESPACE RBS
   STORAGE(INITIAL 50M NEXT 10M MINEXTENTS 2 MAXEXTENTS 249 OPTIMAL 50M)
  //SYSPRINT DD SYSOUT=*
  //*
```

Figure 20. JCL to create rollback segments

5.3.3.1 Creating large rollback segments

With Oracle8, there are several large tables (such as CTX) that need a very large rollback segment, so we created a large rollback segment:

```
create rollback segment R_BIG tablespace RBS storage (initial 20M next 20M
minextents 2 maxtents 121);
```

To look at the rollback segments, enter the following:

```
select * from dba_rollback_segs;
```

Change the initora to reflect the new big rollback segment and then stop and start the Oracle database so the new rollback statements are in effect.

In order to force Oracle to use the big rollback segments, you must take the little ones offline and put the big one online using the following commands:

```
alter rollback segment S1 offline;
alter rollback segment S2 offline;
alter rollback segment S3 offline;
alter rollback segment S4 offline;
alter rollback segment R_BIG online;
```

5.3.4 The initora parameters

To do the import, check the initora file to insure that the shared pool size and db block buffer size are adequate. We set the shared pool size to 40000000 (40 MB) and the db block buffer size to 10000000 (10 MB). The Oracle database must be cycled to grab the new initora file. This initora is good for imports, but it would not be good for OLTP. Therefore, you should have two initoras for these different types of processing.

5.4 Running the import

Once the database is prepared on OS/390, you must return to the AIX system to set up the TWO_TASK variable and the tnsnames.ora file to communicate to the Oracle database on S/390. In our case, this was MPMD.

The other way to do this is to FTP the file to the S/390 and run a job that executes the import. This is described in 5.4.3, “Running the import” on page 42. However, if you do this, you must check that the FTP program does not affect the code page. In several cases, we found we had to do the import from AIX to maintain the consistency of the data.

5.4.1 Parameter files

Next, set up the import parameter file in /home/oracle/scripts/import.par; it looks like the following:

```
USERID=system/manager
BUFFER=1341300
FILE=/ora805/expvidb.dmp
COMMIT=y
IGNORE=Y
GRANTS=y
INDEXES=y
FULL=Y
LOG=/ora805/importv.log
```

Issue the import command:

```
imp parfile=import.par
```

Note

We started with the default BUFFER parameter of 30720 but we received error messages, so we increased it to 1341300.

5.4.2 Importing from the AIX system

In order to import from the AIX side straight through to the S390, we set the TWO_TASK option. The TWO_TASK option should be set to the SID of the machine to which you wish to import.

In our case, the SID was MPMD.

The command on the AIX side was:

```
export TWO_TASK=MPMD
```

To view this, you can echo:

```
echo $TWO_TASK
```

You should see:

```
MPMD
```

You can check to be sure that you are pointing to the SID on the S390 by invoking the `SVRMGRL` command and noting if you are actually seeing the SVRMGR from the S390.

5.4.3 Running the import

To run the import, the command is:

```
imp parfile=/home/oracle/scripts/import.par
```

See the output shown in Figure 21 on page 43.

```
Connected to: Oracle8 Enterprise Edition Release 8.0.4.3.2 - Production
PL/SQL Release 8.0.4.3.0 - Production

Export file created by EXPORT:V08.00.05 via direct path
. importing APPS's objects into APPS
. importing SYSTEM's objects into SYSTEM
. importing CTXSYS's objects into CTXSYS
. importing APPS's objects into APPS
. importing CTXSYS's objects into CTXSYS
. importing APPS's objects into APPS
. importing CTXSYS's objects into CTXSYS
. importing APPS's objects into APPS
. importing CTXSYS's objects into CTXSYS
. importing APPS's objects into APPS
. importing APPLSYS's objects into APPLSYS
. importing APPS's objects into APPS
. importing APPLSYS's objects into APPLSYS
. importing APPS's objects into APPS
. importing APPLSYS's objects into APPLSYS
. importing APPS's objects into APPS
. importing APPLSYS's objects into APPLSYS.

   many lines not shown

. importing SSP's objects into SSP
. importing HXT's objects into HXT
. importing OEMGR's objects into OEMGR
. importing SCOTT's objects into SCOTT
. importing WEBSYS's objects into WEBSYS
. importing WWW_DBA's objects into WWW_DBA
. importing CLR's objects into CLR
. importing DISCO's objects into DISCO
Import terminated successfully with warnings.
```

Figure 21. Import log

5.4.4 Reviewing the import log

The first time we ran this command, we received several types of error messages, as follows:

- Warning messages about existing constraints

```
IMP-00017: following statement failed with ORACLE error 2264:
 "ALTER TABLE "REPCAT$_AUDIT_ATTRIBUTE" ADD  CONSTRAINT
"REPCAT$_AUDIT_ATTRIB"
 "UTE_C1" CHECK ((data_type_id in (2, 4, 5, 6, 7) and
data_length"
 "is not null)         or (data_type_id not in (2, 4, 5, 6, 7) and"
 "data_length is null)) ENABLE NOVALIDATE"
IMP-00003: ORACLE error 2264 encountered
ORA-02264: name already used by an existing constraint
```

  We did not consider this to be a problem.

- Warning messages that unique or primary keys already existed

```
IMP-00003: ORACLE error 2261 encountered
ORA-02261: such unique or primary key already exists in the table
. . importing table               "REPCAT$_REPCAT"          0 rows imported
```

We did not consider this to be a problem. We realized we were receiving the IMP-00017 and IMP-00003 messages because we ran the incorrect scripts to prepare the database--therefore, you should only run the catproc script.

- When we first ran with commit=n, we received messages that the rollback segment was not large enough

```
IMP-00058: ORACLE error 1562 encountered
ORA-01562: failed to extend rollback segment number 1
ORA-01628: max # extents (249) reached for rollback segment S1
IMP-00028: partial import of previous table rolled back: 55060 rows rolled
back.
```

  When we created a large rollback segment and altered the others offline in a second run, we no longer received the message.

- Warning messages about unique constraints

```
IMP-00019: row rejected due to ORACLE error 1
IMP-00003: ORACLE error 1 encountered
ORA-00001: unique constraint (SYSTEM.HELP_TOPIC_SEQ) violated
Column : ACCEPT
Column : 1
Column :
```

  We did not consider these messages to be problems.

- Warning message about compilation errors (this message was received many times)

```
IMP-00041: Warning: object created with compilation warnings
```

  We had to recompile these objects later to correct this error. This is a common occurrence in an import process.

- Warning messages about not enough memory

```
IMP-00044: unable to allocate enough memory for statement
IMP-00021: operating system error - error code (dec 12, hex 0xC)
```

  We had to test this several times. When we increased the value of the BUFFER parameter in the import parameter file to BUFFER = 1341300, the messages did not appear in the second run.

  To determine the new value, we ran the following query against the source database:

```
select max(avg_row_len) from dba_tables;
```

  We added 10% to that value and multiplied by 100 for 100 rows as described in the *Oracle Utilities Guide*.

5.5 Completing the process

You will have to recompile or recreate the objects that had errors.

5.6 Verifying that the database is ready

To complete the import, we had to do some recompiles. We first did a query to see which objects were invalid; see Figure 22 on page 45.

```
SQL> COLUMN owner_name FORMAT a20
SQL> COLUMN object_name FORMAT a35

SQL> SPOOL INVLD_OBJS.lst

SQL> SELECT   object_name, owner, object_type
   1 FROM     dba_objects
   2 WHERE    status='INVALID'
   3 ORDER BY object_type;

OBJECT_NAME                         OWNER                          OBJECT_TYPE
----------------------------------- ------------------------------ -------
PO_HEADERS_SV4                      APPS_MRC                        PACKAGE BODY
PO_HEADERS_SV2                      APPS_MRC                        PACKAGE BODY
PO_HEADERS_PKG_S3                   APPS_MRC                        PACKAGE BODY
PO_HEADERS_PKG_S2                   APPS_MRC                        PACKAGE BODY
PO_HEADERS_PKG_S1                   APPS_MRC                        PACKAGE BODY
PO_HEADERS_PKG_S0                   APPS_MRC                        PACKAGE BODY
PO_CORE_SV1 APPS_ MRC PACKAGE BODY
PO_COPY_DOCUMENTS_S                 APPS_MRC                        PACKAGE BODY
PO_CHORD_WF1                        APPS_MRC                       PACKAGE BODY
PO_BUYER_WORKLOAD_SV                APPS_MRC                        PACKAGE BODY
PO_AUTOSOURCE_SV                    APPS_MRC                        PACKAGE BODY
PO_AUTOCREATE_DOC                   APPS_MRC                        PACKAGE BODY
PO_ASL_UPGRADE_SV2                  APPS_MRC                        PACKAGE BODY
MRP_RELEASE_PLAN_SC                 APPS                            PROCEDURE
MRP_RELEASE_PLAN_SC_AP              APPS                            PROCEDURE
MRP_RELEASE_PLAN_SC                 APPS_MRC                        PROCEDURE
CLR$ENF_CHK_ING_SUPP                APPS_MRC                        PROCEDURE
MRP_RELEASE_PLAN_SC_AP              APPS_MRC                        PROCEDURE
ADS_OTM_PERIOD_OTHOURS_RSP_V        APPS                            VIEW
AP_AWT_TEMP_DISTRIBUTIONS_V         APPS                            VIEW
AP_BATCHES_V                        APPS                           VIEW
```

Figure 22. List of invalid objects

Then, we did a query to produce the input statements for the compiles.

5.7 Alternative approaches

Previously, we had run the import from the AIX system using the TWO_TASK parameter. This section describes running the import job using JCL on the S/390.

5.7.1 Running from the S/390 system

The next method was to try to FTP the export file to the S390. You should FTP the file in binary. The command sequence would look like the following:

```
ftp 9.12.14.219
```

You will be asked for your name and password on the S390 system. Enter the command to indicate you want to send the file in binary. The command is:

```
bin
```

Next, you need to issue the site command to allocate the space for your large data set that will contain the export dump. The command is:

```
quote site cylinders primary=2000 secondary=500 volume=oracl4
```

ORACL4 is the pack that has enough space to place this large data set. FTP will respond with the message `SITE command accepted`.

The other way to do this is to preallocate a large sequential file to receive the data that you will send via the FTP command.

The `put` command would look like:

```
put expvidb.dmp 'mpmd.exportvi.dmp'
```

This can be any name, since you used the `site` command.

When the put finishes, you can then try to import on the S/390 to the S/390 Oracle database.

5.7.2 Handling import errors and rerunning the import on S/390

If you have errors in the import process, you may need to clean up the database before you restart. There are several ways to do this:

1. Start over from the very beginning.

   You can drop all the tablespaces and rerun the jobs that delete all the VSAM clusters, reallocate them, and run the CCF utility to format them for Oracle, then run the create tablespace commands. Now, you have a completely clean database and can rerun the import. The create table commands in the export file will be executed by the import to create the tables again.

   See the job ORPIJE00 in Appendix B, “Running the Export/Import utilities interactively” on page 79 for a sample of this JCL.

2. Use the `truncate` command.

   This is described in Step 3.

   Note: The `truncate` command deletes all data in the table and does not create rollback information. Therefore, this data cannot be recovered. If this is clustered table, it must have indexes.

3. Redo just a few tables.

   In this case, the IGNORE=Y parameter is used and the import process bypasses the tables it cannot handle. It is not necessary to reimport all the tables—only the ones that failed.

   The import of a table may have failed because of space constraints; you will have to allocate sufficient space for large tables. See Chapter 9, “Another example using the Export/Import utilities” on page 65 which describes our experience with this problem.

   At this point, you should use the `truncate` option to clean out the rows that have already been loaded. The `truncate` command keeps the table intact but removes the data. The option of using the delete rows is not recommended, since it may also delete any rows that are referenced by this table (a cascading effect), and you do not want the rows in good tables to be deleted.

   The other option is to drop the table, then export only that table. On the import, a create of the table will be done, since it no longer exists. If this table has relationships with other tables, you must drop the tables in a specific sequence based on the relationship.

5.7.3 Using the truncate command to remove data from one table

The `truncate` command to clear out one table is:

```
truncate table APPLSYS.AP_FALLS_FROM_TREE;
```

Then run the SQL through a JCL file like the one shown in Figure 23.

```
//DUTCH1  JOB (0000,OR),'ORACLE INSTALL',CLASS=A,
//           MSGCLASS=X,PRTY=15,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//ORACATLG EXEC ORADBADD,REGION=6M
//ORA@MPMD DD DUMMY      <=== ORA@SSN (ORACLE SUBSYSTEM NAME).
//ORA$FNA  DD DSN=MPMD.ORA804.PARMLIB(SERVFNA),DISP=SHR
//SQL      DD DISP=SHR,DSN=MPMD.ORA804.SQL
//SYSIN    DD DISP=SHR,DSN=TRUNCATE.OUT
//SYSPRINT DD SYSOUT=A
//SYSIN    DD  *
SET ECHO ON
CONNECT INTERNAL
truncate table APPLSYS.AP_FALLS_FROM_TREE
/*
```

Figure 23. JCL to run truncate commands with the data as SYSIN

5.7.4 Using the truncate command to remove data from all tables

If the import failed previously, you may need to clear out the tables. We did this by using SQL to generate the names of the tables and then executing a truncate table command on each of the tables. Use SQL*Plus and connect to the database, start spool sending output to a data set name, select the owner and names of the tables from the dba_tables, and generate SQL onto the spool file.

The commands are:

```
spool '/DSN/ORACLE1.MONAT.SQL.DROP'
select 'truncate table '||owner||'.'||table_name||';'
from dba_tables
where tablespace_name='USER_DATA';
```

This will generate a listing which is displayed on your screen and is also put into the spool file ORACLE1.MONAT.SQL.DROP. It contains data like the following:

```
truncate table APPLSYS.AP_FALLS_FROM_TREE;
```

You will have to edit this file. At the beginning, delete any garbage lines and put in the following two lines:

```
set echo on
connect system/password
```

You then run the SQL through a JCL file. See Figure 24 on page 48.

```
//INST09  JOB (0000,OR),'ORACLE INSTALL',CLASS=A,
//           MSGCLASS=X,PRTY=15,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//*
//ORACATLG EXEC ORADBA08,REGION=6M
//*
//ORA@MPMD DD DUMMY      <=== ORA@SSN (ORACLE SUBSYSTEM NAME).
//ORA$FNA  DD DSN=MPMD.ORA804.PARMLIB(SERVFNA),DISP=SHR
//SQL      DD DISP=SHR,DSN=MPMD.ORA804.SQL
//SYSIN    DD  DSN=ORACLE1.MONAT.SQL.DROP,DISP=SHR
//SYSPRINT DD SYSOUT=*
//*
```

Figure 24. JCL to run the truncate commands from a file

5.7.5 Running the import command on S/390

To do the import, set up JCL to run the import, as follows:

The parm file for this import follows:

```
USERID=SYSTEM/MANAGER
BUFFER=13413000
FILE=/DSN/MPMD.EXPORTVI
GRANTS=Y
COMMIT=Y
IGNORE=Y
INDEXES=Y
FULL=Y
LOG=/DSN/MPMD.IMPORT.LOG
```

This is where we had to change the BUFFER value to eliminate the memory messages.

The JCL to run the import is shown in Figure 25.

```
//IMPORTVI JOB (0000,OR),'IMPORT V8',CLASS=A,
//            MSGCLASS=X,MSGLEVEL=(1,1),PRTY=15,NOTIFY=&SYSUID
//IMPORT            EXEC  PGM=IMP,PARM='SYSTEM/MANAGER PARFILE=/DD/PARFILE'
//STEPLIB             DD  DSN=MPMD.ORA804.CMDLOAD,DISP=SHR
//SYSOUT              DD  SYSOUT=*,DCB=(LRECL=132,BLKSIZE=1320,RECFM=VB)
//SYSERR              DD  SYSOUT=*
//ORA@MPMD            DD  DUMMY
//SYSIN               DD  DUMMY
//PARFILE             DD  DSN=MPMD.IMPORT.PARFILE,DISP=SHR
//IMPFILE             DD  DSN=MPMD.EXPORTVI,DISP=SHR
//LOGFILE             DD  DSN=MPMD.IMPORT.LOG,DISP=SHR
```

Figure 25. JCL for import job

Note: For security purposes, you may want to put the user ID and password in the parfile, rather than exposing it in the JCL as we have done in Figure 25.

The import was successful, although the log did contain messages about constraints and existing objects.

5.7.5.1 Another problem encountered

In one case, we found that we had problems with the file that had been FTPed to the S/390.

We received the following message:

```
Connected to: Oracle8 Enterprise Edition Release 8.0.4.3.0 - Production
PL/SQL Release 8.0.4.3.0 - Production

IMP-00038: Could not convert to environment character set's handle
IMP-00000: Import terminated unsuccessfully
```

We had to do the import from an AIX platform with the TWO_TASK variable set to the database on S/390.

5.7.6 Exporting the DDL separately

Another alternative is to export the DDL first, create the schema on the target database, and then export and import only the data later.

5.7.6.1 Parameter file

The parameter file is called:

/home/oracle/scripts/expddl.par

The parm file to export only the DDL contained the following:

```
USERID=system/manager
BUFFER=30720
FILE=/ora805/expddl.dmp
COMPRESS=n
GRANTS=y
INDEXES=y
FULL=y
ROWS=n
CONSTRAINTS=y
LOG=/ora805/expddl.log
```

5.7.6.2 Exporting DDL only

Use the following command when you want to export just the DDL:

```
exp userid=system/manager parfile=/home/oracle/scripts/expddl.par
```

The file containing the DDL was /expddl.dmp. This file contained many DDL statements; the first few lines are shown in Figure 26 on page 49.

```
RENTIRE
4096
0
530
4000
#C#A°#BEGINSYS
CREATE TABLESPACE "TEMP" DATAFILE
'/ora805/app/oracle/product/8.0.5/dbs/temp1vd11.dbf' SIZE 209715200 REUSE
DEFAULT STORAGE (INITIAL 1048576 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 50
PCTINCREASE 50) ONLINE PERMANENTCREATE TABLESPACE "USER_DATA" DATAFILE
'/ora805/app/oracle/product/8.0.5/dbs/dat4vd11.dbf' SIZE 524288000 REUSE,
'/ora805/app/oracle/product/8.0.5/dbs/dat3vd11.dbf' SIZE 524288000 REUSE,
'/ora805/app/oracle/product/8.0.5/dbs/dat2vd11.dbf' SIZE 524288000 REUSE,
'/ora805/app/oracle/product/8.0.5/dbs/dat1vd11.dbf' SIZE 524288000 REUSE
DEFAULT STORAGE (INITIAL 262144 NEXT 524288 MINEXTENTS 1 MAXEXTENTS 50
PCTINCREASE 50) ONLINE PERMANENT
```

Figure 26. Example of the DDL output

The log of the export of the DDL was /home/oracle/expddl.log. It contained 168058 characters. The first few lines are shown in Figure 27.

```
Connected to: Oracle8 Enterprise Edition Release 8.0.5.1.0 - Production
PL/SQL Release 8.0.5.1.0 - Production
Export done in WE8ISO8859P1 character set and WE8ISO8859P1 NCHAR character set
Note: table data (rows) will not be exported

About to export the entire database ...
. exporting tablespace definitions
. exporting profiles
. exporting user definitions
. exporting roles
. exporting resource costs
. exporting rollback segment definitions
. exporting database links
. exporting sequence numbers
. exporting directory aliases
. exporting foreign function library names
. exporting object type definitions
. exporting cluster definitions
. about to export SYSTEM's tables via Conventional Path ...
. . exporting table                  BROWSER_DOCS_

many lines not shown

. exporting default and system auditing options
Export terminated successfully without warnings.
~
```

Figure 27. Log for export of DDL

These DDL statements can now be run on the Oracle OS/390 database to create the tablespaces, and then you can export and import only the data.

5.8 The Export and Import parameters

The Export and Import parameters you use can affect the timing of your export and import. See Chapter 6, “Choosing Export/Import parameters” on page 51 for more information on why we chose the options we did.

Chapter 6. Choosing Export/Import parameters

This chapter discusses the points you must consider when preparing to do export/imports to S/390 if you are concerned about the length of time this process will take:

- Your network setup must ensure that your FTP process will be not hindered by delays in the network (sometimes large files get lower priority). If you are using the TWO_TASK variable to run the import from AIX, network delays may increase the time to run the import.
- Be sure to allocate sufficient interim disk space for the export file and for the log files. We used the following parameters for export:

```
BUFFER=40960
CONSISTENT=n
GRANTS= y
HELP=n
INDEXES=y
RECORDLENGTH=4096
ROWS=y
STATISTICS=none
```

- If you use statistics, the Oracle manual says it slows it down considerably on import. We used the following parameters for import:

```
BUFFER=40960
COMMIT=n
DESTROY=n
GRANTS=y
HELP=n
IGNORE=y
INDEXES=y
RECORDLENGTH=4096
```

Some other points to keep in mind are:

- Consider the ASCII-to-EBCIDIC character conversions that are required, especially in NLS situations. The export file is part binary and part ASCII (text), so FTP in binary can be a problem. To avoid this, in some cases, we did the import with the TWO_TASK set.
- Will you need one very large rollback segment?
- Do you want to defer building your indexes?
- If you are using packages or stored procedures, do you have the patch for Oracle7 Release 7.3 to improve the import time? This is included in Oracle8.
  - Commit = yes means to commit the imported rows after importing a buffer full. The issues are time versus rollback segments. If you are importing a large table, you will need large rollback segments; otherwise, the import will abend. If the rollback segments are large enough, it is faster to do one commit at the end of the table.

The following lists the export parameters that available with Oracle. The parameter is named, followed by a brief description and the default value shown in parentheses. For a complete explanation, see the *Oracle Utilities Manual*, A58244.

Note that the USERID parameter must be the first parameter on the command line.

USERID	Username/password
FULL	Export entire file (n)
BUFFER	Size of data buffer
OWNER	List of owner usernames
FILE	Output file (EXPDAT.DMP)
TABLES	List of table names
COMPRESS	Import into one extent (y)
RECORDLENGTH	Length of IO record
GRANTS	Export grants (y)
INCTYPE	Incremental export type
INDEXES	Export indexes (y)
RECORD	Track incr. export (y)
ROWS	Export data rows (y)
PARFILE	Parameter file name
CONSTRAINTS	Export constraints (y)
CONSISTENT	Cross-table consistency
LOG	Log file of screen output
STATISTICS	Analyze objects (ESTIMATE)
DIRECT	Direct path (n)
FEEDBACK	Display progress every x rows (0)
POINT_IN_TIME_RECOVER	Tablespace point-in-time recovery (n)
RECOVERY_TABLESPACES	List of tablespace names to recover
VOLSIZE	Number of bytes to write to each tape volume

The import parameters are:

USERID	Username/password
FULL	Import entire file (n)
BUFFER	Size of data buffer
FROMUSER	List of owner usernames
FILE	Input file (EXPDAT.DMP)
TOUSER	List of usernames
SHOW	Just list file contents (n)
TABLES	List of table names
IGNORE	Ignore create errors (n)
RECORDLENGTH	Length of IO record
GRANTS	Import grants (y)
INCTYPE	Incremental import type
INDEXES	Import indexes (y)
COMMIT	Commit array insert (n)
ROWS	Import data rows (y)
PARFILE	Parameter file name
LOG	Log file of screen output
DESTROY	Overwrite tablespace datafile (n)
INDEXFILE	Write table/index info to specified file
CHARSET	Character set of export file (NLS_LANG)
POINT_IN_TIME_RECOVER	Tablespace point-in-time recovery (n)

SKIP_UNUSABLE_INDEXES	Skip maintenance of unusable indexes (n)
ANALYZE	Execute ANALYZE statements in dump file (y)
FEEDBACK	Display progress every x rows (0)
VOLSIZE	Number of bytes in file on each volume of a file on tape

The following describes some of the Oracle8 Export parameters in more detail.

The export parameter file allows you to specify Export parameters in a file where they can be easily modified or reused. Create the file using any flat file text editor. The command line option `PARFILE=file` tells Export to read the parameters from the specified file, rather than from the command line. The syntax for the parameter file specification is one of the following:

KEYWORD=value

KEYWORD=(value)

KEYWORD=(value1, value2, ...)

You can add comments to the parameter file by preceding them with the pound (#) sign. Export ignores all characters to the right of the pound (#) sign.

You needs may vary depending on the specific task you are doing. Here are some common parameters that can be used in the parameter file:

BUFFER
The default is operating system-dependent. BUFFER specifies the size, in bytes, of the buffer used to fetch rows. As a result, this particular parameter determines the maximum number of rows in an array fetched by Export.

Use the following formula to calculate the buffer size:

buffer_size=rows_in_array + maximum_row size

If you specify zero, the Export utility fetches only one row at a time. Tables with LONG, LOB, BFILE, REF, ROWID, or type columns are fetched one row at a time.

COMPRESS
The default is y (yes). This parameter specifies how export and import manage the initial extent for table data. The default value, COMPRESS=y, causes export to flag table data for consolidation into one initial extent upon Import.

This option can have positive effects and it can have negative effects. For example, if you are reorganizing tables to minimize the number of extents, the option is useful, except for the fact that you must have a contiguous block of space for the initial extent.

This particular parameter needs to be analyzed before you use it, since the import may not find enough space for the initial extent.

If you specify COMPRESS=n, Export uses the current storage parameters, including the values of initial extent size and next extent size. The values may be

the ones specified in the CREATE TABLE or ALTER TABLE statements, or the values modified by the database system

Note

You can specify the COMPRESS option only on export, not when you import. The Export utility generates the data definitions language (DDL), not the Import utility.

CONSISTENT
The default is n (no). This parameter specifies whether or not Export uses the SET TRANSACTION----------S READ ONLY statement to insure that the data seen by Export is consistent to a single point in time and does not change during the execution of the export command. You should use CONSISTENT=y when you anticipate that other applications will be updating the database after an export has started.

If you specify CONSISTENT=n, tables are usually exported in a single transaction. If a table contains nested tables, the outer table and each inner table are exported as separate transactions. If a table is partitioned, each partition is exported as a separate transaction.

CONSTRAINTS
The default is y (yes). This parameter specifies whether or not the Export utility exports table constraints.

DIRECT
The default is n (no). This parameter specifies whether or not to use direct path or conventional path Export. Specifying DIRECT=y causes Export to extract data by reading the data directly, bypassing the SQL Command Processing layer (evaluating buffer).

This method can be much faster than a conventional path export. You can further improve performance by using direct path export with the database in direct read mode. Contention for resources with other users is eliminated, because database blocks are read into the private buffer cache, rather than a public buffer cache.

RECORDLENGTH
The maximum block size for VSAM files for OS/390 is 4096. The record length must be less than or equal to this. It should be equal to the widest row in the database.

STATISTICS
This specifies which type of database optimizer statistics are to be generated during the export. The default is ESTIMATE.

Chapter 7. Using the OS/390 Oracle Export/Import utilities

This chapter describes the steps to export a database on OS/390 (MPMT) and then to import it to another database instance on OS/390 (MPMN).

The steps to do a full database import into MPMN are:

1. Run the export job on the MPMT instance.
2. Prepare a new Oracle target instance (MPMN).
   To do this you can rerun the tapes to create the new database, or use the customization panels to create a secondary instance.
3. Create the database files and tablespaces for the target instance MPMN. This is described in “Information about tablespaces” on page 32.
4. Start up MPMN with the create option.
5. Run the import job.
6. Check the logs for any errors.

7.1 Running the export job on S/390

We ran an export job that created a file ORACLE1.MPMT.FULL.EXPORT that was a full export of the database on the Oracle instance MPMT.

The JCL we used to run the export using ORACLE1.JCL.CNTL(ORAXPORT) CURRENTLY POINTS TO MPMT is shown in Figure 28.

```
//ORAEXPF  JOB (999,POK),'ORACLE EXPORT',NOTIFY=ORACLE1,
//         TIME=1439,CLASS=A,MSGCLASS=X
//*
//* JCL TO RUN EXPORT FROM BATCH ENVIRONMENT
//*
//STEP1    EXEC ORAEXPV8,INDEX=MPMT,LIBV=ORA8043,
//      PARM='SYS/CHANGE_ON_INSTALL PARFILE=/DD/PARFILE'
//ORA@MPMT DD DUMMY
//SYSIN    DD DUMMY
//PARFILE  DD DSN=ORACLE1.XPORT.PARMS,
//            DISP=SHR
//EXPFILE  DD DSN=ORACLE2.MPMT.FULL.EXPORT,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=3490,
//            DCB=(RECFM=FB,LRECL=4096,BLKSIZE=24576)
//LOGFILE  DD DSN=ORACLE1.XPORT.LOG,DISP=SHR
/*
```

Figure 28. JCL used for the export utility

The following is the parameter file (ORACLE1.XPORT.PARMS) that we used:

```
FILE=/DD/EXPFILE
LOG=/DD/LOGFILE
DIRECT=N
BUFFER=41932
FULL=Y
```

Figure 29 on page 56 shows the log file (ORACLE1.XPORT.LOG) that was created.

```
Connected to: Oracle8 Enterprise Edition Release 8.0.4.3.2 - Production
PL/SQL Release 8.0.4.3.0 - Production
Export done in WE8EBCDIC1047 character set and WE8EBCDIC1047 NCHAR chara

About to export the entire database ...
. exporting tablespace definitions
. exporting profiles
. exporting user definitions
. exporting roles
. exporting resource costs
. exporting rollback segment definitions
. exporting database links
. exporting sequence numbers
. exporting directory aliases
. exporting foreign function library names
. exporting object type definitions
. exporting cluster definitions
. about to export SYSTEM's tables via Conventional Path ...
. . exporting table                       DEF$_AQCALL          0 rows exported
. . exporting table                      DEF$_AQERROR          0 rows exported
. . exporting table                     DEF$_CALLDEST          0 rows exported


****2600+ lines deleted      total of 2673

 . . exporting table            ECE_SPSO_ITEM_DET_X          0 rows exported
 . . exporting table                 ECE_SPSO_ITEMS          0 rows exported
 . . exporting table               ECE_SPSO_ITEMS_X          0 rows exported
 . . exporting table                 ECE_TP_DETAILS          0 rows exported
 . . exporting table                   ECE_TP_GROUP          0 rows exported
 . . exporting table                 ECE_TP_HEADERS          0 rows exported
 . . exporting table              ECE_XREF_CATEGORIES         73 rows exported
 . . exporting table                    ECE_XREF_DATA          0 rows exported
 . about to export JG's tables via Conventional Path ...
 . about to export APPS's tables via Conventional Path ...
 . about to export CTXSYS's tables via Conventional Path ...
 . exporting referential integrity constraints
 . exporting posttables actions
 . exporting synonyms
 . exporting views
 . exporting stored procedures
 . exporting triggers
 . exporting snapshots
 . exporting snapshot logs
 . exporting job queues
 . exporting refresh groups and children
 . exporting user history table
 . exporting default and system auditing options
 Export terminated successfully without warnings.
```

Figure 29. Export log from the S/390 job

7.2 Creating the target instance

Follow the steps in Chapter 5 or Chapter 10 (for a secondary instance) in the *Oracle Enterprise Edition for OS/390 Installation Guide* to create the target instance.

7.3 Running the import job

We were now ready to run the import job. The export file was on a cataloged tape. The Import parameters are as follows:

```
BUFFER=30720
FILE=/DD/IMPFILE
GRANTS=Y
COMMIT=Y
IGNORE=Y
INDEXES=Y
ROWS=Y
FULL=Y
LOG=/DD/LOGFILE
```

Figure 30 shows the ORACLE1.JCL.CNTL (ORAIMPRT) job we used to run the import.

```
//ORAIMPT JOB (999,POK),'ORACLE IMPORT',NOTIFY=ORACLE1,
//         TIME=1439,CLASS=A,MSGCLASS=X
//*
//* JCL TO RUN IMPORT FROM BATCH ENVIRONMENT
//*
//STEP1    EXEC ORAIMPV8,INDEX=MPMJ,LIBV=ORA8043,
//      PARM='SYS/CHANGE_ON_INSTALL PARFILE=/DD/PARFILE'
//ORA@MPMN DD DUMMY
//SYSIN    DD DUMMY
//PARFILE  DD DSN=ORACLE1.IMPORT.PARMS,
//            DISP=SHR
//IMPFILE  DD DSN=ORACLE1.MPMT.FULL.EXPORT,
//            DISP=SHR
//LOGFILE  DD DSN=ORACLE1.IMPORT.LOG,
//            DISP=SHR
```

Figure 30. JCL to run the import

Figure 31 on page 58 shows the log file that was produced during the import.

```
Connected to: Oracle8 Enterprise Edition Release 8.0.4.3.2 - Production
 PL/SQL Release 8.0.4.3.0 - Production

 Export file created by EXPORT&colon.V08.00.04 via direct path
 . importing SYSTEM's objects into SYSTEM

 IMP-00015: following statement failed because the object already exists:
  "CREATE TABLESPACE "DRSYS" DATAFILE  '/DSN/MPMT.ORA804.CONTEXT.DB2' SIZ
  "83584      , '/DSN/MPMT.ORA804.R11.DRSYS1/' SIZE 314814464      , '/DS
  "T.ORA804.R11.DRSYS2/' SIZE 210120704      , '/DSN/MPMT.ORA804.R11.DRSY
  "SIZE 524939264      , '/DSN/MPMT.ORA804.R11.DRSYS4/' SIZE 524939264
  "'/DSN/MPMT.ORA804.R11.SYSTEM3/' SIZE 524939264       DEFAULT STORAGE (
  "AL 20480 NEXT 20480 MINEXTENTS 1 MAXEXTENTS 249 PCTINCREASE 50) ONLINE
  "ANENT"


 54400+ Lines deleted              total of 54424 lines

IMP-00015: following statement failed because the object already exists:
 "CREATE TRIGGER "SYSTEM".repcatlogtrig"
 "AFTER UPDATE OR DELETE ON system.repcat$_repcatlog"
 ""
 "BEGIN"
 "  sys.dbms_alert.signal('repcatlog_alert', '');"
 "END;"
. importing APPS's objects into APPS
. importing SYSTEM's objects into SYSTEM
About to enable constraints...
. importing SCOTT's objects into SCOTT
. importing APPLSYS's objects into APPLSYS
. importing AX's objects into AX
. importing HR's objects into HR
. importing SSP's objects into SSP
. importing HXT's objects into HXT
. importing OTA's objects into OTA
. importing AR's objects into AR
. importing SYSTEM's objects into SYSTEM
. importing SCOTT's objects into SCOTT
. importing APPLSYS's objects into APPLSYS
. importing AX's objects into AX
. importing HR's objects into HR
. importing SSP's objects into SSP
. importing HXT's objects into HXT
. importing OTA's objects into OTA
Import terminated successfully with warnings
```

Figure 31. Log file after the import was completed

7.4 Checking the logs

Once the import is completed, you must go through the log in detail and handle any error messages. The log is a large file, so you must be very careful when going through looking for messages. We found that using the ISPF editor and searching for IMP, ERR, or WARNING was the easiest way to do this.

7.5 A customer's experience

One customer had a project to move an 8 GB Oracle Applications database from UNIX to OS/390. They exported to disk and ran FTP to move the data to OS/390. They installed Oracle7 Release 7.3.3.1 on OS/390 and applied the patch set to

bring the level up to 7.3.3.6.2. Then, a full database import was run. It took 26 hours to finish the job.

The customer used Strobe to identify a section of code that was used when the import job was creating the packages. Oracle took the code and optimized it. Oracle sent the customer a patch that improved the compile time. The customer reran the full database import and the job ran in 13 hours.

The only problem encountered was a signon issue with the encrypted passwords. Oracle gave the customer a job to run that decrypts the passwords and then re-encrypts them. This is required for the Oracle ERP Applications. After running this job, the signon problem disappeared.

Chapter 8. Upgrading Oracle Applications from 10.7 to 11

This chapter briefly describes the steps to upgrade Oracle Applications Release 10.7 running on S/390 to Oracle Applications Release 11 on S/390. There are additional steps for each application that must be performed. These are described in the Oracle Application Release 11 documentation.

After you have migrated the database from Oracle7 to Oracle8, the steps to run the upgrade process are as follows:

1. Prepare the AIX platform by installing the Oracle8 libraries, Oracle Application Server, and Oracle Developer. Our experiences with this are documented in the redbook *Oracle Applications 11 for OS/390 Installation Guide*, SG24-4980. That material is not repeated in this redbook.
2. Establish communication from the AIX platform to OS/390 with Net8.
3. Size the Release 11 database and increase the number of VSAM files on S/390.
4. Prepare the S/390 Oracle8 database by running the supplied SQL scripts.
5. Upgrade the passwords to Release 11 format.
6. Run the adaimgr script to load the Release 11 code.
7. Add patches and maintenance:
   - Add the NLS patches if using NLS.
   - Apply the 11.0.2 Maintenance Pack.
   - Apply the NLS 11.0.2 Maintenance Pack if using NLS.

Each of the activities is described briefly in the following sections. Follow the Oracle instructions in *Oracle Applications Release 11 for UNIX Installation*, A579803. This material is also covered in the redbook *Oracle Applications 11 for OS/390 Installation Guide*, SG24-4980. That material is not repeated in this redbook.

Note: This activity will take many hours to complete, since it is, in essence, the same procedure that occurs with a new installation of Oracle Applications, so plan on this step taking several days to complete.

8.1 Establish communication from the UNIX platform to OS/390 with NET8

The steps to establish communication to the new Oracle8 database on S/390 are as follows:

1. Add an entry for the new database in the tnsnames.ora file on UNIX:

```
MPM8  = (DESCRIPTION =
           (ADDRESS_LIST=
             (ADDRESS=
               (PROTOCOL=TCP)
               (HOST=wtsc48)
               (PORT=1521)
               (SSN=TNS8)
             )
           )
         )
```

2. Change the TWO_TASK variable:

```
export TWO_TASK=MPM8
```

3. Test it by connecting using the `svrmgrl` command:

```
svrmgrl connect system/manager@MPM8
```

The reply should indicate that you are connected to the database on OS/390.

8.2 Size the database and increase the number of OS/390 VSAM files

One of the preparation activities must be to run the sizing spreadsheet for the Release 11 applications you will use on OS/390. In one test, 50 MB of space was added for the FND and FNDIX tablespaces, with 20 MB added for all other tablespaces. However, during the installation, more space had to be added to the SHR1, GLIX, and GL tablespaces.

This should be planned for in advance and the allocations tested in the migration test run so that you do not have this problem when you are upgrading the production database for the live run.

8.3 Prepare the S/390 Oracle8 database by running the SQL scripts

Using SQLPLUS, the scripts that must be run are:

- sqlplus sys/manager @$APPL_TOP/admin/addb804.sql
- sqlplus system/manager @$APPL_TOP/admin/adsy804.sql
- sqlplus system/manager @$APPL_TOP/admin/adauapp.sql MANAGER
- sqlplus sys/manager @$APPL_TOP/admin/adsysctx.sql MANAGER FND TEMP
- sqlplus apps/apps @$APPL_TOP/admin/nlsmldel.sql
- sqlplus apps/appsr @$APPL_TOP/admin/nlsdeact.sql

8.4 Upgrade the passwords to Release 11 format

Before you start the upgrade AutoInstall process, you must acquire from Oracle Support the encryption patch that allows the AutoInstall to successfully upgrade passwords stored in a database with an EBCDIC-based character set.

8.5 Run adaimgr script to load the Release 11 code

When you run adaimgr, the script is the same as in an initial install. The types of questions you are asked are as follows:

```
You are about to install or upgrade Oracle Applications product tables in your
ORACLE database 'MPMT' using ORACLE executables in
/oranca/app/oracle/product/8.0.4'.

Is this the correct database “Yes”? Yes
```

Choose option **1** when presented with the following choices in the adaimgr script:

```
AutoInstall Main Menu
   -------------------------------------------------
1.    Select products to install or upgrade
2.    Select additional modules
3.    Choose database parameters
4.    Choose overall tasks and their parameters
5.    Run the selected tasks
6.    Exit AutoInstall
```

8.6 Apply all necessary patches

You need to apply the necessary patches. In one test, the necessary patches were:

- The NLS patches, if using NLS

Patches 668916:

- The 11.0.2 Maintenance Pack
- The NLS 11.0.2 Maintenance Pack, if using NLS

Chapter 9. Another example using the Export/Import utilities

This chapter provides another example of using the Export and Import utilities. We also show that if only one table fails, it is not necessary to rerun the entire import; instead, just the failed table can be imported.

The Oracle Application 11.03 Standard Benchmark has about 27.29 GB of data. Since there were no tapes cut for S/390, the import of the application data to MVS was done through NET8 from an NT machine using the export file generated by ORACLE. The export file was delivered in zip format, called export.dmp.Z, and contained 1097326 KB. The size of unzipped export file, called export.dmp, expanded to 11122313 KB.

The command initiated at the NT machine was as follows:

```
imp80 system/manager@VIS
FILE=export.dmp
ROWS=y
BUFFER=8388608
RECORDLENGTH=8388608
COMMIT=y
FEEDBACK=20000
LOG=import.log
IGNORE=y
FULL=y
```

To send the data to the S/390, the TWO_TASK environment variable was set with the value of VIS.

The import ran successfully but with the warning that some objects had become invalid. Also, one big table was not imported at its DDL creation time. The reason for the big table import failure was because there was not 1.5 GB of space available to satisfy its INITIAL_EXT request. The log file, import.log, took 10655 KB of space on NT.

9.1 Importing only one large table

To correct the table import failure, a large VSAM datafile was created and the command `ALTER TABLESPACE ADD DATAFILE` was executed to add the tablespace to the database instance on S/390. Then, another import process was started only for the big table, called GL_JE_LINES, from the NT machine. The following was the command syntax:

```
imp80 system/manager@VIS
FILE=export.dmp
ROWS=y
BUFFER=8388608
RECORDLENGTH=8388608
COMMIT=y
FEEDBACK=20000
LOG=implines.log
IGNORE=y
FROMUSER=apps
TABLES=GL_JE_LINES
```

Note that FULL= and FROMUSER= are mutually exclusive.

The content of the implines.log file is shown in Figure 32.

```
Connected to: Oracle8 Enterprise Edition Release 8.0.4.3.2 - Production
PL/SQL Release 8.0.4.3.0 - Production

Export file created by EXPORT:V08.00.05 via direct path
. importing GL's objects into GL
. . importing table                  "GL_JE_LINES"
.....................................................................
.....................................................................
.....................................................................
.....................................................................
.....................................................
                                                    7294127 rows imported
Import terminated successfully without warnings.
```

Figure 32. Import log for GL_JE_LINES

The five lines of dots shows the progress of the Import utility. It displays a dot for each 20000 rows imported, since the parameter FEEDBACK=20000 was specified on the command line.

9.2 Using Export/Import for backup purposes

Other export/import activities were done in the Oracle8 Release 8.0.4 database instance V11H on S/390. The following JCL was used to export the table GL.GL_JE_LINES for backup purposes. Note that the allocated block size was 24576. With the LRECL = 4096, the largest BLKSIZE that can be used is 28672. See Figure 33 on page 67.

```
//EXPTLINE JOB (0000,EXPT),'STEVEN C. HWUNG',MSGCLASS=X,
//          NOTIFY=&SYSUID,CLASS=A,REGION=4096K
//*
//*
//DELETE   EXEC PGM=IEFBR14
//DD1      DD DSN=MPMA.V11H.GL.JE.LINES.EXPORT,DISP=(OLD,DELETE)
//DD2      DD DSN=MPMA.V11H.GL.JE.LINES.EXPORT.LOG,DISP=(OLD,DELETE)
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//*
//EXPORT   EXEC PGM=EXP,PARM='PARFILE=/DD/PARFILE'
//*
//STEPLIB  DD DSN=MPMA.ORA804.CMDLOAD,DISP=SHR
//ORA@V11H DD DUMMY
//ORA$LIB  DD DSN=MPMA.ORA804.CMDLOAD,DISP=SHR
//ORA$ENV  DD DSN=MPMA.ORA.CNTL(ENVPERF),DISP=SHR
//SYSOUT   DD SYSOUT=*,DCB=(LRECL=132,BLKSIZE=1320,RECFM=VB)
//SYSERR   DD SYSOUT=*,DCB=(LRECL=132,BLKSIZE=1320,RECFM=VB)
//EXPDMP   DD DSN=MPMA.V11H.GL.JE.LINES.EXPORT,DISP=(NEW,CATLG),
//*           UNIT=3490,VOL=(,,,12),
//*           LABEL=(1,SL,EXPDT=99000)
//            UNIT=SYSDA,VOL=SER=ORA010,SPACE=(CYL,(1500,100),RLSE),
//            DCB=(RECFM=FB,LRECL=4096,BLKSIZE=24576)
//LOG      DD DSN=MPMA.V11H.GL.JE.LINES.EXPORT.LOG,DISP=(NEW,CATLG),
//            UNIT=SYSDA,SPACE=(CYL,(10,10),RLSE),
//            DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//PARFILE  DD *
BUFFER=40960
COMPRESS=N
CONSISTENT=Y
CONSTRAINTS=Y
DIRECT=Y
FEEDBACK=20000
FILE=/DD/EXPDMP
GRANTS=Y
HELP=N
INDEXES=Y
LOG=/DD/LOG
RECORD=N
RECORDLENGTH=24576'''
ROWS=Y
STATISTICS=COMPUTE
TABLES=GL_JE_LINES
USERID=GL/GL
//*
//SYSIN    DD DUMMY
The following JCL was used to import the table GL.GL_AUTOMATIC_POSTING_SETS into the same
database instance V11H.
//IMPTPOST JOB (0000,EXPT),'STEVEN C. HWUNG',MSGCLASS=X,
//          NOTIFY=&SYSUID,CLASS=A,REGION=4096K
//*
//IMPORT   EXEC PGM=IMP,PARM='PARFILE=/DD/PARFILE'
//*
//STEPLIB  DD DSN=MPMA.ORA804.CMDLOAD,DISP=SHR
//ORA@V11H DD DUMMY
//ORA$LIB  DD DSN=MPMA.ORA804.CMDLOAD,DISP=SHR
//ORA$ENV  DD DSN=MPMA.ORA804.PARMLIB(MPMPARM),DISP=SHR
//SYSOUT   DD SYSOUT=*,DCB=(LRECL=132,BLKSIZE=1320,RECFM=VB)
//SYSERR   DD SYSOUT=*,DCB=(LRECL=132,BLKSIZE=1320,RECFM=VB)
//EXPDMP   DD DSN=IBMU01.AUTOPOST.DMP,DISP=SHR
//LOG      DD DSN=IBMU01.AUOTPOST.IMPLOG,DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,SPACE=(CYL,(10,10),RLSE),
//            DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//PARFILE  DD *
BUFFER=8388608
COMMIT=Y
DESTROY=Y
FEEDBACK=20000
FILE=/DD/EXPDMP
GRANTS=Y
HELP=N
IGNORE=Y
INDEXES=Y
LOG=/DD/LOG
RECORDLENGTH=8388608
ROWS=Y
FROMUSER=GL
TABLES=GL_AUTOMATIC_POSTING_SETS
USERID=SYSTEM/MANAGER
/*
//*
//SYSIN    DD DUMMY
```

Figure 33. JCL used to export the table GL_JE_LINES

Appendix A. Problems encountered

The following is a list of some of the problems we encountered:

- Export file incomplete.
  The export was done on the RS/6000 using parameter DIRECT=Y and therefore could not export some system tables with column types of LOB. We changed the parameters to DIRECT=N to resolve this problem.
- FTP problem sending file to S/390 in the correct format.
  This occurred when we did not specify `bin` before we did the FTP.
- FTP problem when the target file ran out of extents.
  The export file is very large, so you must allocate a sufficiently large S/390 data set to receive the FTP file.
- Running the import command from the TSO command line.
  The Import utility could not open the parameter file, since we had not specified /DSN/ in front of the data set name.

  The Import utility then started from the TSO session, but the utility was writing the log file back to the TSO session. This tied up the TSO session. We then ran the import job as a batch job.
- The import log had many memory error messages.
  We changed many parameters to try to resolve this problem such as SHARED_POOL_SIZE = 40,000,000 and DB_BLOCK_BUFFERS= 37,200. When we changed DB_BLOCK_BUFFERS = 100,000, we eliminated some of the messages. However, this problem was solved only when we changed the Import parameter to BUFFER=13413000.

  We did run some BSTAT and ESTAT queries to help us determine what was the optimum size. The jobs are shown in A.2.5, “BSTAT job” on page 76 and A.2.6, “ESTAT job” on page 77.
- Resetting the database instance after a bad import attempt.
  Usually, the import will not work the first time, so you will have to have a process to clean up the database. See the following section.

A.1 Resetting the database to restart the import

These are the steps to clean up the database to retry the import:

1. Reset the log data set.
   We deleted and defined the file MPMD.IMPORT.LOG.
2. Bring down the Oracle instance.
   Issue the command:

   ```
   MPMD START SVRMGRL COMMAND='SHUTDOWN IMMEDIATE'
   ```
3. Run the JOB ORPIJEOO to delete and reallocate the database as shown in A.2.1, “ORPIJE00 job” on page 70.
4. Run the jobs to delete, reallocate, and format the tablespaces as shown in A.2.2, “Tablespace jobs” on page 74.
5. Start the Oracle instance with the create option:

   ```
   START ORAMPMD, START=CREATE
   ```

6. Run the install job ORPIJH00 to set up the views as shown in A.2.3, “ORPIJH00 job” on page 75.
7. Run the import job as shown in A.2.4, “Import job” on page 76.

A.2 Jobs used to resolve the problems

The following is the JCL we used to resolve some of our problems.

A.2.1 ORPIJE00 job

```
//ORPIJE00 JOB (0000,OR),'ORACLE INSTALL',CLASS=A,                      00020000
//          MSGCLASS=X,PRTY=15,MSGLEVEL=(1,1),NOTIFY=&SYSUID            00020002
//*                                                                     00020004
//*                                                                     00020006
//*                                                                     00020008
//* LIBRARY ===> MPMD.ORA804.INSTLIB                                    00020010
//* MEMBER  ===> ORPIJE00                                               00020012
//* AUTHOR  ===> ORACLE CORPORATION                                     00020014
//*-------------------------------------------------------------*       0020016
//*             ORACLE CORPORATION                              *     00020018
//*             CORPORATE PRODUCT SUPPORT                       *     00020020
//*             500 ORACLE PARKWAY                              *     00020022
//*             REDWOOD SHORES CALIFORNIA 94065 U.S.A           *     00020024
//*                                                             *     00020026
//*                              ATTENTION: MVS PRODUCT SUPPORT *     00020028
//*                                         650/506-7000        *     00020030
//*  ORACLE SERVER LICENSED MATERIALS - PROPERTY OF ORACLE CORP.*     00020032
//*  CONTAINS RESTRICTED MATERIALS OF ORACLE CORP.              *     00020034
//*  REFER TO COPYRIGHT INSTRUCTIONS DOCUMENT NUMBER XXXX-XXXX. *     00020036
//*  (C) COPYRIGHT ORACLE CORP. 1986                            *     00020038
//*-------------------------------------------------------------------* 00020040
//*  JOB DESCRIPTION: DELETE AND ALLOCATE THE  ORACLE/MVS DATABASE,* 00020042
//*                   CONTROL, AND REDO LOG VSAM CLUSTERS.       *     00020044
//*-------------------------------------------------------------------* 00020046
//*                                                                     00020048
//STEPAMS1 EXEC PGM=IDCAMS                                              00020050
//SYSPRINT DD SYSOUT=*                                                  00020052
//SYSIN    DD *                                                         00020054
 /*-----------------------------------------------------*/              00020056
 /*                                                     */              00020058
 /*       DELETE THE ORACLE DB, CONTROL, AND LOG        */              00020060
 /*       CLUSTERS.                                     */              00020062
 /*                                                     */              00020064
 /*-----------------------------------------------------*/              00020066
   DELETE (MPMD.DBF.SYSTEM.DB1) -                                       00020068
          CLUSTER PURGE                                                 00020070
   DELETE (MPMD.DBF.CONTEXT.DB2) -                                      00020072
          CLUSTER PURGE                                                 00020074
   DELETE (MPMD.DBF.USER3.DB3) -                                        00020076
          CLUSTER PURGE                                                 00020078
   DELETE (MPMD.DBF.CONTROL2) -                                         00020080
          CLUSTER PURGE                                                 00020082
   DELETE (MPMD.DBF.CONTROL1) -                                         00020084
          CLUSTER PURGE                                                 00020086
   DELETE (MPMD.DBF.SNAPCF) -                                           00020088
          CLUSTER PURGE                                                 00020090
```

```
  DELETE (MPMD.DBF.LOG2) -                                              00020092
         CLUSTER PURGE                                                  00020094
  DELETE (MPMD.DBF.LOG1) -                                              00020096
         CLUSTER PURGE                                                  00020098
  SET MAXCC=0    /* REMOVE AFTER FIRST TIME EXECUTION */               00020100
  SET LASTCC=0   /* REMOVE AFTER FIRST TIME EXECUTION */               00020102
/*-------------------------------------------------------*/             00020104
/*                                                       */             00020106
/*       DEFINE THE DB (ORACLE DATA BASE) CLUSTERS.      */             00020108
/*                                                       */             00020110
/*-------------------------------------------------------*/             00020112
  DEFINE CLUSTER                                  -                     00020114
    (                                             -                     00020116
     NAME(MPMD.DBF.SYSTEM.DB1)  -                                       00020118
         VOLUMES(ORACLH)                 -                            00020120
         CONTROLINTERVALSIZE(4096)                -                    00020122
         RECORDS(12000 0)                      -                      00020124
         RECORDSIZE(4089 4089)                    -                    00020126
         NONSPANNED                               -                    00020128
         UNIQUE                                   -                    00020130
         NONINDEXED                               -                    00020132
         SHR(3 3)                                 -                    00020134
    )                                             -                    00020136
    DATA                                          -                    00020138
    (                                             -                    00020140
    NAME(MPMD.DBF.SYSTEM.DB1.DATA) -                                    00020142
    )                                                                   00020144
/*-------------------------------------------------------*/ 00020146
/*                                                     */             00020148
/*       LIST THE CATALOG CLUSTER ENTRY                */             00020150
/*                                                     */             00020152
/*-------------------------------------------------------*/ 00020154
  LISTC   ENT(MPMD.DBF.SYSTEM.DB1) ALL                                 00020156
  DEFINE CLUSTER                                  -                     00020158
    (                                             -                    00020160
     NAME(MPMD.DBF.CONTEXT.DB2)  -                                      00020162
         VOLUMES(ORACLH)                 -                            00020164
         CONTROLINTERVALSIZE(4096)                -                    00020166
         RECORDS(10000 0)                      -                      00020168
         RECORDSIZE(4089 4089)                    -                    00020170
         NONSPANNED                               -                    00020172
         UNIQUE                                   -                    00020174
         NONINDEXED                               -                    00020176
         SHR(3 3)                                 -                    00020178
    )                                             -                    00020180
    DATA                                          -                    00020182
    (                                             -                    00020184
    NAME(MPMD.DBF.CONTEXT.DB2.DATA) -                                   00020186
    )                                                                   00020188
/*-------------------------------------------------------*/ 00020190
/*                                                     */             00020192
/*       LIST THE CATALOG CLUSTER ENTRY                */             00020194
/*                                                     */             00020196
/*-------------------------------------------------------*/ 00020198
  LISTC   ENT(MPMD.DBF.CONTEXT.DB2) ALL                                00020200
  DEFINE CLUSTER                                  -                     00020202
    (                                             -                    00020204
     NAME(MPMD.DBF.USER3.DB3)  -                                        00020206
```

```
        VOLUMES(ORACLH)                 -                                00020208
        CONTROLINTERVALSIZE(4096)               -                        00020210
        RECORDS(4000 0)                     -                            00020212
        RECORDSIZE(4089 4089)                   -                        00020214
        NONSPANNED                              -                        00020216
        UNIQUE                                  -                        00020218
        NONINDEXED                              -                        00020220
        SHR(3 3)                                -                        00020222
    )                                           -                        00020224
    DATA                                        -                        00020226
    (                                           -                        00020228
    NAME(MPMD.DBF.USER3.DB3.DATA) -                                      00020230
    )                                                                    00020232
/*-------------------------------------------------------*/ 00020234
/*                                                       */             00020236
/*       LIST THE CATALOG CLUSTER ENTRY                  */             00020238
/*                                                       */             00020240
/*-------------------------------------------------------*/ 00020242
  LISTC   ENT(MPMD.DBF.USER3.DB3) ALL                                    00020244
/*-------------------------------------------------------*/ 00020246
/*                                                       */             00020248
/*       DEFINE THE ORACLE CONTROL FILE CLUSTERS.        */             00020250
/*                                                       */             00020252
/*-------------------------------------------------------*/ 00020254
  DEFINE CLUSTER                                -                        00020256
    (                                           -                        00020258
     NAME(MPMD.DBF.CONTROL2) -                                           00020260
        VOLUMES(ORACLH)                         -                        00020262
        CONTROLINTERVALSIZE(4096)               -                        00020264
        RECORDS(1100 500)                   -                            00020266
        RECORDSIZE(4089 4089)                   -                        00020268
        NONSPANNED                              -                        00020270
        UNIQUE                                  -                        00020272
        NONINDEXED                              -                        00020274
        SHR(3 3)                                -                        00020276
    )                                           -                        00020278
    DATA                                        -                        00020280
    (                                           -                        00020282
    NAME(MPMD.DBF.CONTROL2.DATA) -                                       00020284
    )                                                                    00020286
/*-----------------------------------------------------*/               00020288
/*                                                     */               00020290
/*       LIST THE CATALOG CLUSTER ENTRY                */               00020292
/*                                                     */               00020294
/*-----------------------------------------------------*/               00020296
  LISTC   ENT(MPMD.DBF.CONTROL2) ALL                                     00020298
  DEFINE CLUSTER                                -                        00020300
    (                                           -                        00020302
     NAME(MPMD.DBF.CONTROL1) -                                           00020304
        VOLUMES(ORACLH)                         -                        00020306
        CONTROLINTERVALSIZE(4096)               -                        00020308
        RECORDS(1100 500)                   -                            00020310
        RECORDSIZE(4089 4089)                   -                        00020312
        NONSPANNED                              -                        00020314
        UNIQUE                                  -                        00020316
        NONINDEXED                              -                        00020318
        SHR(3 3)                                -                        00020320
    )                                           -                        00020322
```

```
    DATA                                                -                00020324
    (                                                   -                00020326
    NAME(MPMD.DBF.CONTROL1.DATA) -                                       00020328
    )                                                                    00020330
/*-----------------------------------------------------*/               00020332
/*                                                     */               00020334
/*       LIST THE CATALOG CLUSTER ENTRY                */               00020336
/*                                                     */               00020338
/*-----------------------------------------------------*/               00020340
  LISTC   ENT(MPMD.DBF.CONTROL1) ALL                                     00020342
  DEFINE CLUSTER                                    -                  00020344
    (                                               -                  00020346
     NAME(MPMD.DBF.SNAPCF) -                                           00020348
         VOLUMES(ORACLH)                            -                  00020350
         CONTROLINTERVALSIZE(4096)                  -                  00020352
         RECORDS(1100 500)                    -                       00020354
         RECORDSIZE(4089 4089)                      -                  00020356
         NONSPANNED                                 -                  00020358
         UNIQUE                                     -                  00020360
         NONINDEXED                                 -                  00020362
         SHR(3 3)                                   -                  00020364
    )                                               -                  00020366
    DATA                                            -                  00020368
    (                                               -                  00020370
    NAME(MPMD.DBF.SNAPCF.DATA) -                                         00020372
    )                                                                    00020374
/*-----------------------------------------------------*/               00020376
/*                                                     */               00020378
/*       LIST THE CATALOG CLUSTER ENTRY                */               00020380
/*                                                     */               00020382
/*-----------------------------------------------------*/               00020384
  LISTC   ENT(MPMD.DBF.SNAPCF) ALL                                       00020386
/*-------------------------------------------------------*/ 00020388
/*                                                       */             00020390
/*       DEFINE THE ORACLE REDO LOG FILE CLUSTERS.       */             00020392
/*                                                       */             00020394
/*-------------------------------------------------------*/ 00020396
  DEFINE CLUSTER                                    -                  00020398
    (                                               -                  00020400
     NAME(MPMD.DBF.LOG2) -                                             00020402
         VOLUMES(ORACLH)                            -                  00020404
         CONTROLINTERVALSIZE(4096)                  -                  00020406
         RECORDS(4000 0)                    -                         00020408
         RECORDSIZE(4089 4089)                      -                  00020410
         NONSPANNED                                 -                  00020412
         UNIQUE                                     -                  00020414
         NONINDEXED                                 -                  00020416
         SHR(3 3)                                   -                  00020418
    )                                               -                  00020420
    DATA                                            -                  00020422
    (                                               -                  00020424
    NAME(MPMD.DBF.LOG2.DATA) -                                           00020426
    )                                                                    00020428
/*-----------------------------------------------------*/               00020430
/*                                                     */               00020432
/*       LIST THE CATALOG CLUSTER ENTRY                */               00020434
/*                                                     */               00020436
/*-----------------------------------------------------*/               00020438
```

```
  LISTC   ENT(MPMD.DBF.LOG2) ALL                                    00020440
  DEFINE CLUSTER                                  -                 00020442
    (                                             -                 00020444
     NAME(MPMD.DBF.LOG1) -                                          00020446
         VOLUMES(ORACLH)                          -                 00020448
         CONTROLINTERVALSIZE(4096)                -                 00020450
         RECORDS(4000 0)                    -                      00020452
         RECORDSIZE(4089 4089)                    -                 00020454
         NONSPANNED                               -                 00020456
         UNIQUE                                   -                 00020458
         NONINDEXED                               -                 00020460
         SHR(3 3)                                 -                 00020462
    )                                             -                 00020464
    DATA                                          -                 00020466
    (                                             -                 00020468
    NAME(MPMD.DBF.LOG1.DATA) -                                      00020470
    )                                                               00020472
 /*-----------------------------------------------------*/          00020474
 /*                                                     */          00020476
 /*       LIST THE CATALOG CLUSTER ENTRY                */          00020478
 /*                                                     */          00020480
 /*-----------------------------------------------------*/          00020482
  LISTC   ENT(MPMD.DBF.LOG1) ALL                                    00020484
/*                                                                  00020486
```

A.2.2 Tablespace jobs

```
//ADDDAT1 JOB (0000,OR),'ORACLE INSTALL',CLASS=A,                   00020000
//          MSGCLASS=X,PRTY=15,MSGLEVEL=(1,1),NOTIFY=&SYSUID        00020002
//*-------------------------------------------------------------------*
00020040
//*  JOB DESCRIPTION: ALLOCATE VSAM FILE RUN CCF AGAINST THE FILE, *  00020042
//*    CREATE A TABLESPACE AND ADD THE DATAFILE                   *  00020044
//*-------------------------------------------------------------------*
00020046
//*                                                                 00020048
//STEPAMS1 EXEC PGM=IDCAMS                                          00020050
//SYSPRINT DD SYSOUT=*                                              00020052
//SYSIN    DD *                                                     00020054
 /*-----------------------------------------------------*/          00020056
 /*                                                     */          00020058
 /*       DELETE THE ORACLE DB, CONTROL, AND LOG       */          00020060
 /*       CLUSTERS. CHANGE TO CORRECT DBF NAME        */          00020062
 /*                                                     */          00020064
 /*-----------------------------------------------------*/          00020066
  DELETE (MPMD.DBF.UDAT1)                                         00020068
         CLUSTER PURGE                                            00020070
  SET MAXCC=0    /* REMOVE AFTER FIRST TIME EXECUTION */          00020100
  SET LASTCC=0   /* REMOVE AFTER FIRST TIME EXECUTION */          00020102
 /*-----------------------------------------------------*/          00020104
 /*                                                     */          00020106
 /*       DEFINE THE DB (ORACLE DATA BASE) CLUSTERS.   */          00020108
 /*                                                     */          00020110
 /*-----------------------------------------------------*/          00020112
  DEFINE CLUSTER                                  -                 00020114
    (                                             -                 00020116
     NAME(MPMD.DBF.UDAT1)                    -                      00020118
         VOLUMES(ORACLJ)                          -                 00020120
```

```
          CONTROLINTERVALSIZE(4096)              -                  00020122
          MEGABYTES(525)                         -                  00020124
          RECORDSIZE(4089 4089)                  -                  00020126
          NONSPANNED                             -                  00020128
          UNIQUE                                 -                  00020130
          NONINDEXED                             -                  00020132
          SHR(3 3)                               -                  00020134
     )                                           -                  00020136
     DATA                                        -                  00020138
     (                                           -                  00020140
     NAME(MPMD.DBF.UDAT1.DATA) -                                    00020142
     )                                                              00020144
/*                                                                  00020486
//ORACCF EXEC PGM=CCF,REGION=3072K,
//      PARM='/DSN/MPMD.DBF.UDAT1/'
//STEPLIB DD DSN=MPMN.ORA804.CMDLOAD,DISP=SHR
//SYSOUT DD SYSOUT=*,DCB=(LRECL=132,BLKSIZE=1320,RECFM=VB)
//SYSERR DD SYSOUT=*,DCB=(LRECL=132,BLKSIZE=1320,RECFM=VB)
//SYSIN DD DUMMY
//*
```

We had a total of 11 jobs like this to execute.

A.2.3 ORPIJH00 job

```
//DUTCH1  JOB (0000,OR),'ORACLE INSTALL',CLASS=A,
//           MSGCLASS=X,PRTY=15,MSGLEVEL=(1,1),NOTIFY=&SYSUID
//*
//*
//*
//* LIBRARY ===> MPMD.ORA804.INSTLIB
//* MEMBER  ===> ORPIJH00
//* AUTHOR  ===> ORACLE CORPORATION
//*-------------------------------------------------------------------*
//*              ORACLE CORPORATION                                   *
//*              CORPORATE PRODUCT SUPPORT                            *
//*              500 ORACLE PARKWAY                                   *
//*              REDWOOD SHORES CALIFORNIA 94065 U.S.A                *
//*                                                                   *
//*                                ATTENTION: MVS PRODUCT SUPPORT     *
//*                                           650/506-7000            *
//*  ORACLE SERVER LICENSED MATERIALS - PROPERTY OF ORACLE CORP.      *
//*  CONTAINS RESTRICTED MATERIALS OF ORACLE CORP.                    *
//*  REFER TO COPYRIGHT INSTRUCTIONS DOCUMENT NUMBER XXXX-XXXX.       *
//*  (C) COPYRIGHT ORACLE CORP. 1986                                  *
//*-------------------------------------------------------------------*
//*  JOB DESCRIPTION: INITIALIZE THE ORACLE DATABASE DICTIONARY.      *
//*-------------------------------------------------------------------*
//*
//ORACATLG EXEC ORADBADD,REGION=6M
//*
//ORA@MPMD DD DUMMY      <=== ORA@SSN (ORACLE SUBSYSTEM NAME).
//ORA$FNA  DD DSN=MPMD.ORA804.PARMLIB(SERVFNA),DISP=SHR
//SQL      DD DISP=SHR,DSN=MPMD.ORA804.SQL
//SYSIN    DD *
SET ECHO ON
CONNECT INTERNAL
@CATALOG
```

```
@CATPROC
/*
//SYSPRINT DD SYSOUT=*
//*
//
//* THE FOLLOWING FOUR PROCEDURES WERE REMOVED FROM THE EXECUTION
//* STREAM BECAUSE BY RUNNING THESE ERRORS WILL BE PRODUCED WHEN
//* AN IMPORT WITH THE FULL=Y OPTION SPECIFIED.
@CATHS
@CATHSOCI
@OTRCSVR
@CATREP
```

A.2.4 Import job

```
//IMPORTVI JOB (0000,OR),'IMPORT V8',CLASS=A,
//            MSGCLASS=X,MSGLEVEL=(1,1),PRTY=15,NOTIFY=&SYSUID
//IMPORT   EXEC PGM=IMP,PARM='SYSTEM/MANAGER PARFILE=/DD/PARFILE'
//STEPLIB  DD   DSN=MPMD.ORA804.CMDLOAD,DISP=SHR
//SYSOUT   DD   SYSOUT=*,DCB=(LRECL=132,BLKSIZE=1320,RECFM=VB)
//SYSERR   DD   SYSOUT=*
//ORA@MPMD DD   DUMMY
//SYSIN    DD   DUMMY
//PARFILE  DD   DSN=MPMD.IMPORT.PARFILE,DISP=SHR
//IMPFILE  DD   DSN=MPMD.EXPORTVI,DISP=SHR
//LOGFILE  DD   DSN=MPMD.IMPORT.LOG,DISP=SHR
```

A.2.5 BSTAT job

```
//BSTAT2   JOB (0000,OR),'ORACLE INSTALL',CLASS=A,                      00020000
//          MSGCLASS=X,PRTY=15,MSGLEVEL=(1,1),NOTIFY=&SYSUID            00020002
//*                                                                     00020004
//*                                                                     00020006
//*                                                                     00020008
//* LIBRARY ===> MPMD.ORA804.SQL                                        00020010
//* MEMBER  ===> UTLBSTAT                                               00020012
//* AUTHOR  ===> ORACLE CORPORATION                                     00020014
//*-------------------------------------------------------------------*
00020016
//*             ORACLE CORPORATION                              *     00020018
//*             CORPORATE PRODUCT SUPPORT                       *     00020020
//*             500 ORACLE PARKWAY                              *     00020022
//*             REDWOOD SHORES CALIFORNIA 94065 U.S.A           *     00020024
//*                                                             *     00020026
//*                           ATTENTION: MVS PRODUCT SUPPORT    *     00020028
//*                                      650/506-7000           *     00020030
//*  ORACLE SERVER LICENSED MATERIALS - PROPERTY OF ORACLE CORP.  *     00020032
//*  CONTAINS RESTRICTED MATERIALS OF ORACLE CORP.                *     00020034
//*  REFER TO COPYRIGHT INSTRUCTIONS DOCUMENT NUMBER XXXX-XXXX.   *     00020036
//*  (C) COPYRIGHT ORACLE CORP. 1986                              *     00020038
//*-------------------------------------------------------------------*
00020040
//*  JOB DESCRIPTION: RUN BSTAT TO GENERATE AND CLEAR TABLES      *     00020042
//*-------------------------------------------------------------------*
00020046
//*
//ORACATLG EXEC ORADBADD,REGION=6M
//*
```

```
//ORA@MPMD DD DUMMY      <=== ORA@SSN (ORACLE SUBSYSTEM NAME).
//ORA$FNA  DD DSN=MPMD.ORA804.PARMLIB(SERVFNA),DISP=SHR
//SQL      DD DISP=SHR,DSN=MPMD.ORA804.SQL
//SYSIN    DD *
SET ECHO ON
CONNECT INTERNAL
@UTLBSTAT
/*
//SYSPRINT DD SYSOUT=*
//*
```

A.2.6 ESTAT job

```
//ESTAT2   JOB (0000,OR),'ORACLE INSTALL',CLASS=A,                    00020000
//          MSGCLASS=X,PRTY=15,MSGLEVEL=(1,1),NOTIFY=&SYSUID          00020002
//*                                                                   00020004
//*                                                                   00020006
//*                                                                   00020008
//* LIBRARY ===> MPMD.ORA804.SQL                                      00020010
//* MEMBER  ===> UTLESTAT                                             00020012
//* AUTHOR  ===> DENNY AND LOU INC                                    00020014
//*-------------------------------------------------------------------*
00020016
//*             ORACLE CORPORATION                                *   00020018
//*             CORPORATE PRODUCT SUPPORT                         *   00020020
//*             500 ORACLE PARKWAY                                *   00020022
//*             REDWOOD SHORES CALIFORNIA 94065 U.S.A             *   00020024
//*                                                               *   00020026
//*                              ATTENTION: MVS PRODUCT SUPPORT   *   00020028
//*                                         650/506-7000          *   00020030
//*  ORACLE SERVER LICENSED MATERIALS - PROPERTY OF ORACLE CORP.  *   00020032
//*  CONTAINS RESTRICTED MATERIALS OF ORACLE CORP.                *   00020034
//*  REFER TO COPYRIGHT INSTRUCTIONS DOCUMENT NUMBER XXXX-XXXX.   *   00020036
//*  (C) COPYRIGHT ORACLE CORP. 1986                              *   00020038
//*-------------------------------------------------------------------*
00020040
//*  JOB DESCRIPTION: RUN ESTAT STATISTICS                        *   00020042
//*-------------------------------------------------------------------*
00020046
//*
//ORACATLG EXEC ORADBADD,REGION=6M
//*
//ORA@MPMD DD DUMMY      <=== ORA@SSN (ORACLE SUBSYSTEM NAME).
//ORA$FNA  DD DSN=MPMD.ORA804.PARMLIB(SERVFNA),DISP=SHR
//SQL      DD DISP=SHR,DSN=MPMD.ORA804.SQL
//SYSIN    DD *
SET ECHO ON
CONNECT INTERNAL
@UTLESTAT
/*
//SYSPRINT DD SYSOUT=*
//*
```

Appendix B. Running the Export/Import utilities interactively

This appendix describes how to use the Oracle Export/Import utilities to port an Oracle8 Enterprise Edition Version 8.0.4 database from an RS/6000 to a S/390 Model 9672 R76 (a G5 R model). The database ported was the Oracle Applications Vision demonstration database (approximately 4.5 GB), which provides a sample set of transaction data for a fictitious company that primarily uses Oracle Applications Release 11 products.

Steps for moving a database include:

1. Exporting the database
2. Setting up the S/390 for the database
3. Using the TWO_TASK variable or the FTP command
4. Importing the database
5. Verifying the database

B.1 Exporting the database

The database was exported using Oracle's Export utility. In this case, the utility was used interactively. The smaller type below indicates the DBA commands. Comments indicated by **Note**: are our comments.

Note: The first three responses indicate that the user doing the export is system, the buffer size is 30000, and the name of the export file will be expdat.dmp.

```
exp
Username: system
Password:
Connected to: Oracle8 Enterprise Edition Release 8.0.5.1.0
- Production PL/SQL Release 8.0.5.1.0 - Production
Enter array fetch buffer size: 4096 >30000
```

Note: Array fetch is the concept of fetching multiple rows per trip to the database.

```
Export file: ./expdat.dmp > expdat.dmp
 (1) E(ntire), (2)U(sers), or (3)T(ables): (2)U > E
Export grants (yes/no): yes > y
Export table data (yes/no): yes > y
Compress extents (yes/no): yes > n
```

Note: A DBA will typically use compression of extents when reorganizing a table.

The export utility responds with the following:

```
About to export......
. exporting snapshots
. exporting snapshot logs
. exporting database links

Many lines not shown here
```

Note: Export time will vary, depending on the database size and hardware being used.

Note: To copy the database while it is active, you must use the `consistent=Y` parameter. This places a much higher overhead on the rollback segments for the

RDBMS to ensure data integrity in an active database by ensuring a read-consistent view.

B.2 Setting up S/390 for the database

Next, you must create the database on the S/390 using the following steps:

1. Set up the VSAM files.
2. Create the database.
3. Define the tablespaces:
   - For your application
   - For TEMP
   - The rollback tablespace for the rollback segments
4. Put the tablespaces online.
5. Check the INITORA file to insure that the SHARED_POOL_SIZE and DB_BLOCK_BUFFERS sizes are adequate.
6. If you have changed the INITORA file, you must shut down and restart the Oracle database.
7. Rollback segments can be added dynamically.

B.3 Using the TWO_TASK variable or the FTP command

There are two ways to transfer the file from RS/6000 to the S/390:

- Use the FTP command to move the export file to the S/390.
- Use the import command on the RS/6000 to move the file to the S/390 using the TWO_TASK option.

B.3.1 Using the FTP command

Note: If your export file is large, you can preallocate a large file on OS/390 or you can use the `SITE` command to create a large file.

```
ftp "to the IP Address"
connected to "IP Address"
name UserID
password
UserID is logged on, working directory is "oracle1."
ftp>
put expdat.dmp 'oracle1.expdat.dmp'
port request OK
storing dataset oracle1.mona
transfer complete (data was truncated)
```

The data was truncated because the FTP command did not allocate a large enough file for our export file. We retried using the site command to set up a large file on OS/390. We had a DASDpack called ORACL4 that had available space, so we issued the following command in the FTP session:

```
ftp>
quote site cylinders primary=2000 secondary=500 volume=oracl4
```

FTP responded with SITE command accepted, then we reissued the `put` command, which completed successfully:

```
put expdat.dmp 'oracle1.expdat1.dmp'
```

B.3.2 Importing from AIX

In this case, the import was done using the TWO_TASK option. To import from the AIX RS/6000 side straight through to the S/390, the TWO_TASK option was set to the SID (Oracle ID of the instance) of the machine to which it was being imported. This allowed the AIX RS/6000 machine to import into the S/390. In our case, the SID was MPMI.

The command on the AIX side was:

```
export TWO_TASK=MPMI
```

To verify, you can issue the `echo` command:

```
echo $TWO_TASK
```

At this point, you should see the SID. In our case, we saw:

```
MPMI
```

You can also check to be sure that you are pointing to the SID on the S/390 by invoking the `SVRMGRL` command and noting if you are actually seeing the SVRMGR from the S/390.

B.4 Importing the database

Note: As stated earlier, the Export/Import utilities can be run interactively or with a script file (also known as a parameter file). In this case, a parameter file was used. The parameter file was created at /home/oracle/scripts/import.par.

The parameter file looked like the following:

```
BUFFER=30720
FILE=/ora805/expdat.dmp
COMMIT=Y
IGNORE=Y
GRANTS=N
INDEXES=N
FULL=Y
LOG=/ora805/import.log
```

Note: We used the LOG option so that we could analyze the log if anything went wrong.

Previously, we had set the TWO_TASK variable to point to the Oracle instance on S/390. With the TWO_TASK variable set, we use the Oracle import command on AIX. The RS/6000 machine imports into the database on S/390.

As shown in B.3, “Using the TWO_TASK variable or the FTP command” on page 80, the command to set the TWO_TASK variable on the AIX side is:

```
export TWO_TASK=MPMI
```

The actual command to do the import using a parameter file is:

```
imp userid=system/manager parfile=/home/oracle/scripts/import.par
```

If you use the FTP command to move the file from the RS/6000 to the S/390, the method of executing the command will be different, but you will still use the same Oracle import command (IMP) either through the command line interface or using JCL.

The Oracle command is passed to the S/390 through the use of a call to the CMDLOAD utility where MPMI.ORA804 points to the path for the CMDLOAD. The actual command is:

```
CALL 'MPMI.ORA804.CMDLOAD(IMP USERID=SYSTEM/MANAGER
PARFILE=/HOME/ORACLE/SCRIPTS/IMPORT.PAR'
```

Note: The time needed to do the import will vary, based on your hardware and the size of your database.

B.5 Verifying the database

Verify the accuracy of your database by querying the V$ tables and the application tables.

Note: Don't forget to change the INITORA back to its previous state if you made any changes during the export/import process.

Appendix C. Installing the Oracle8i Database on OS/390

This chapter describes the various steps needed and the experiences encountered during the installation of the Oracle8*i* Database on OS/390.

We followed the steps for installation without SMP/E given in Chapter 5 in *Oracle8i Enterprise Edition for OS/390 Installation Guide*. To perform this installation, we assume that you are familiar with OS/390, JCL, and TSO; that you can work with ISPF; and that you know SDSF and basic OS/390 operator commands.

We further assume that you have a running OS/390 system and a DASD volume for the installation.

The number of installation steps will vary, depending on your selection of Oracle products. The main ones are:

- Add the Oracle Subsystem names to the SYS1.PARMLIB member.
- Add the Oracle AUTHLOAD library as an APF-authorized file.
- Unload the initial job from the Oracle installation tape.
- Set up the Oracle ISPF libraries.
- Execute the installation dialogue.
- Run the installation jobs.
- Edit the startup procedures and parmlib members.
- Create the Oracle Database.
- Run the verification programs.

C.1 Add the subsystem names

We added the Oracle subsystem names by using ISPF. You need to specify two subsystems:

- MPM for the Oracle8*i* OS/390 server
- TNS for the Oracle NET8.

For this installation we chose MPM5 as the subsystem name of the Oracle database, with an asterisk (*) as the communication character (COMCHAR) and TNS5 as the subsystem name for Net8, with an at sign (@) as the COMCHAR.

We already had an MPM8 instance for Oracle 8.0.4, so we choose MPM5 for our instance of Oracle 8.1.5.

Recent MVS releases and OS/390 allow you to add subsystem names dynamically by using either an operator command under SDSF or on the system console. Adding a subsystem name dynamically saves you performing an IPL. Make sure you have the new subsystem names in your IEFSSNxx PARMLIB member (in our case, IEFSSN00), so that the subsystem names are active after the next IPL, and then add them dynamically as shown in Figure 34 on page 84.

```
EDIT       SYS1.PARMLIB(IEFSSN04) - 01.08               Columns 00001 00072
****** ***************************** Top of Data ******************************
000100 JES2,,,PRIMARY               JOB ENTRY SUBSYSTEM
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 40 Line(s) not Displayed
001100 SUBSYS SUBNAME(MPM5)         /* ORACLE8i SUBSSTEM NAME */
001200 SUBSYS SUBNAME(TNS5)         /* ORACLE NET8 SUBSYSTEM NAME */
****** **************************** Bottom of Data ****************************
```

```
Display  Filter  View  Print  Options  Help
 -------------------------------------------------------------------------------
 SDSF SYSLOG  19794.101 SC04 SC04 05/30/1999 LINE  14,695     COLUMNS  51130
 COMMAND INPUT ===> /setssi add,s=MPM5                        SCROLL == => PAGE

 SETSSI ADD COMMAND for SUBSYSTEM MPM5 COMPLETED SUCCESSFULLY
```

```
Display  Filter  View  Print  Options  Help
 -------------------------------------------------------------------------------
 SDSF SYSLOG  19794.101 SC04 SC04 05/30/1999 LINE  14,695     COLUMNS  51130
 COMMAND INPUT ===> /setssi add,s=TNS5                        SCROLL == => PAGE

 SETSSI ADD COMMAND for SUBSYSTEM TNS5 COMPLETED SUCCESSFULLY
```

Figure 34. Add the Oracle subsystem names

C.2 APF-authorize the Oracle AUTHLOAD library

The next pre-installation step is to authorize the Oracle AUTHLOAD library. You first specify the dataset name and volume of the library, and then activate this authorization. You again have the choice of accomplishing this dynamically, as shown in Figure 35, or by IPLing your system.

Note: With Oracle8 and Oracle8*i*, the APF library must be placed on DFSMS-managed storage, so you must have at least one volume of SMS-managed storage. If you have more than one volume in your stogroup, you will not know which volume the AUTHLOAD file will reside on, so you can only do this step after you have run the job to create the AUTHLOAD file.

```
EDIT       SYS1.PARMLIB(PROG00) - 01.15                 Columns 00001 00072
****** ***************************** Top of Data ******************************
000100 APF FORMAT(DYNAMIC)
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 40 Line(s) not Displayed
003000 APF ADD DSNAME(MPM5.ORA815.AUTHLOAD)   VOLUME(ORACL8)
****** **************************** Bottom of Data ****************************
```

```
  Display  Filter  View  Print  Options  Help
 -------------------------------------------------------------------------------
 SDSF SYSLOG   176.101 ORACLE1   05/30/1999 LINE  14,695     COLUMNS  51 130
 COMMAND INPUT ===> /set prog=00                              SCROLL ===> PAGE
```

```
  Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF SYSLOG   176.101 ORACLE1   05/30/1999 LINE  15,025 COMMAND ISSUED
RESPONSE=ORACLE1   CSV410I APF FORMAT IS NOW DYNAMIC
```

Figure 35. APF-authorize an Oracle library dynamically

Make sure you have the dynamic APF-authorization set in your IEASYSxx PARMLIB member so that the Oracle library stays authorized after the next IPL. You do this with the entry PROG=xx in your IEASYSxx PARMLIB member.

C.2.1 High-level qualifiers for data set names

Note that we specified MPM5.ORA815.AUTHLOAD. We used MPM5 as the high-level qualifier for our data sets, with ORA815 as the second-level qualifier. References to MPM5.ORA815 are specific to our installation. For the VSAM files, we used MPM5.DBF as the high-level qualifier. You may choose different names. Names for the high-level qualifier should be one to four characters and begin with a letter.

A good naming strategy would put the Oracle executables under one high-level qualifier (hlq), and the database files (VSAM files) under a second hlq. The executables can be shared between instances and are release-sensitive. The database files are instance-specific. Considering this, you might put the executables in MPM5.ORA815.library (members) and put the data files in MPM5.DBF.SYSTEM (the data file name). The installation panels provide this flexibility.

C.3 Unload the Oracle installation JCL from the distribution tape

Loading the Oracle installation JCL from the distribution tape creates your installation library, reads one data set off the installation tape, and stores it as the first member in this library under the default name OSPIJA00. We used the JCL shown in Figure 36 on page 86 to retrieve the first member of the installation library. The expiration date for the tape was 98000, which we changed to 99365 to be able to read the tape. We were using an LPAR SC04 in a sysplex. We installed the installation library on disk ORACLK.

```
EDIT       ORACLE.TAPE.JCL(TAPEREAD) - 01.03               Columns 00001
Command ===>                                                 Scroll ===>
****** **************************** Top of Data ************************
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001 //ORATAPE  JOB (999,POK),'ORACLE TAPE READ',NOTIFY=&SYSUID,
000002 //    CLASS=A,MSGCLASS=T,TIME=1439,
000003 //    MSGLEVEL=(1,1)
000004 //***
000005 //***
000006 //***
000007 /*JOBPARM SYSAFF=SC04
000008 //S1     EXEC PGM=IEBCOPY
000009 //SYSPRINT DD SYSOUT=A
000010 //SYSUT1   DD UNIT=3490,VOL=SER=OS9908,LABEL=(1,SL,EXPDT=99365),
000011 //         DISP=OLD,DSN=OORX998.F1
000012 //SYSUT2   DD DSN=MPM5.ORA815.INSTLIB,
000013 //            UNIT=3390,VOL=SER=ORACLK,
000014 //            SPACE=(27920,(600,100,20)),
000015 //            DCB=(LRECL=80,RECFM=FB,BLKSIZE=27920),
000016 //            DISP=(NEW,CATLG,DELETE)
000017 //SYSIN    DD *
000018   COPY INDD=SYSUT1,OUTDD=SYSUT2
000019   SELECT MEMBER=(OSPIJA00)
000020 /*
```

Figure 36. JCL to load the first job from the tape

The jobparm card is specified because we were running in LPAR SC04 in a sysplex environment.

C.4 Create the ISPF libraries used by the installation

You have to edit the previously created sample job (member OSPIJA00 in MPM5.ORA815.INSTLIB) before you submit it for execution:

- The JOB card information probably will not conform to your installation.
- Choose an appropriate INDEX parameter; this will be the high-level qualifier of all Oracle data sets.
- Tape unit and volume serial number have to be set (TPUNIT, TPVOL); this is your installation input tape.
- DASD unit and volume serial number have to be set (PDASD, PDVOL); this is the volume on which Oracle allocates its data set.
- The device type for allocation of the temporary data set (TDASD) has to be specified.
- The DCBxxxx parameter points to your current ISPF libraries. Oracle will create its own ISPF libraries with the same DCB characteristics as the corresponding ISPF libraries on your system.

You can issue the following TSO command from the TSO command line to determine the correct allocations:

```
LISTALC STATUS
```

If you do not want to have all Oracle data sets in your master catalog, you should first do the following:

- Define a USERCATALOG for Oracle.

- Define an ALIAS (using your first or only part of the INDEX parameter) that points to the new catalog.

After editing, our OSPIJA00 member looked as shown in Figure 37:

```
EDIT       MPM5.ORA815.INSTLIB(OSPIJA00) - 01.00          Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 //MPM5CLIS JOB (0000,ORA),'ORAIPO INSTALLATION',
000002 //             REGION=1024K,NOTIFY=ORACLE1,
000003 //             CLASS=A,MSGCLASS=X
000004 //ORAISPF PROC INDEX='MPM5.ORA815',
000005 //             TPUNIT=3490,
000006 //             TPVOL=OS9908,
000007 //             PDASD=3390,
000008 //             PDVOL=ORACLK,
000009 //             TDASD=SYSDA,
000010 //             DCBPLIB='ISP.SISPPENU',
000011 //             DCBSLIB='ISP.SISPSENU',
000012 //             DCBMLIB='ISP.SISPMENU',
000013 //             DCBCLIB='ISP.SISPCLIB'
000014 //*
- - -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  -  - 97 Line(s) not Displayed
****** **************************** Bottom of Data ****************************
```

Figure 37. JCL to install the customization dialog

After this job finishes successfully, you can invoke the Oracle installation dialog. This job has created the installation dialog as a member ORIPO01 in MPM5.ORA815.ISPCLIB.

C.5 Execute ISPF and invoke the Oracle customization process

While the database can be installed with SMP/E, there is no real advantage to using SMP/E. Normal maintenance is not installed with SMP/E. Maintenance is generally installed by concatenating modules, replacing modules, or applying ZAPs (program fixes) provided by Oracle.

You invoke the installation dialog as shown in Figure 38 on page 88.

```
   Menu  List  Mode  Functions  Utilities  Help
 ------------------------------------------------------------------------------
                              ISPF Command Shell
 Enter TSO or Workstation commands below:

 ===> exec 'MPM5.ORA815.ISPCLIB(ORIPO01)'

 Please enter the first and second level data set name qualifiers
 for the ISPF data sets you have copied from the tape.

 Example:  If your ISPF Clist data set name is ORA1.ORAPROD.ISPCLIB,
 then you would enter  ORA1.ORAPROD  without quotes or an ending period.

MPM5.ORA815

 You entered MPM5.ORA815 as the data set prefix for your ISPF files.

 If this is correct enter C to continue, otherwise you will have
 to reenter this information:
C

 Is this going to be an SMP/E install?  Please answer Y or N.
N
```

Figure 38. The Oracle installation dialog

Enter `s` as shown in Figure 39 to select Oracle8 Server and add the high-level qualifiers for the target data sets.

```
OR@PRIM  --------- ORACLE PRODUCTS FOR OS/390 INSTALLATION --- Row 1 to 1 of 1
                      ORACLE PRODUCT INSTALL MENU           USERID   - ORACLE1
                                                            DATE     - 00/03/09
 OPTION ===>                SELECT PRODUCT SET              TIME     - 09:50


   Select only one product set to install, make your selection by placing
   any non-blank character next to it.

   You must enter a different first and second level qualifier(s)
   for each selected product set that appears on this panel.

        Product Set                  Target Dataset
        Description                  Name Qualifiers
 ------------------------------------------------------------------------------
   S  Oracle8 Server              MPM5.ORA815
 ******************************* Bottom of data ********************************
```

Figure 39. Selecting a product to install

When you run ORIPO01, you create an Oracle ISPF data set environment profile library, tso_userid.ORISPF.ISPFPROF. There are some interesting things kept here:

- A high-level qualifier for data sets.
- An indicator of whether this was an SMP/E install. If so, all other installs are forced to be the same.
- The volume IDs of the tapes that need to be mounted.
- The level of the database (for example, Version 8).

If you want to restart from the very beginning, you may have to modify or delete this file.

Eventually you arrive at the Oracle primary option menu, as shown in Figure 40. Select option **1, Define Primary ORACLE PRODUCTS Installation Parameters** on this menu.

```
OR@INST  --------- ORACLE PRODUCTS FOR OS/390 INSTALLATION ---------
OPTION  ===> 1  ORACLE PRIMARY OPTION MENU       USERID   - ORACLE1
                                                  DATE     - 00/03/09
                                                  TIME     - 09:54
     1  Primary      - Define Primary ORACLE PRODUCTS -     TERMINAL - 3278
                       Installation Parameters.
     2  Generate (P) - Generate Installation Job "********".

     3  Secondary    - Define Secondary ORACLE SERVER Installation
                       Parameters.
     4  Generate (S) - Generate Installation Job "********".

     5  Reset all Product and Language selections.

     X  EXIT         - Exit ORACLE Install Dialog facility.

                         (P) Primary   ORACLE PRODUCTS process.
                         (S) Secondary ORACLE SERVER Subsystem process.
 Enter END command to terminate ORACLE Install Dialog facility.
```

Figure 40. The Oracle Primary Option Menu

After selecting 1, the following series of screens show the entries we made for our system. We proceeded sequentially through the screens.

Primary install parameters

```
ORPRIM0  --------- ORACLE PRODUCTS FOR OS/390 INSTALLATION ---------
OPTION  ===>           DEFINE PRIMARY INSTALL PARAMETERS    USERID   - ORACLE1
                                                            DATE     - 00/03/09
Press ENTER to proceed sequentially.              TIME     - 10:01

      0  Enter SMP/E Environment variables.
      1  Modify tape unit and non-VSAM library index.
      2  Select Oracle product(s) to be installed.
      3  Select National Language Support module to be installed.
      4  Define Oracle subsystem name and VSAM library index.
      5  INSTLIB/ISPSLIB file tailoring information.
      6  Define JOB card for installation jobs.
      7  Define datasets for PROCs, CLISTs, and temporary disk unit.
      8  Review/modify space specifications for major libraries.
      9  Specify VOLSER for other libraries.
     10  Select ORACLE Server option.
     11  Define database/control/log files and their residence.
     12  Define Transparent Gateway for DB2 subsystem. (TG4DB2 install only)
     13. Define Oracle Transparenet Gateway for EDA/SQL subsystem
           (TG4EDA/SQL install only)
     14  Define USS products installation parameters.
           (USS products install only)
```

Specifying the non-VSAM file names

```
ORPTIP00 --------- ORACLE PRODUCTS FOR OS/390 INSTALLATION ---------
COMMAND ===> C MODIFY TAPE UNIT & LIBRARY INDEX       USERID  ORACLE1
                                                             DATE  - 00/03/09
Enter C to continue. Re-enter parameters to change.

 1  ORACLE DISTRIBUTION LIBRARY:

     TAPE UNIT NAME      ===> 3490                 User's Tape Generic Unit Name
     TAPE VOLUME SERIAL  ===> OS9908,OS9924,OS9940 ORACLE DLIB Tape Volume Label

 2  ORACLE NONVSAM LIBRARY INDEX: (AUTHLOAD, CMDLOAD, PARMLIB,
                                   SRCLIB, SQLLIB)

        HIGH   LEVEL QUALIFIER ===> MPM5
        SECOND LEVEL QUALIFIER ===> ORA815
```

Selecting the products to install

```
ORPRODS  --------- ORACLE PRODUCTS FOR OS/390 INSTALLATION --- Row 1 to 8 of 8
COMMAND ===> C   SELECT ORACLE PRODUCT(S) FOR INSTALLATION  SCROLL ===>
                                                            USERID - ORACLE1
Enter S  to select a product, otherwise leave it blank.     DATE   - 00/03/09
Enter C  to continue. Press END to return the previous panel. TIME   - 10:06



         Select  Product
        s      Oracle8 Server
         -       Access Manager for CICS
         -       Access Manager for IMS/TM
        s      Oracle Precompilers
        s      Net8
        s      SQL*Plus and Help Files
        s      Oracle Ent. Mgr Agent (USS product)
        s      SQL*Plus (USS product)
        s      USS Utilities (Wrapper, NLS, JAVA)
```

Selecting the language

You will be presented with a screen listing all the supported languages. English(US) is the default, so enter `C` to continue and press Enter and no selection below.

Specifying the VSAM file nNames

Note: You must fill in the second-level qualifier for the VSAM files even though the dialog will not force you to do so. We specified to have DBF as the second-level qualifier. The asterisk (*) is the character that you will use as the shortcut for operator commands.

```
ORPTIP05 --------- ORACLE PRODUCTS FOR OS/390 INSTALLATION ---------
COMMAND ===> C  DEFINE SUBSYSTEM AND VSAM LIBRARY INDEX   USERID   - ORACLE1
                                                            DATE     - 00/03/09
Enter C to continue.

 1  ORACLE SERVER SUBSYSTEM NAME:

       SUBSYSTEM NAME      ===>  MPM5       ORACLE Subsystem Name
       PREFIX CHARACTER    ===>  *          Operator Communication         Prefix
                                            Character

 2  ORACLE VSAM LIBRARY INDEX: (DB, CONTROL, and REDOLOG files)

       HIGH   LEVEL QUALIFIER ===> MPM5
       SECOND LEVEL QUALIFIER ===> DBF
```

Verifying the INSTLIB name

```
ORPTIP15 --------- ORACLE PRODUCTS FOR OS/390 INSTALLATION ---------
COMMAND ===>C   INSTLIB/ISPSLIB FILE TAILORING INFORMATION  USERID - ORACLE1
                                                              DATE   - 00/03/09
Enter C to continue. Re-enter parameters to change.

 1  ORACLE ISPSLIB (SKELETON) LIBRARY MEMBER:

      SKELETON LIBRARY MEMBER ===> NEWSKEL  Untailored Input JCL Configuration
                                            Skeleton Member Name
 2  ORACLE INSTALLATION LIBRARY: (INSTLIB)

       INSTLIB DATASET NAME ===> MPM5.ORA815.INSTLIB
       INSTLIB MEMBER NAME  ===> ORPIJA01   Tailored JCL Configuration Member
       VOLUME SERIAL        ===>            (If Not Cataloged)
       DEVICE TYPE          ===>

       DATASET DISPOSITION  ===> SHR        (NEW or SHR) DISP=SHR

       REPLACE LIKE-NAMED INSTLIB MEMBER   ===> NO       (YES or NO)
```

Specifying the model job card

```
ORPTIP20  --------- ORACLE PRODUCTS FOR OS/390 INSTALLATION ---------
 COMMAND ===> C  DEFINE JOB CARDS FOR INSTALL JOBS   USERID- ORACLE1
                                                       DATE     - 00/03/09
 Enter C to continue. Re-enter parameters to change.

  1  DEFINE ORACLE INSTALLATION JOB STATEMENT INFORMATION:


  ===> //MPMINS1     JOB (0000,OR),'ORACLE INSTALL',CLASS=A,
  ===> //           MSGCLASS=X,PRTY=15,MSGLEVEL=(1,1),NOTIFY=ORACLE1
  ===> //*
  ===> //*
  ===> //*
  ===>
  ===>
  ===>
```

Specifying the proclib to use

If you are creating several Oracle instances in the same LPAR and you are using the same SYS1.PROCLIB, you must change the ORACLE JCL PROCEDURE SUFFIX to a unique identifier or you will overwrite other procedures used by other instances.

```
ORPTIP25 --------- ORACLE PRODUCTS FOR OS/390 INSTALLATION ---------
 COMMAND ===> C  DEFINE PROCS, CLISTS, & TEMPORARY SPACE  USERID   - ORACLE1
                                                          DATE     - 00/03/09
 Enter C to continue. Re-enter parameters to change.

  1   DEFINE ORACLE PROCEDURE AND CLIST TARGET LIBRARIES:

        PROCLIB LIBRARY   ===> SYS1.PROCLIB

        ORACLE JCL PROCEDURE SUFFIX  ===> 8i

        TSO CLIST LIBRARY   ===> MPM5.ORA815.ISPCLIB

  2  ORACLE TEMPORARY DISK SPACE UNIT NAME:

        GENERIC UNIT NAME FOR DISK WORK SPACE ===> SYSDA
```

Specifying the VOLSER and UNIT entries

The AUTHLOAD file must be on SMS-managed DASD. Our storage class was SCORAC. We chose not to use SMS-managed DASD for the other files, so we had to specify the VOLSER and the UNIT values.

```
ORDSN  --------- ORACLE PRODUCTS FOR OS/390 INSTALLATION ------- Row 1 from 20
COMMAND ===> C    REVIEW/MODIFY SPACE SPECS - MAJOR LIBS    SCROLL ===> PAGE

COMMAND: Enter C  to continue
Type over the input fields to change the dataset attributes

----------------------------------------------------------------------------
DSNAME     VOLSER  UNIT                       SPACE
                          ---------------------------------------------------
                          TYPE  LRECL RECFM  PRI     SEC     BLKSIZE   DIRBLK
-----------------------------------------------------------------------------
 AUTHLOAD                 TRK    0      U    4118     810      27998       0
         MGMTCLAS:              STORCLAS:  SCORAC     DATACLAS:
 CMDLOAD  ORACLK  3390      TRK    0    U    4810     587      27998     432
         MGMTCLAS:              STORCLAS:             DATACLAS:
 PARMLIB  ORACLK  3390    TRK    256    VB   1          1       4096       2
         MGMTCLAS:              STORCLAS:             DATACLAS:
 SQL      ORACLK  3390    TRK    256    VB   366      73        4096      68
         MGMTCLAS:              STORCLAS:             DATACLAS:
 SRCLIB   ORACLK  3390    TRK    80     FB   95      20       27920     22
```

Specifying the volume for other libraries

Note that we have used only one volume for all our files as it was a test system. For a production system, you would want to spread these over several volumes.

```
ORDSNO  --------- ORACLE PRODUCTS FOR OS/390 INSTALLATION ---------
COMMAND ===> C   SPECIFY VOLSER FOR OTHER LIBRARIES         SCROLL ===> PAGE
                                                              USERID - ORACLE1
Enter C  to continue                                          DATE=  - 99/05/0
Type over on input fields to change


        VOLUME SERIAL       ===> ORACLA     Resident Library Volume Name

        DEVICE TYPE         ===> 3390       Device Type


        MGMTCLAS:              STORCLAS:              DATACLAS:
```

Specifying other options

We were not planning to use these options so we specified `n`.

```
ORPTIP30 --------- ORACLE PRODUCTS FOR OS/390 INSTALLATION ---------
 COMMAND ===> C          SELECT ORACLE SERVER OPTIONS      USERID   - ORACLE1
                                                             DATE     - 00/03/09
 Enter C to continue. Re-enter parameters to change.




 1 PARTITIONING OPTION           ====> n      Enter 'y' for partitioning option
                                              else 'n'


 Enter END command to return.
```

Specifying the database file information

We used the defaults at this time, planning to increase the size later when we installed Oracle Applications.

```
ORPTIP35  --------- ORACLE PRODUCTS FOR OS/390 INSTALLATION -- Row 1 to 2 of 2
COMMAND ===> C      DEFINE DATABASE FILE INFORMATION      SCROLL ==  => PAGE

Enter C to continue.                                        USERID   - ORACLE1
Enter A to add an additional database.                      DATE     - 00/03/09
      Ax where x is number of DB files to add.
Enter D beside any row to be deleted.

   ---DB residence---
  Primary  Optional  Device     Primary  Secondary     Units
  volume   volume    type       space    space         TRK/BLK/CYL
  ORACLK             3390       18000    0               BLK
           Tablespace           Dataset
           SYSTEM           MPM5.DBF.SYSTEM.DB1

  ORACLK             3390       4000     0               BLK
           Tablespace           Dataset
           DRSYS            MPM5.DBF.ROLLBACK.DB2

  ORACLK             3390       4000     0               BLK
           Tablespace           Dataset
           USER3            MPM5.DBF.USER3.DB3

****************************** Bottom of data ********************************
```

Specifying the control files information

```
ORPTIP40 --------- ORACLE PRODUCTS FOR OS/390 INSTALLATION --- Row 1 to 3 of 3
 COMMAND ===> C      DEFINE ORACLE CONTROL FILE INFORMATION  SCROLL ==  => PAGE

 Enter C to continue.                                        USERID   - ORACLE1
 Enter A to add an additional control file.                  DATE     - 00/03/09
       Ax where x is number of files to add.
 Enter D beside any row to be deleted.

 Please note that you should have one snapshot control file and at least
 2 control files and that each file should reside on a unique volume.

     Residence   Primary    Secondary
      volume     space      space        Units
       name     (in Units)  (in Units)
      ORACL6       2000         500      BLK
                Control File name  MPM5.DBF.CONTROL2
      ORACL7       2000         500      BLK
                Control File name  MPM5.DBF.CONTROL1
      ORACL6       2000         500      BLK
                Control File name  MPM5.DBF.SNAPCF
```

Specifying the REDO LOG information

```
ORPTIP45 --------- ORACLE PRODUCTS FOR OS/390 INSTALLATION --- Row 1 to 2 of 2
COMMAND ===> C      DEFINE ORACLE REDO LOG INFORMATION      SCROLL ==  => PAGE

Enter C to continue.                                        USERID   - ORACLE1
Enter A to add an additional REDO LOG file.                 DATE     - 00/03/09
      Ax where x is number of files to add.
Enter D beside any row to be deleted.


    Residence
     volume    Device     Primary  Secondary    Units
      name      type      space    space(BLK/TRK/CYL)
     ORACL6      3390       4000         0      BLK
               REDO LOG file name  MPM5.DBF.LOG2
     ORACL7      3390       4000         0      BLK
               REDO LOG file name  MPM5.DBF.LOG1
****************************** Bottom of data ********************************
```

For a production environment, you should ensure that you have three log files and that they do not reside on the same physical volume. You can add an additional REDO LOG file by using the A option.

There is a new panel with Oracle8.1.5 that asks you to enter the UNIX Systems Services (USS) directories that Oracle will use. We added the home directory for Oracle and used the default /tmp directory for temporary files.

```
T ORPTIP50 --------- ORACLE PRODUCTS FOR OS/390 INSTALLATION ---------
 COMMAND ===> C          DEFINE USS PRODUCTS             USERID   - ORACLE2
                           INSTALLATION PARAMETERS          DATE     - 00/03/09
 Enter C to continue.

  1  Unix System Services directory path for Oracle Home:

        ===> /home/oracle

  2  Unix System Services directory path for temporary files:
      Note: This directory should exist prior to completing the
            USS Product installation.

        ===>  /tmp
```

Completing the product definition process

At this point you can press PF4 to return to the Oracle Primary Menu Option, unless you wish to review or change your choices.

```
ORPTIP60 --------- ORACLE PRODUCTS FOR OS/390 INSTALLATION ---------
COMMAND ===>                                                 USERID   - ORACLE1
                                                             DATE     - 99/05/0


        This completes the Primary ORACLE products definition
        process.


        Enter 'PF4' to return to ORACLE Primary Option Menu

        Enter 'PF3' to backup and review the selections made




Enter END command to return to previous menu.
```

C.6 Generate and run the customization job

After you complete the installation parameter settings and have returned to the Oracle primary option menu, specify option **2, Generate Installation Job**, as shown in Figure 41.

```
OR@INST  --------- ORACLE PRODUCTS FOR OS/390 INSTALLATION ---------
OPTION  ===> 2 ORACLE PRIMARY OPTION MENU       USERID   - ORACLE1
                                                DATE     - 00/03/09
                                                TIME     - 09:54
     1  Primary      - Define Primary ORACLE PRODUCTS -     TERMINAL - 3278
                       Installation Parameters.
     2  Generate (P) - Generate Installation Job "********".

     3  Secondary    - Define Secondary ORACLE SERVER Installation
                       Parameters.
     4  Generate (S) - Generate Installation Job "********".

     5  Reset all Product and Language selections.

     X  EXIT         - Exit ORACLE Install Dialog facility.

                       (P) Primary   ORACLE PRODUCTS process.
                       (S) Secondary ORACLE SERVER Subsystem process.
 Enter END command to terminate ORACLE Install Dialog facility.
```

Figure 41. Generate installation job

This process puts member ORPIJA01 into your installation library.

You are required to run this customization job, as it will put a series of installation and verification jobs into your installation library. In addition, it creates the members that will be copied to the Oracle parameter library by the installation.

At this point, enter X, as shown in Figure 42 on page 98, to exit the Oracle installation dialogue.

```
OR@INST  --------- ORACLE PRODUCTS FOR OS/390 INSTALLATION ---------
OPTION  ===>  x            ORACLE PRIMARY OPTION MENU       USERID   - ORACLE1
                                                            DATE     - 00/03/09
                                                            TIME     - 09:54
     1  Primary      - Define Primary ORACLE PRODUCTS -     TERMINAL - 3278
                       Installation Parameters.
     2  Generate (P) - Generate Installation Job "********".

     3  Secondary    - Define Secondary ORACLE SERVER Installation
                       Parameters.
     4  Generate (S) - Generate Installation Job "********".

     5  Reset all Product and Language selections.

     X  EXIT         - Exit ORACLE Install Dialog facility.

                       (P) Primary   ORACLE PRODUCTS process.
                       (S) Secondary ORACLE SERVER Subsystem process.
 Enter END command to terminate ORACLE Install Dialog facility.
```

Figure 42. Ending the Oracle installation dialogue

C.7 Run the Generated installation jobs

This section lists the series of jobs we had to submit. This list of jobs will vary based on the install options you select.

Submit the customization job that is in MPM5.ORA815.INSTLIB:

ORPIJA01 Creates the jobs in MPM5.ORA815.INSTLIB.

Submit the following series of installation jobs:

ORPIJD00 Allocates and loads runtime libraries from tape.
(We had to change the expdate on the tape from 98000 to 99365 to be able to execute this job, and we also had to change the job card to be able to execute the jobs.)
ORPIJE00 Allocates database, control, and redo log clusters.
ORPIJF01 Creates MPM parameters and Oracle proclib members.
ORPIJF05 Copies procedure ORASQLV7 to your proclib.
ORPIJF06 Creates Oracle precompiler procedures.
ORPIJF07 Copies Oracle Context Cartridge.
ORPIJF08 Copies Oracle Reports.

Note: You only need to run ORPIJF06 if you selected the precompilers on the product selection screen ORPRODS.

C.8 Edit the PARMLIB members and startup procedure

The initial PARMLIB members are shown in Figure 43 through Figure 46 on page 101.

```
EDIT       MPM5.ORA815.PARMLIB(CREATE) - 01.00            Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 SET ECHO ON
000002 CONNECT INTERNAL
000003 STARTUP PFILE=/DD/INITORA NOMOUNT
000004 CREATE DATABASE MPM5 MAXDATAFILES 256
000005 LOGFILE '/DSN/MPM5.DBF.LOG2'   ,
000006 '/DSN/MPM5.DBF.LOG1'
000007 DATAFILE '/DSN/MPM5.DBF.SYSTEM.DB1';
000008 CREATE TABLESPACE ROLLBACK DATAFILE '/DSN/MPM5.DBF.ROLLBACK.DB2';
000009 CREATE TABLESPACE USER3 DATAFILE '/DSN/MPM5.DBF.USER3.DB3';
000010 CREATE ROLLBACK SEGMENT S1 TABLESPACE SYSTEM;
000011 CREATE ROLLBACK SEGMENT S2 TABLESPACE SYSTEM;
000012 CREATE ROLLBACK SEGMENT S3 TABLESPACE SYSTEM;
000013 CREATE ROLLBACK SEGMENT S4 TABLESPACE SYSTEM;
000014 ALTER ROLLBACK SEGMENT S1 ONLINE;
000015 ALTER ROLLBACK SEGMENT S2 ONLINE;
000016 ALTER ROLLBACK SEGMENT S3 ONLINE;
000017 ALTER ROLLBACK SEGMENT S4 ONLINE;
```

Figure 43. Oracle PARMLIB: initial database build options

The default settings are shown in Figure 43. For example, later on we found we needed one large rollback segment that we had to create. Any large rollback segment should have its own tablespace. MAXDATAFILES determines the maximum number of datafiles in your database. You may want to consider increasing the number from 256 to 4096. At this point, you can continue with the defaults, as shown in Figure 44 on page 100.

```
EDIT       MPM5.ORA815.PARMLIB(INITORA) - 01.00            Columns 0000
Command ===>                                                 Scroll ==
****** ***************************** Top of Data **********************
000001 CONTROL_FILES = "/DSN/MPM5.DBF.CONTROL2"
000002 CONTROL_FILES = "/DSN/MPM5.DBF.CONTROL1"
000003 SHARED_POOL_SIZE = 20000000
000004 DB_BLOCK_BUFFERS = 500
000005 DB_BLOCK_MAX_DIRTY_TARGET = 0
000006 DB_FILES = 256
000007 DB_NAME = MPM5
000008 LOG_BUFFER = 65536
000009 LOG_CHECKPOINT_INTERVAL = 3000
000010 OPEN_CURSORS = 120
000011 ROLLBACK_SEGMENTS = (S1,S2,S3,S4)
000012 TRANSACTIONS = 55
000013 SESSIONS = 55
000014 PROCESSES = 50
000015 DML_LOCKS = 220
000016 COMPATIBLE = 8.1.0
****** *************************** Bottom of Data ********************
```

Figure 44. Oracle PARMLIB: database instance initialization

We added a second privileged user ID, ORACLE2, as shown in Figure 45, so that two users can execute server manager privileged commands such as `STARTUP`, `SHUTDOWN` or `CONNECT INTERNAL`. This includes `SVRMGRL` in both cmdload and in mpmcmd(console commands).

```
EDIT       MPM5.ORA815.PARMLIB(MPMPARM) - 01.00            Columns 00001
Command ===>                                                 Scroll ===
****** ***************************** Top of Data **********************
000001 LANGUAGE='AMERICAN_AMERICA.WE8EBCDIC1047'
000002 SSNAME='MPM5'
000003 COMCHAR='*'
000004 USERS=50
000005 CONBUF=13
000006 # Customers should tune Stack parameters appropriately for their
000007 MAXSTACKBUF=7
000008 MAXINCREMENT=4
000009 STACK_INIT_SIZE=262144
000010 STACK_INCR_SIZE=32768
000011 #
000012 TRACESIZE=32767
000013 TRACEDS="MPM5.ORA815.TRACE** "
000014 PRIVUSER=ORACLE1
000015 PRIVUSER=ORACLE2
000016 SSCOMMENT='VER(VERSION8)'
000017 UPPERCASE
****** *************************** Bottom of Data ********************
```

Figure 45. Oracle PARMLIB: MPM initialization

Our startup procedure initially looked as shown in Figure 46 on page 101.

```
EDIT       MPM5.ORA815.PARMLIB(STARTUP) - 01.02            Columns 00001 00072
****** ***************************** Top of Data ****************************
000001 SET ECHO ON
000002 CONNECT INTERNAL
000003 STARTUP PFILE=/DD/INITORA NOMOUNT
000004 ALTER DATABASE MPM5 MOUNT;
000005 ALTER DATABASE OPEN;
****** **************************** Bottom of Data ***************************
```

Figure 46. Oracle PARMLIB: normal startup

Note: The parameter Region=0M on line 000022 of the startup procedure, shown in Figure 47 on page 102, must be set as shown so that Oracle can get the amount of virtual memory it needs; review the *Oracle System Administration Guide*, Section 8 on memory requirements.

```
EDIT       SYS1.PROCLIB(ORAMPM5) - 01.00                    Columns 0000100072
Command ===>                                                 Scroll ===> PAGE
****** ***************************** Top of Data ******************************
000001 //ORAMPM5  PROC SYSOUT='SYSOUT=X', JES SYSOUT CLASS.
000002 //          INDEX='MPM5',
000003 //*                           NONVSAM LIBRARY HLINDEX.
000004 //          LIBV='ORA815',
000005 //*                           ORACLE/INST VERSION.
000006 //          START=STARTUP     USE SVRMGRL TO START AN INSTANCE.
000007 //*
000008 //*-------------------------------------------------------------------*
000009 //*              ORACLE CORPORATION                                   *
000010 //*              CORPORATE PRODUCT SUPPORT                            *
000011 //*              500 ORACLE PARKWAY                                   *
000012 //*              REDWOOD SHORES CALIFORNIA 94065 U.S.A                *
000013 //*                                                                   *
000014 //*                                 ATTENTION: MVS PRODUCT SUPPORT    *
000015 //*                                            650/506-7000           *
000016 //*  ORACLE SERVER LICENSED MATERIALS - PROPERTY OF ORACLE CORP.       *
000017 //*  CONTAINS RESTRICTED MATERIALS OF ORACLE CORP.                    *
000018 //*  REFER TO COPYRIGHT INSTRUCTIONS DOCUMENT NUMBER XXXX-XXXX.       *
000019 //*  (C) COPYRIGHT ORACLE CORP. 1986                                  *
000020 //*-------------------------------------------------------------------*
000021 //*
000022 //IEFPROC  EXEC PGM=ORACLE,PARM='>/OPER/',REGION=0M,
000023 //             DYNAMNBR=40,
000024 //             TIME=1440
000025 //STEPLIB  DD DSN=&INDEX..&LIBV..AUTHLOAD,
000026 //            DISP=SHR
000027 //SYSOUT   DD &SYSOUT,DCB=(LRECL=133,BLKSIZE=137,RECFM=VB,BUFNO=1)
000028 //SYSERR   DD SYSOUT=*,DCB=(LRECL=133,BLKSIZE=137,RECFM=VB,BUFNO=1)
000029 //ORA$LIB  DD DSN=&INDEX..&LIBV..AUTHLOAD,
000030 //            DISP=SHR
000031 //SQLBSQ   DD DSN=&INDEX..&LIBV..SQL(SQLBSQ),
000032 //            DISP=SHR
000033 //SYSPRINT DD DSN=&INDEX..&LIBV..MPM5ALRT,DISP=SHR
000034 //INITORA  DD DSN=&INDEX..&LIBV..PARMLIB(INITORA),
000035 //            DISP=SHR
000036 //STARTUP  DD DSN=&INDEX..&LIBV..PARMLIB(&START),
000037 //            DISP=SHR
000038 //SYSIN    DD DSN=&INDEX..&LIBV..PARMLIB(MPMPARM),
000039 //            DISP=SHR
000040 //         DD DSN=&INDEX..&LIBV..PARMLIB(SQLDBA),
000041 //            DISP=SHR
000042 //SNAPCF   DD DSN=MPM5.DBF.SNAPCF,
000043 //            DISP=SHR
000044 //*    ORA$VSAM used by RMAN for dynamic allocation of DB files
000045 //ORA$VSAM DD DSN=MPM5.DBF.SNAPCF,
000046 //            DISP=SHR
000047 //*
000048 //SQLNET   DD DUMMY,DCB=BLKSIZE=20
000049 //SQLNETTC DD DUMMY,DCB=BLKSIZE=20
000050 //SQLNETLG DD DUMMY,DCB=BLKSIZE=20
000051 //INTCHG   DD DUMMY,DCB=BLKSIZE=20
000052 //TNSNAV   DD DUMMY,DCB=BLKSIZE=20
****** ************************** Bottom of Data ***************************
```

Figure 47. Oracle startup procedure

C.9 Initialize the Oracle database

Now you are ready to initialize the Oracle database instance. You can enter the following command on the system console or under SDSF:

```
S ORAMPM5.MPM5,START=CREATE
```

Alternatively, you can run the generated job ORPIJG00 from your installation library. As the Oracle database instance starts you will see the messages, shown in Figure 48 on page 104, on the console:

```
 COMMAND INPUT ===> /s ORAMPM5.MPM5,start= create            SCROLL === > PAGE
        IEF403I ORAMPM5 - STARTED - TIME=14.25.41            ASID=008C
MPM MPM002I MPM INITIALIZING
*MPM517I ************************************************************
*MPM558I * MPM VERSION      1.1.09.00.00
*MPM559I * MPM LINK EDIT DATE:09/28/99 TIME:09.50
*MPM517I ************************************************************
*MPM085I 50 USERS, EACH 266K PRIVATE AREA (31K PGA), TOTAL OF 15023K
BYTES
*MPM151I LXRES: PC NUMBER 27648 (X00006C00) ASSIGNED to MPM5)
*MPM523I MPM BUILT ESA CROSS MEMORY ENTRY TABLE
*MPM160I ORACLE MPM SUBSYSTEM MPM5 IS READY TO ACCEPT REQUESTS.
*MPM193I MPM5 SMFRECN=199  SMF RECORDING IS ACTIVE
*MPM008I OK.
*MPM008I OK.
*MPM008I OK.
*MPM008I OK.
*MPM008I OK.
*MPM025I TASK MANAGER COMING UP.
*MPM099I ATTACHED TCB AT X7DB3F0 FOR TASK 'SVRMGRL'
*MPM099I ATTACHED TCB AT X7DB258 FOR TASK 'TASK0001'
*MPM099I ATTACHED TCB AT X7D6E88 FOR TASK 'TASK0002'
*MPM099I ATTACHED TCB AT X7D6CF0 FOR TASK 'TASK0003'
*MPM099I ATTACHED TCB AT X7D6B58 FOR TASK 'TASK0004'

Oracle Server Manager Release 3.1.5.0.0 - Production

(c) Copyright 1997, Oracle Corporation.  All Rights Reserved.

Oracle8 Enterprise Edition Release 8.1.5.0.0 - Production
PL/SQL Release 8.1.5.0.0 - Production

Echo                           ON
SVRMGR> CONNECT INTERNAL
Connected.
SVRMGR> STARTUP PFILE=/DD/INITORA NOMOUNT
*MPM099I ATTACHED TCB AT X7BF688 FOR TASK 'PMON'
*MPM099I ATTACHED TCB AT X7BF330 FOR TASK 'DBW0'
*MPM099I ATTACHED TCB AT X7BF198 FOR TASK 'LGWR'
*MPM099I ATTACHED TCB AT X7D6938 FOR TASK 'CKPT'
*MPM099I ATTACHED TCB AT X7D6668 FOR TASK 'SMON'
*MPM099I ATTACHED TCB AT X7D64D0 FOR TASK 'RECO'
ORACLE instance started.
Total System Global Area      23407072 bytes
Fixed Size                       76256 bytes
Variable Size                 21209088 bytes
Database Buffers               2048000 bytes
Redo Buffers                     73728 bytes
SVRMGR> CREATE DATABASE MPM5 MAXDATAFILES 256
2> LOGFILE '/DSN/MPM5.DBF.LOG2'   ,
3>         '/DSN/MPM5.DBF.LOG1'
4> DATAFILE '/DSN/MPM5.DBF.SYSTEM.DB1';
```

Figure 48. Oracle database creation messages

It will take several minutes for the datafiles to be created, as the Oracle CCF utility takes time to format the VSAM files for use by the Oracle database. The message `Server Manager complete` indicates that the instance has been started successfully.

C.10 Run the remaining generated installation jobs

Submit the rest of your installation jobs:

ORPIJH00 initializes the Oracle database dictionary.
ORPIJI00 initializes the help tables.
ORPIJJ00 creates user and demonstration tables.

Note: Be sure to check the output of these jobs. As the listings are long, you could search for the strings `rror` and `arning` as this will find all instances of error, Error, warning and Warning.

C.11 Run the installation verification programs

Verify that your installation was successful by running the following programs:

ORIVJA01 Database engine install verification
ORIVJA04 SQL*Plus install verification
ORIVJA03 PreCompilers

C.12 Oracle files after installation

The file names shown in Figure 49 are the non-VSAM files we had on our DASD after the installation was complete:

```
MPM5.ORA815.AUTHLOAD
MPM5.ORA815.CMDLOAD
MPM5.ORA815.DOC
MPM5.ORA815.INSTLIB
MPM5.ORA815.ISPCLIB
MPM5.ORA815.ISPMLIB
MPM5.ORA815.ISPPLIB
MPM5.ORA815.ISPSLIB
MPM5.ORA815.ISPTLIB
MPM5.ORA815.IVPLOG
MPM5.ORA815.IVPOUT
MPM5.ORA815.MACLIB
MPM5.ORA815.MPM5ALRT
MPM5.ORA815.MSG
MPM5.ORA815.NET8.SAMPLIB
MPM5.ORA815.PARMLIB
MPM5.ORA815.REPORTS.PRT
MPM5.ORA815.REPORTS.RES
MPM5.ORA815.REPORTS.REX
MPM5.ORA815.SQL
MPM5.ORA815.SQLHELP
MPM5.ORA815.SQLLIB
MPM5.ORA815.SRCLIB
MPM5.ORA815.USSTAR
```

Figure 49. Non-VSAM files on our DASD at install completion (Part 1 of 2)

The file names shown in Figure 50 on page 106 are the VSAM files we had on our DASD.

```
MPM5.DBF.CONTEXT.DB2.DATA
MPM5.DBF.CONTROL1
MPM5.DBF.CONTROL1.DATA
MPM5.DBF.CONTROL2
MPM5.DBF.CONTROL2.DATA
MPM5.DBF.LOG1
MPM5.DBF.LOG1.DATA
MPM5.DBF.LOG2
MPM5.DBF.LOG2.DATA
MPM5.DBF.ROLLBACK.DB2
MPM5.DBF.SNAPCF
MPM5.DBF.SNAPCF.DATA
MPM5.DBF.SYSTEM.DB1
MPM5.DBF.SYSTEM.DB1.DATA
MPM5.DBF.USER3.DB3
MPM5.DBF.USER3.DB3.DATA
```

Figure 50. VSAM files on our DASD at install completion (Part 2 of 2)

C.13 Using archive log mode for the production database

We recommend that you use archive log mode when running an Oracle database in production. The steps to implement this are:

- Add the following parameters to the INITORA member:
  - LOG_ARCHIVE_DEST='LOC=DASD UNIT=3390"
  - LOG_ARCHIVE_FORMAT='/DSN/MPM5.ORA815.S%S.LOG"

    %S is an Oracle symbolic parameter.
- Add the following parameter to the START member:
  - LOG_ARCHIVE_START=TRUE
- Verify using the following SVRMGRL command:
  - archive log list

    This should show automatic archive as ENABLED.

Appendix D. Configuring Net8

We configured Net8 by following the instructions given in Chapter 9 of *Oracle8i Enterprise Edition for OS/390 Installation Guide*. We chose TCP/IP for our network software. You should be sure that your network is functioning, because Net8 will not check or debug the underlying network. From OS/390 you can verify your network connections by using `ping` from OS/390 to AIX, and vice versa.

The default port number (as shipped in the supplied Oracle server sample definitions) is 1521. You should verify that no other application or service is using this port number on your system. If this port number is in use or if you have more than one Oracle instance, choose another port for your Oracle8 Server.

You can include the following line in your TCP/IP profile data set or member in your port definitions:

```
1521 TCP                        ; Oracle
1522 TCP                        ; Oracle Second Instance
```

Note: If you add TNS after the TCP field, you must start TNS with this command:

```
start oratns.tns5
```

We recommend that, with the latest version of TCP/IP, you do not include the subsystem name in the TCP/IP profile dataset.

These entries are required for Net8 to function and to allow clients to log in to the database. However, the ports listed in the MPMTNS member (listener file) and the TNSNAMES.ORA file in the client must be the same. Therefore, it is good practice and helpful for other TCP/IP-related work to add the listener port to your TCP/IP profile data set or member in your port definitions.

D.1 Configuring the TNS subsystem

We edited member TNSPROC from the Oracle NET8.SAMPLIB and copied it under the name ORATNS5 to our PROCLIB, as shown in Figure 51.

```
EDIT       SYS1.PROCLIB(ORATNS5) - 01.05                 Columns 00001 00072
 ****** ***************************** Top of Data ******************************
 //ORATNS5 PROC INDEX=MPM5,
 //             LIBV=ORA815,
 //             P='CCHAR=@ SSN=TNS5 HPNS TCPSTK=TCPIPOE,
 //             R=0M,
 //             SYSOUT='SYSOUT=X'
 //IEFPROC  EXEC PGM=TNSMAIN,REGION=&R,TIME=1440,
 //  PARM='&P'
 //STEPLIB  DD DSN=&INDEX..&LIBV..AUTHLOAD,DISP=SHR
 //ORA$ENV  DD DSN=&INDEX..&LIBV..NET8.SAMPLIB(TNSENV),DISP=SHR
 //SYSTCPD  DD DSN=TCPIPOE.SC04.TCPPARMS(TCPDATA),DISP=SHR
 //SYSOUT   DD &SYSOUT,DCB=(LRECL=133,BLKSIZE=137,RECFM=VB,BUFNO=1)
 //SYSERR   DD SYSOUT=*,DCB=(LRECL=133,BLKSIZE=137,RECFM=VB,BUFNO=1)
 //SYSPRINT DD SYSOUT=*
 //SYSIN    DD DUMMY
 ****** **************************** Bottom of Data ************************
```

Figure 51. JCL to start the TNS subsystem

The member in your Oracle NET8.SAMPLIB data set will only contain the communication character definition:

```
P='CCHAR=@, ... '
```

Be sure that you change the P parameter to the following, where *name* is the name of the started task for TCP/IP, in our case TCPIPOE. This is new with OS/390 Release 2.5:

```
P='CCHAR=@ SSN=TNS5 HPNS TCPSTK=name',
```

We authorized the AUTHLOAD library and specified the subsystem name earlier; see “APF-authorize the Oracle AUTHLOAD library” on page 84.

Because we were not using the default stack name of TCPIP, we had to add the following DD card to enable Net8 to resolve the host name:

```
//SYSTCPD  DD DSN=TCPIPOE.SC04.TCPPARMS(TCPDATA),DISP=SHR
```

Our TNS environment contained only one line, as shown in Figure 52.

```
EDIT       MPM5.ORA815.NET8.SAMPLIB(TNSENV) - 01.00        Columns 00001 00072
 ****** ***************************** Top of Data ******************************
 000001 LANGUAGE='AMERICAN_AMERICA.WE8EBCDIC1047'
 ****** **************************** Bottom of Data ****************************
```

Figure 52. TNSENV file

D.1.1 Previous versions of TCP/IP

Prior to OS/390 Release 2.5, the setup of TCP/IP for TNS required other changes.

A HOSTNAME DD statement was included in the startup procedure pointing to the name of your TCP/IP host. Installations with multiple TCP/IP protocol stacks will require this additional DD statement for each stack.

The HOSTNAME information can be found in one of two places:

1. In the member pointed to by the DD SYSTCPD statement in the start proc for the TCP/IP started task (in our case, this was TCPIPMVS.SC04.TCPIP.DATA).
2. Or in the IEFSSNxx member in the PARMLIB (see Figure 34 on page 84).

Prior to OS/390 Release 2.5, the entry in the IEFSSNxx member would look more like VMCF.MVPXSSI,WTSC04 where VMCF is the stack name, MVPXSSI is the initialization routine, and WTSC04 is the HOSTNAME.

By default, Net8 will use your normal HOSTS.SITEINFO data set. If need be, you can use a TCPHOSTS DD statement.

D.2 Configuring Net8 for OS/390 servers

Add the following MPMTNS DD statement to your Oracle start procedure ORAMPM5 (see Figure 47 on page 102):

```
//MPMTNS    DD DISP=SHR,DSN=MPM5.ORA815.PARMLIB(MPMTNS)
```

We had to copy the member MPM5.ORA815.NET8.SAMPLIB(MPMORAT) to MPM5.ORA815.PARMLIB(MPMTNS) and make the changes. The MPMTNS DD statement refers to our TNS connect descriptor, shown in Figure 53:

```
EDIT       MPM5.ORA815.PARMLIB(MPMTNS) - 01.03            Columns 00001 00072
****** ***************************** Top of Data ******************************
000001 MPM5 = (DESCRIPTION =
000002          (ADDRESS_LIST=
000003            (ADDRESS=
000004              (PROTOCOL=TCP)
000005              (HOST=wtsc040e)
000006              (PORT=1521)
000006              (SSN=TNS5)
000007            )
000008          )
000009        )
****** *************************** Bottom of Data
```

Figure 53. TNS connect descriptor

You may include a SQLNET DD statement such as the following example, which points to your diagnostic parameter settings, as shown in Figure 54 on page 110:

```
//SQLNET    DD DISP=SHR,DSN=MPM5.ORA815.PARMLIB(SQLNET),
```

We had to copy the member MPM5.ORA815.NET8.SAMPLIB(SQLNET8S) to MPM5.ORA815.PARMLIB(SQLNET) and make the changes. We did not turn tracing on.

```
EDIT       MPM5.ORA815.PARMLIB(SQLNET) - 01.01            Columns 00001
00072
 000001 automatic_ipc=off
 000002 sqltrace=off
```

Figure 54. Net8 diagnostic parameter

If you set any trace options on, be sure to include the corresponding output data sets:

```
//MPMTNSLG DD SYSOUT=*
//MPMTNSTC DD SYSOUT=*
//SQLNETLG DD SYSOUT=*
//SQLNETTC DD SYSOUT=*
```

We ran without these output data sets and without the SQLNET DD statement, and did not encounter any error messages.

Recycle Oracle to activate the MPMTNS DD statement

You have to stop and start Oracle to include the MPMTNS DD statement so the listener will start correctly. To do this go to the SDSF console log and enter:

```
/ ( to be able to enter a log command)
MPM5 start svrmgrl command='shutdown immediate'
```

Then you can restart Oracle with the command: `/s orampm5.mpm5`

D.3 Starting Net8

We did not configure Oracle Names, since we were not using this product. Oracle recommends the use of Oracle Names for installations with more than 200 nodes.

We had to change the TCP/IP name (TCPSTK parameter) in Net8, however, because our name was not the default TCP/IP name. Since our TCP/IP stack name was not the default of TCP/IP, we had to add the following DD card to allow Net8 to resolve the `gethostbyname` command that it issues:

```
//SYSTCPD  DD DSN=TCPIPOE.SC04.TCPPARMS(TCPDATA),DISP=SHR
```

Now you can start Net8 by using operator commands on the system console or under SDSF, as follows:

```
S ORATNS5.TNS5

MPM5 START NET.MPMTNS
```

The first command starts the TNS subsystem. The second command starts the Net8 master task. This is the Net8 master task that listens for incoming requests.

Starting the listener when starting the database

To automate the Net8 listener whenever the database is started, you include the following statement, where the * is the COMCHAR selected on line 000003 in the MPMPARM member (Net8 must be running):

```
* START NET.MPMTNS
```

```
EDIT       MPM5.ORA815.PARMLIB(MPMPARM) - 01.04            Columns 00001 00072
 ****** ***************************** Top of Data ****************************
 000001 LANGUAGE='AMERICAN_AMERICA.WE8EBCDIC1047'
 000002 SSNAME='MPM5'
 000003 COMCHAR='*'
 000004 SERVER_OPTS=(PARQ,DIST)
 000005 KERNEL='ORACODE'
 000006 USERS=50
 000007 CONBUF=13
 000008 STACKSIZE=256000
 000009 TRACESIZE=32767
 000010 TRACEDS="MPM5.ORA815.TRACE** "
 000011 PRIVUSER=ORACLE4
 000012 SSCOMMENT='VER(VERSION8)'
 000013 * START NET.MPMTNS
 ****** **************************** Bottom of Data **********************
```

Figure 55. Oracle PARMLIB: MPM initialization with Net8 Start

When you start TNS, you should see the console output shown in Figure 56:

```
/ S ORATNS5.TNS5

 $HASP100 ORATNS5 ON STCINRDR
 IEF695I START ORATNS5 WITH JOBNAME ORATNS5 IS ASSIGNED TO USER
           , GROUP SYS1
 $HASP373 ORATNS5 STARTED
 IEF403I ORATNS5 - STARTED - TIME=17.04.59
 TXM00001I TNS5 subsystem initialization in progress, version
      8.0.4.0.50 Nov  5 1998 - Production
 TXM00002I TGV is at 0BE5B400
 TXM00003I Reserved LX number 22528 (x5800)
 TXM00005I Subsystem command character is '@'
 TXM00002I SSVT is at 00AA66D8
 TXM00006I TNS5 subsystem initialization complete
 TXM00100I TCP/IP networking task initializing
 TXM11825I IBM TCP/IP Macro API Networking Task (TNTJ) version
      8.0.4.0.50 Jan 13 1999 - Limited Production
 TXM11826I IBM TCP/IP is Initialized for IP=9.12.2.20, AS=TNS5
      Host=wtsc04oe
```

Figure 56. Console output at TNS startup

When you then start MPM, you should see the console output shown in Figure 57 on page 112.

```
S ORAMPM5.MPM5
 $HASP100 ORAMPM5 ON STCINRDR
 IEF695I START ORAMPM5 WITH JOBNAME ORAMPM5 IS ASSIGNED TO USER STC
    , GROUP SYS1
 $HASP373 ORAMPM5 STARTED
 IEF403I ORAMPM5 - STARTED - TIME=14.41.08 - ASID=0099.
 MPM MPM002I MPM INITIALIZING
 >MPM217I ************************************************************
 >MPM258I * MPM VERSION      1.1.09.00.00
 >MPM259I * MPM LINK EDIT DATE:09/28/99 TIME:09.50
 >MPM217I ************************************************************
 >MPM085I 50 USERS, EACH 266K PRIVATE AREA (31K PGA), TOTAL OF 15023K
 BYTES
 >MPM223I MPM BUILT ESA CROSS MEMORY ENTRY TABLE
 >MPM160I ORACLE MPM SUBSYSTEM MPM5 IS READY TO ACCEPT REQUESTS.
 >MPM193I MPM5 SMFRECN=199  SMF RECORDING IS ACTIVE
 >MPM008I OK.
 >MPM008I OK.
 >MPM008I OK.
 >MPM008I OK.
 >MPM008I OK.
 >MPM025I TASK MANAGER COMING UP.
 >MPM099I ATTACHED TCB AT X7DA4A0 FOR TASK 'TASK0001'
 >MPM099I ATTACHED TCB AT X7DA1D0 FOR TASK 'SVRMGRL'
 >MPM099I ATTACHED TCB AT X7DA038 FOR TASK 'TASK0002'
 >MPM099I ATTACHED TCB AT X7C1200 FOR TASK 'TASK0003'
>MPM099I ATTACHED TCB AT X7C4AC8 FOR TASK 'TASK0004'

  Oracle Server Manager Release 3.1.5.0.0 - Production

  (c) Copyright 1997, Oracle Corporation.  All Rights Reserved.

  Oracle8 Enterprise Edition Release 8.1.5.0.0 - Production
  PL/SQL Release 8.1.5.0.0 - Production

  Echo                          ON
  SVRMGR> CONNECT INTERNAL
  Connected.
  SVRMGR> STARTUP PFILE=/DD/INITORA NOMOUNT
 >MPM099I ATTACHED TCB AT X7C47C8 FOR TASK 'PMON'
 >MPM099I ATTACHED TCB AT X7C4470 FOR TASK 'DBW0'
 >MPM099I ATTACHED TCB AT X7C41A0 FOR TASK 'LGWR'
 >MPM099I ATTACHED TCB AT X7C1E88 FOR TASK 'CKPT'
 >MPM099I ATTACHED TCB AT X7C1CF0 FOR TASK 'SMON'
 >MPM099I ATTACHED TCB AT X7C1B58 FOR TASK 'RECO'
  ORACLE instance started.
  Total System Global Area      23407072 bytes
  Fixed Size                       76256 bytes
  Variable Size                 21209088 bytes
  Database Buffers               2048000 bytes
  Redo Buffers                     73728 bytes
 SVRMGR> ALTER DATABASE MPM5 MOUNT;
Statement processed.
SVRMGR> ALTER DATABASE OPEN;
 Statement processed.

  Server Manager complete.
 >MPM056I 'SVRMGRL' (TCB X7DA1D0) SUBTASK TERMINATION - X0
```

Figure 57. Console output at MPM startup

When you start the listener, you should see the console output shown in Figure 58.

```
MPM5 START NET.MPMTNS
 >MPM008I OK.
 >MPM099I ATTACHED TCB AT X7C15E8 FOR TASK 'MPMTNS'
TXM00307I Listening on    (ADDRESS=(PROTOCOL=TCP)(HOST=9.12.2.10)(PORT
  =1521)(SSN=TNS5))
 TXM11800I TCP/IP listen now active for ORAMPM5 on port 1521 socket
 0000(1).
```

Figure 58. Console output at listener start

This message will appear as part of the database initialization if you have included the start command in the MPMPARM as described in Figure 55.

Appendix E. Special notices

This publication is intended to help customers who are migrating an Oracle database to an Oracle database on OS/390. The information in this publication is not intended as the specification of any programming interfaces that are provided by Oracle. See the PUBLICATIONS section of the IBM Programming Announcement for OS/390 for more information about what publications are considered to be product documentation.

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

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

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 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 information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

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.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

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

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

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

AIX
AS/400
DB2
IBM
Netfinity
NetView
OS/390
RS/6000
S/390
SP
System/390

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Appendix F. Related publications

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

F.1 IBM Redbooks

For information on ordering these publications see “How to get IBM Redbooks” on page 119.

- *Oracle Applications 11 for OS/390 Installation Guide*, SG24-4980 (version 02)
- *Oracle Applications for OS/390 Presentation Guide*, SG24-2084 (version 01)

F.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at `http://www.redbooks.ibm.com/` for information about all the CD-ROMs offered, updates and formats.

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 Redbooks Collection	SK2T-8038
Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
AS/400 Redbooks Collection	SK2T-2849
Netfinity Hardware and Software Redbooks Collection	SK2T-8046
RS/6000 Redbooks Collection (BkMgr Format)	SK2T-8040
RS/6000 Redbooks Collection (PDF Format)	SK2T-8043
Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

F.3 Other resources

These publications are also relevant as further information sources, and are available from the Oracle Web site at `http://www.oracle.com`

- *Oracle8 for OS/390 Installation Guide*, A68790-1
- *Oracle8 for OS/390 Message Guide*, A68157-01
- *Oracle8 for OS/390 Migration Guide*
- *Oracle8 for OS/390 System Administration Guide Release 8.0.4*, A68791-01
- *Oracle8 for OS/390 User's Guide*, A68158-1
- *Oracle Utilities Manual*, A58244
- *Oracle Applications Release 11 for UNIX Installation*, A579803
- *Oracle8i Enterprise Edition for OS/390 Installation Guide*, A77059-1

F.4 Referenced Web sites

These Web sites are also relevant as further information sources:

Oracle home page:

- http://www.oracle.com

Oracle products and platforms:

- http://www.oracle.com/ibm
- http://www.oracle.com/products/applications
- http://www.oracle.com/platforms/ibm/s390/html/s390index.html

Oracle Data Warehouse:

- http://www.oracle.com/products/olap/html:exph

Oracle Client/Server Development Tools:

- http://www.oracle.com/products/tools:exph

Oracle Transparent Gateways for IBM:

- http://www.oracle.com/products/gateways/html/transparent.html:exp

Oracle8*i* for OS/390:

- http://www.oracle.com/products/ibm:exph

IBM S/390 Web sites:

- http://www.s390.ibm.com
- http://www.s390.ibm.com/oracle
- http://www.s390.ibm.com/applications/oracle.html
- http://www.ibm.com/erp/oracle_apps

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM 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-ROM 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 by e-mail including information from the IBM Redbooks fax order form to:

	e-mail address
In United States	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 may be found at the Redbooks Web site.

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 access MyNews at http://w3.ibm.com/ for redbook, residency, and workshop announcements.

IBM Redbooks 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

P

R

S

T

U

V

W

IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at http://www.redbooks.ibm.com/
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Document Number **Redbook Title**	SG24-4981-00 Experiences with Migrating Oracle to OS/390
Review	
What other subjects would you like to see IBM Redbooks address?	
Please rate your overall satisfaction:	O Very Good O Good O Average O Poor
Please identify yourself as belonging to one of the following groups:	O Customer O Business Partner O Solution Developer O IBM, Lotus or Tivoli Employee O None of the above
Your email address: The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.	O Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.
Questions about IBM's privacy policy?	The following link explains how we protect your personal information. http://www.ibm.com/privacy/yourprivacy/

SG24-4981-00

Printed in the U.S.A.

Experiences with Migrating Oracle to OS/390

SG24-4981-00