akadia
Information Technology

How to migrate Oracle 7.3 to 8.1.6

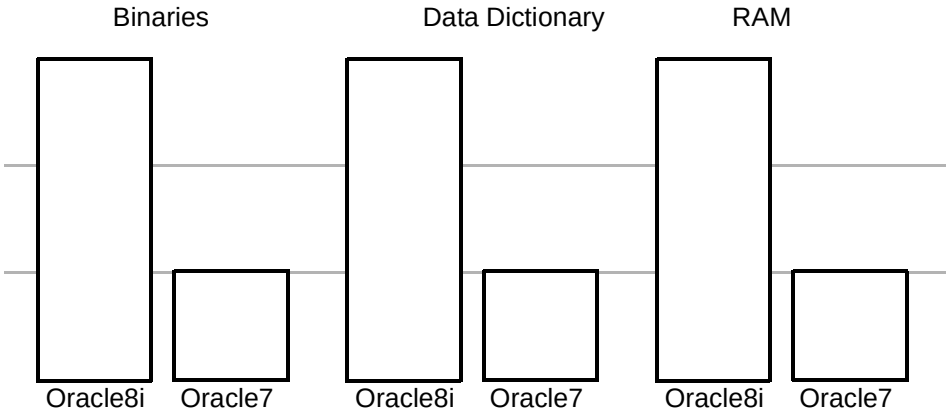
Martin Zahn / 20.03.2000

Akadia AG
Information Technology
Arvenweg 4
CH-3604 Thun
Tel 033 335 86 20
Fax 033 335 86 25
E-Mail info@akadia.com
Web http://www.akadia.com

1. Requirements

Hardware Requirements

Oracle 8.1.6 binaries may require as much as three times the disk space required by Oracle7 binaries. In addition, the **Oracle 8.1.6 data dictionary** may require as much as double the space of the Oracle7 data dictionary in the SYSTEM tablespace. Also, Oracle 8.1.6 may require up to twice as much **RAM** as Oracle7. The amount of RAM required also depends on the options you choose for the Oracle 8.1.6 environment.



Software Requirements

The version 2.5.1 of Solaris is not supported for Oracle 8.1.6 and higher, therefore it is necessary to migrate to Solaris 2.6 / 2.7. These steps can be made independent from the Oracle 8.1.6 migration, so they are not taken into account in this paper. The following software requirements must be ready and agreed with the different key persons.

Nr	Requirement
1	Operating System: Sun Solaris **2.6** or **2.7**

2	Operating System Patch:
	At least kernel jumbo **Patch Revision #105181-15** is required for successful installation of Release 8.1.6.
	- Download the Patch from: http://sunsolve.sun.com
	- Read the README File included in the Patch
	- Usually the only thing you have to do is:
	`$ cd <patch cluster directory>`
	`$ ./install_custer`
	`$ cat /var/sadm/install_data/<luster name>_log`
	`$ showrev -p`
	- Reboot the system

3	Operating System Packages: SUNWarc, SUNWbtool, SUNWlibm, SUNWlibms, SUNWsprot, SUNWtoo, SUNWhea.

Checks

To determine your current operating system information:

```
$ uname -a
```

To determine which operating system patches are installed:

```
$ showrev -p
```

To determine which operating system packages are installed:

```
$ pkginfo -i [package_name]
```

To determine if your X-windows system is working properly on your local system, but you can redirect the X-windows output to another system.

```
$ xclock
```

To determine if you are using the correct system executables:

```
$ /usr/bin/which make
$ /usr/bin/which ar
$ /usr/bin/which ld
$ /usr/bin/which nm
```

Each of the four commands above should point to the */usr/ccs/bin* directory. If not, add */usr/ccs/bin* to the beginning of the PATH environment variable in the current shell.

Java Runtime Environment (JRE)

The JRE shipped with Oracle 8.1.6 is used by Oracle Java applications such as the Oracle Universal Installer is the only one supported to run with these applications. Customers should not modify this JRE, unless it is done through a patch provided by Oracle Support Services. The inventory can contain multiple versions of the JRE, each of which can be used by one or more products or releases. The Installer creates the *oraInventory* directory the first time it is run to keep an inventory of products that it installs on your system as well as other installation information. The location of *oraInventory* is defined in */var/opt/oracle/oraInst.loc*. Products in an ORACLE_HOME access the JRE through a symbolic link in $ORACLE_HOME/JRE to the actual location of a JRE within the inventory. Customers should not modify the symbolic link.

The following Solaris 2.6 patches are required or recommended for the Java Runtime Environment JRE 1.1.8_10.

Patch ID	Description	Required or Recommended?
106040-11	X Input and Output Method patch	Required
105181-15	Kernel patch	Required
105284-25	Motif Runtime Library Patch	Recommended
105490-07	Dynamic linker patch	Recommended
105633-21	OpenWindows 3.6: Xsun patch (1)	Recommended
105568-13	Libthread patch	Recommended
105210-19	LibC patch	Recommended
105669-07	CDE 1.2: libDTSvc patch (dtmail)	Recommended
107636-01	X Input and Output Method patch	Required
106980-05	Libthread patch	Recommended
107607-01	Motif fontlist, fontset, libxm	Recommended
107078-10	Open Windows 3.6.1 Xsun patch (1)	Recommended

Check Software Limits

Oracle 8.1.6 includes native support for files greater than 2 GB. Check your shell to determine whether it will impose a limit.

To check current soft shell limits, enter the following command:

```
$ ulimit -Sa
```

To check maximum hard limits, enter the following command:

```
$ ulimit -Ha
```

The file (blocks) value should be multiplied by 512 to obtain the maximum file size imposed by the shell. A value of unlimited is the operating system default and is the maximum value of 1 TB.

Setup the Solaris Kernel

Set to the sum of the PROCESSES parameter for each Oracle database, adding the largest one twice, then add an additional 10 for each database. For example, consider a system that has three Oracle instances with the PROCESSES parameter in their initsid.ora files set to the following values:

```
ORACLE_SID=SOL1, PROCESSES=100
ORACLE_SID=SOL2, PROCESSES=100
ORACLE_SID=SOL3, PROCESSES=200
```

The value of SEMMNS is calculated as follows:

```
SEMMNS =  [(A=100) + (B=100)] + [(C=200) * 2] +
          [(# of instances=3) * 10] = 630
```

Setting parameters too high for the operating system can prevent the machine from booting up. Refer to Sun Microsystems Sun SPARC Solaris system administration documentation for parameter limits.

```
*
* Kernel Parameters SUN Enterprise with 640MB
*
set shmsys:shminfo_shmmax=4294967295
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=100
set shmsys:shminfo_shmseg=10
set semsys:seminfo_semmni=100
set semsys:seminfo_semmsl=100
set semsys:seminfo_semmns=2500
set semsys:seminfo_semopm=100
set semsys:seminfo_semvmx=32767
```

2. Migration Methods

Migration Utility

Use the Migration utility to migrate an Oracle7 database to Oracle 8.1.6 The Migration utility is a command-line utility for migration of a complete database from Oracle7 to Oracle 8.1.6. It changes datafile headers but leaves actual data unchanged. It does not copy data.

Export / Import

Perform a full or partial export of an Oracle7 source database, followed by a full or partial import into an Oracle 8.1.6 target database. Export/Import can migrate parts of the database. Export/Import leaves datafile headers and actual data unchanged, and makes a new copy of the data.

Copy over DB-Link

Data copying can migrate parts of the database. Data copying leaves datafile headers and actual data unchanged, and makes a new copy of the data.

Characteristics of Different Migration Methods:

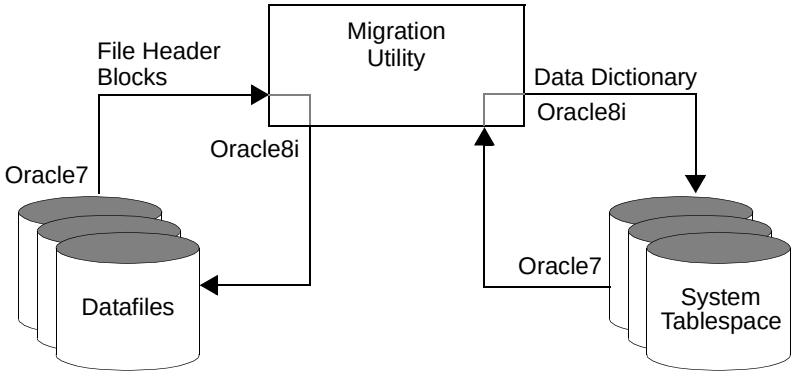
Migration Utility	Export/Import	Data Copying
Requires **minimal interaction** by the DBA.	Can migrate version 6 and Oracle7 databases to Oracle 8.1.6.	Datafiles **can be defragmented**, and migrated data compacted, to improve performance.
Relatively fast, whatever the size of the database, because the data dictionary objects are the only objects that are changed.	Can migrate specific parts of a database.	A database **can be restructured** with modified or new tablespaces.
Imposes essentially **no limit on the size of the database** it can migrate.	Can be used to downgrade between versions of Oracle, for example, downgrading from Oracle 8.1.6 to Oracle7.	Can migrate version 6 or Oracle7 databases to Oracle 8.1.6.
Requires relatively **little additional disk space**, when compared with other migration methods.	Datafiles **can be defragmented**, and migrated data compressed, to improve performance.	Can migrate specific parts of a database.
Performs only Oracle7 to Oracle 8.1.6 migrations, and cannot downgrade back to Oracle7	A database **can be restructured** with modified or new tablespaces, or by table partitioning.	Can be used to migrate to a different operating system and hardware platform.
Cannot migrate selected parts of a database; migrates only the entire database.	Can be used to migrate to a different operating system and hardware platform.	**Extremely slow** except for very small databases. Large databases of several gigabytes may take many hours.
Cannot migrate to a different operating system or hardware platform.	**Extremely slow** except for very small databases. Large databases of several gigabytes may take many hours	Requires that both source and target databases be available at once during copying operations.
	Requires large amounts of disk space for copying data into export file(s).	PL/SQL scripts to coy the data must be developed.

The Migration Utility performs best for the Migration of the Databases from Oracle 7.3.4.4 to Oracle 8.1.6, it has been tested several times on test systems.

3. Migration Overview

Migration Utility

The Migration utility is a command-line utility that converts files and structures in the Oracle7 source database to Oracle 8.1.6 format, changing only the file headers and, if necessary, the definitions of the data in the files. The Migration utility does not change the data portions of the datafiles, nor their format or content.



- The primary advantages of using the Migration utility are speed and relative ease of use.
- The Migration utility requires only enough temporary space in the SYSTEM tablespace to hold both the Oracle7 (source) and Oracle 8.1.6 (target) data dictionaries simultaneously.
- The Migration utility converts the entire database, including database files, rollback segments, and the control file(s).

Common Migration Problems

Time can be saved by eliminating common migration problems before migrating the databases. Common problem areas include the following:

- Rowids stored in columns or in application code. Because the **format for rowids is different in version 8**, the old rowids are invalid and must be converted.
- Make sure all Oracle product versions, operating system versions, and **third-party software versions are certified against Oracle 8.1.6** on your hardware platform.
- Before migrating, we should understand **the performance profile of the application under the source database**.
  For best results, run the SQL scripts *utlbstat.sql* and *utlestat.sql* to collect V$SYSSTAT statistics for a specific period. Use a collection timeframe that most consistently reflects peak production loads with consistent transaction activity levels.

Block Size Considerations

- The value of DB_BLOCK_SIZE in the Oracle7 database and in the migrated Oracle 8.1.6 database must be the same.

Migrate as Unix User «oracle»

Please note, that you MUST migrate as the «oracle» user, never as «root» or a normal UNIX user.

Outline of the Migration Process

In the Oracle7 Environment

- Run the Oracle 8.1.6 Migration utility, which creates and populates a new data dictionary based on the data dictionary of the Oracle7 database, and also creates a binary file based on the control file of the Oracle7 database. This binary file is called the **convert file**.

In the Oracle 8.1.6 Environment

- Execute an ALTER DATABASE CONVERT command, which creates a new control file based on the convert file generated by the Migration utility, converts all online datafile headers to Oracle 8.1.6 format, and mounts the Oracle 8.1.6 database.

  The file headers of offline datafiles and read-only tablespaces are not updated during migration. The file headers of offline datafiles are converted later when they are brought online, and the file headers of read-only tablespaces are converted if and when they are made read-write sometime after migration; however, they never have to be made read-write.

- Execute an ALTER DATABASE OPEN RESETLOGS statement, which automatically converts all objects and users defined in the new dictionary to Oracle 8.1.6 specifications, and converts all rollback segments to Oracle 8.1.6 format.

  If a source database rollback segment is in a tablespace that is offline when the Oracle 8.1.6 database is opened, the rollback segment is not converted immediately to Oracle 8.1.6 database format. Instead, the rollback segment is converted the first time the tablespace is brought online in Oracle 8.1.6.

- Run the database conversion scripts. The primary conversion script is the **u0703040.sql** script. This script creates and alters certain system tables and drops the MIGRATE user. **It also runs the catalog.sql and catproc.sql scripts**, which create the system catalog views and all the necessary packages for using PL/SQL.

- Other conversion scripts perform the necessary operations to convert specific components to the current release. For example, the catrep.sql script is one of the conversion scripts for Advance Replication.

Changing Word-Size

It is possible to change the word-size of the Oracle database server during a migration operation. A change in word-size includes the following scenarios:

- 32-bit Oracle software is installed on 64-bit hardware and want to be changed to 64-bit Oracle software.
- 64-bit Oracle software installed on 64-bit hardware and want to be changed to 32-bit Oracle software.

To change word-size during the migration, no additional action is required. The word-size is changed automatically during migration.

Replication Environments

It’s best to drop all snapshots and recreate them in the Oracle 8.1.6 environment. Note that Oracle 8.1.6 uses «Primary Key» Replication instead of the Oracle7 «ROWID Replication».

New Oracle Home Directory

An Oracle Home directory for the new Oracle 8.1.6 release that is separate from the Oracle7 Oracle home directory must be defined. It is not possible to install the Oracle 8.1.6 software into the same Oracle Home directory that is used for Oracle7.
Using separate installation directories enables to keep the Oracle7 software installed along with the Oracle 8.1.6 software. This method enables to test the migration process on an Oracle7 test database before replacing the production environment entirely.

The old ORACLE_HOME is: /opt/oracle/product/**7.3.4**
The new ORACLE_HOME is: /opt/oracle/product/**8.1.6**

Time needed

The whole migration using the migration utility, without rebuilding the replication and without the time needed for the backup[1] lasts about 2 - 3 hours per database.

1. Do not start the migration without a certified backup, this is the only way to go back when the migration fails. The Time needed for the backup is highly operator specific.

4. Install new Oracle 8.1.6 Release from CD-ROM

Setup Oracle 8.1.6 Environment

Setup ORACLE environment (Example)

```
ORACLE_HOME=/opt/oracle/product/8.1.6; export ORACLE_HOME
ORACLE_SID=SOL1; export ORACLE_SID
ORACLE_TERM=xterm; export ORACLE_TERM
TNS_ADMIN=/export/home/oracle/config/8.1.6; export TNS_ADMIN
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1; export NLS_LANG
ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data; export ORA_NLS33
LD_LIBRARY_PATH=$ORACLE_HOME/lib:/usr/lib:/usr/openwin/lib
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/dt/lib:/usr/ucblib
export LD_LIBRARY_PATH

PATH=/bin:/usr/bin:/usr/sbin:/opt/bin:/opt/local/GNU/bin
PATH=$PATH:/opt/local/bin:/opt/NSCPnav/bin:$ORACLE_HOME/bin:/usr/ucb:.
export PATH
```

Please note, that *$ORACLE_HOME/bin* must be in the PATH and check, that */usr/ucb* is behind */usr/ccs/bin*.

JDBC-Installation, set CLASSPATH

```
CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib
CLASSPATH=$CLASSPATH:$ORACLE_HOME/network/jlib
```

Setup Response File for Silent Install

Edit the Installer Response File, we used the file: ***svrtypical.rsp*** as a template. This File can be found on the CD-ROM: *<cdrom_mount_point>/stage/Response*. Please verify (customize) the file for your environment.

Check oraInst.loc File

If you used Oracle 8.1.x before on your system, then you must edit the Oracle Inventory File, located in */var/opt/oracle*.

```
# Oracle Installer Location File (/var/opt/oracle/oraInst.loc)

inventory_loc=/opt/oracle/product/8.1.6/oraInventory
```

Install with Response File

Install Oracle 8.1.6 with customized Response File: *svrtypical.rsp* as User «oracle». We suggest to start *runInstaller* in the foreground, therefore remove the "&" from runInstaller at the end of the file.

```
$ cd /cdrom/oracle8i
$ DISPLAY=<Any X-Window Host>:0.0
$ export DISPLAY
$ ./runInstaller -silent -responseFile \
   /export/home/oracle/config/8.1.5/svrtypical.rsp
```

Watch the File *$ORACLE_HOME/oraInventory/logs/installActions.log*. At the end of the Installation, check the file: *$ORACLE_HOME/oraInventory/logs/silentInstall.log*

Output in silentInstall2000-01-17_04-23-52-PM.log:

A configuration script needs to be run as root before installation can proceed. Please leave this window up, go run */opt/oracle/product/8.1.6/root.sh* as root from another window, then come back here and click OK to continue.The installation of Oracle8i Enterprise Edition was successful.

Start root.sh as «root»

Start the script root.sh as User «root». Note that the File */var/opt/oracle/oratab* is not used in the High Availability Environment.

```
$ su root
$ cd $ORACLE_HOME
$ ./root.sh
```

Setup Listener for Oracle 8.1.6

This must be done for all databases. Create the necessary log- and config directories. Set ORACLE_HOME to the new location.

```
$ cd $ORACLE_HOME
$ mkdir sqlnet
$ cd sqlnet
$ mkdir config log
$ cd config
```

Start /Stop and check the listener

It is very important that the listener works correctly. You must check the listener on system-1 and on system-2.

```
$ lsnrctl
LSNRCTL> start <ListenerName>
```

Check that listener starts without an error, check that listener is running.

```
$ ps -ef | grep tns
```

5. Prepare the Oracle 7 Source Database for Migration

Complete the following steps before migrating the Oracle7 database to Oracle 8.1.6. Prepare as much as you can, to keep the downtime as short as possible. This steps should be done a short time before the real migration starts.

Check Oracle 7 Release Number

The needed Release is 7.3.4 with the Procedural Option installed. Start Server Manager to check this.

```
$ svrmgrl

Oracle Server Manager Release 2.3.4.0.0 - Production
Copyright (c) Oracle Corporation 1994, 1995. All rights reserved.
Oracle7 Server Release 7.3.4.4.0 - Production
With the distributed, replication and parallel query options
PL/SQL Release 2.3.4.4.0 - Production
```

Check State of Ora7 DB-Files

Make sure all datafiles and tablespaces are either online or offline normal.

```
SVRMGR> SELECT file#,status,enabled,name FROM v$datafile;
```

To determine whether any datafiles require recovery:

```
SVRMGR> SELECT * FROM v$recover_file;
```

We should see a "0 rows selected" message, which indicates that all datafiles are either online or offline normal. If any datafiles are listed, we must restore the datafiles before we migrate the database. We can use the V$DATAFILE dynamic performance view to find the datafile name based on the datafile number. The Oracle 8.1.6 Migration utility will not proceed, and will display an error, if any datafiles require media recovery.

Check OUTLN User or Role

Make sure no user or role has the name OUTLN, because this schema is created automatically when you install Oracle 8.1.6. If you have a user or role named OUTLN, you must drop the user or role and recreate it with a different name.

```
SVRMGR> SELECT username FROM dba_users WHERE username = 'OUTLN';
SVRMGR> SELECT role FROM dba_roles WHERE role = 'OUTLN';
```

Check MIGRATE User or Role

Make sure no user or role has the name MIGRATE, because the Oracle 8.1.6 Migration utility creates this schema and uses it to replace any pre-existing user or role with this name, and finally drops it from the system.

```
SVRMGR> SELECT username FROM dba_users WHERE username = 'MIGRATE';
SVRMGR> SELECT role FROM dba_roles WHERE role = 'MIGRATE';
```

Check for OPTIMAL setting

Make sure the SYSTEM rollback segment does not have an OPTIMAL setting. An OPTIMAL setting may cause errors during migration.

```
SVRMGR> SELECT a.usn, a.name, b.optsize
        FROM v$rollname a, v$rollstat b
        WHERE a.usn = b.usn AND name = 'SYSTEM';
```

Your output should be similar to the following:

```
USN        NAME                           OPTSIZE
---------- ------------------------------ ----------
         0 SYSTEM
```

If there is a value in the OPTSIZE column, issue the following SQL statement to set optimal to NULL:

```
SVRMGR> ALTER ROLLBACK SEGMENT SYSTEM STORAGE (OPTIMAL NULL);
```

You must reset OPTIMAL when migration is complete.

Check In-Doubt Transactions

Make sure that no redo information and no uncommitted transaction are outstanding; resolve every pending «In-Doubt Transaction». You can check the «DBA_2PC_PENDING» view; if there are no rows, then you are clean. If you have pending transaction, try to overwrite the «In-Doubt Transaction» manually. More about «In-Doubt Transaction» can be found in the Oracle8 Server Distributed Manual on how to overwrite the «In-Doubt Transaction» manually.

In doubt transactions may occur at double phase commit time for network, break node reason. Normally if a double phase commit failed, you may have some entries in views DBA_2PC_PENDING and DBA_2PC_NEIGHBORS. To force the in doubt transaction to rollback you have to use the command:

```
SVRMGR> SELECT SUBSTR(local_tran_id,1,20) "Local Trans ID",
               SUBSTR(global_tran_id,1,20) "Global Trans ID",
                state "State"
          FROM dba_2pc_pending;

SVRMGR> ROLLBACK FORCE <local_tran_id>;
```

Unfortunately, sometime the entries are still there ... and you may discover in your alert<sid>.log file something like: ora-600 [18104] ...

Now it's possible to use package DBMS_TRANSACTION to solve the problem if rollback force do not clean all entries. Do as follow:

```
SVRMGR> EXECUTE DBMS_TRANSACTION.PURGE_LOST_DB_ENTRY('xid');
```

Where 'xid' come from:

```
SVRMGR> SELECT local_tran_id FROM DBA_2PC_PENDING;
```

Prepare Listener.ora for Oracle 8.1.6

The configuration file for the Net8 listener needs only small modifications. Replace the old log- and trace destination: /opt/oracle/product/**7.3.4** with the new one: /opt/oracle/product/**8.1.6.**

Prepare INIT.ora for Oracle 8.1.6

Each new release of Oracle introduces new initialization parameters, changes some parameters, and obsoletes some parameters. You must adjust your *INITSID.ORA* file to account for these changes and to take advantage of new initialization parameters that may be beneficial for your database. The COMPATIBLE initialization parameter controls the compatibility level of your database. Set the COMPATIBLE initialization parameter in your *INITSID.ORA* to 8.1.6.0.

Obsolete initialization parameters in Oracle 8.0

CHECKPOINT_PROCESS	FAST_CACHE_FLUSH
GC_DB_LOCKS	GC_FREELIST_GROUPS
GC_ROLLBACK_SEGMENTS	GC_SAVE_ROLLBACK_LOCKS
GC_SEGMENTS	GC_TABLESPACES
IO_TIMEOUT	INIT_SQL_FILES
IPQ_ADDRESS	IPQ_NET
LM_DOMAINS	LM_NON_FAULT_TOLERANT
MLS_LABEL_FORMAT	OPTIMIZER_PARALLEL_PASS
PARALLEL_DEFAULT_MAX_SCANS	PARALLEL_DEFAULT_SCAN_SIZE
POST_WAIT_DEVICE	SEQUENCE_CACHE_HASH_BUCKETS
UNLIMITED_ROLLBACK_SEGMENTS	USE_IPQ
USE_POST_WAIT_DRIVER	USE_READV
USE_SIGIO	V733_PLANS_ENABLED

Obsolete initialization parameters in Oracle 8.1

ALLOW_PARTIAL_SN_RESULTS	ARCH_IO_SLAVES
B_TREE_BITMAP_PLANS	BACKUP_DISK_IO_SLAVES
CACHE_SIZE_THRESHOLD	CLEANUP_ROLLBACK_ENTRIES
CLOSE_CACHED_OPEN_CURSORS	COMPATIBLE_NO_RECOVERY
COMPLEX_VIEW_MERGING	DB_BLOCK_CHECKPOINT_BATCH
DB_BLOCK_LRU_EXTENDED_STATISTICS	DB_BLOCK_LRU_STATISTICS
DB_FILE_SIMULTANEOUS_WRITES	DELAYED_LOGGING_BLOCK_CLEANOUTS
DISCRETE_TRANSACTIONS_ENABLED	DISTRIBUTED_LOCK_TIMEOUT
DISTRIBUTED_RECOVERY_CONNECTION_HOLD_TIME	FAST_FULL_SCAN_ENABLED
FREEZE_DB_FOR_FAST_INSTANCE_RECOVERY	GC_LATCHES
GC_LCK_PROCS	JOB_QUEUE_KEEP_CONNECTIONS
LARGE_POOL_MIN_ALLOC	LGWR_IO_SLAVES
LOCK_SGA_AREAS	LOG_ARCHIVE_BUFFER_SIZE
LOG_ARCHIVE_BUFFERS	LOG_BLOCK_CHECKSUM
LOG_FILES	LOG_SIMULTANEOUS_COPIES
LOG_SMALL_ENTRY_MAX_SIZE	MAX_TRANSACTION_BRANCHES
MTS_LISTENER_ADDRESS	MTS_MULTIPLE_LISTENERS
MTS_RATE_LOG_SIZE	MTS_RATE_SCALE
MTS_SERVICE	OGMS_HOME
OPS_ADMIN_GROUP	PARALLEL_DEFAULT_MAX_INSTANCES
PARALLEL_MIN_MESSAGE_POOL	PARALLEL_SERVER_IDLE_TIME
PARALLEL_TRANSACTION_RESOURCE_TIMEOUT	PUSH_JOIN_PREDICATE
REDUCE_ALARM	ROW_CACHE_CURSORS
SEQUENCE_CACHE_ENTRIES	SEQUENCE_CACHE_HASH_BUCKETS
SHARED_POOL_RESERVED_MIN_ALLOC	SNAPSHOT_REFRESH_KEEP_CONNECTIONS
SNAPSHOT_REFRESH_PROCESSES	SORT_DIRECT_WRITES
SORT_READ_FAC	SORT_SPACEMAP_SIZE
SORT_WRITE_BUFFER_SIZE	SORT_WRITE_BUFFERS
SPIN_COUNT	TEMPORARY_TABLE_LOCKS
TEXT_ENABLE	USE_ISM

Initialization Parameters Added in Release 8.0

ALLOW_PARTIAL_SN_RESULTS	ALWAYS_SEMI_JOIN
AQ_TM_PROCESSES	ARCH_IO_SLAVES
BACKUP_DISK_IO_SLAVES	BACKUP_TAPE_IO_SLAVES
BUFFER_POOL_KEEP	BUFFER_POOL_RECYCLE
COMPLEX_VIEW_MERGING	CONTROL_FILE_RECORD_KEEP_TIME
DB_BLOCK_MAX_DIRTY_TARGET	DB_FILE_DIRECT_IO_COUNT
DB_FILE_NAME_CONVERT	DB_WRITER_PROCESSES
DBWR_IO_SLAVES	DISK_ASYNCH_IO
FAST_FULL_SCAN_ENABLED	FREEZE_DB_FOR_FAST_INSTANCE_RECOVERY
GC_DEFER_TIME	GC_LATCHES
HI_SHARED_MEMORY_ADDRESS	INSTANCE_GROUPS
LARGE_POOL_MIN_ALLOC	LARGE_POOL_SIZE
LGWR_IO_SLAVES	LM_LOCKS
LM_PROCS	LM_RESS
LOCAL_LISTENER	LOCK_NAME_SPACE
LOCK_SGA	LOCK_SGA_AREAS
LOG_ARCHIVE_DUPLEX_DEST	LOG_ARCHIVE_MIN_SUCCEED_DEST
LOG_FILE_NAME_CONVERT	MTS_RATE_LOG_SIZE
MTS_RATE_SCALE	NLS_CALENDAR
O7_DICTIONARY_ACCESSIBILITY	OBJECT_CACHE_MAX_SIZE_PERCENT
OBJECT_CACHE_OPTIMAL_SIZE	OGMS_HOME
OPEN_LINKS_PER_INSTANCE	OPS_ADMIN_GROUP
OPTIMIZER_FEATURES_ENABLE	OPTIMIZER_INDEX_CACHING
OPTIMIZER_INDEX_COST_ADJ	OPTIMIZER_MAX_PERMUTATIONS
PARALLEL_ADAPTIVE_MULTI_USER	PARALLEL_BROADCAST_ENABLED
PARALLEL_EXECUTION_MESSAGE_SIZE	PARALLEL_INSTANCE_GROUP
PARALLEL_MIN_MESSAGE_POOL	PARALLEL_SERVER
PARALLEL_TRANSACTION_RESOURCE_TIMEOUT	PLSQL_V2_COMPATIBILITY
PUSH_JOIN_PREDICATE	READ_ONLY_OPEN_DELAYED
REPLICATION_DEPENDENCY_TRACKING	SERIAL_REUSE
SESSION_MAX_OPEN_FILES	SHARED_MEMORY_ADDRESS
STAR_TRANSFORMATION_ENABLED	TAPE_ASYNCH_IO
TIMED_OS_STATISTICS	TRANSACTION_AUDITING
USE_INDIRECT_DATA_BUFFERS	

Initialization Parameters Added in Release 8.1

DB_BLOCK_CHECKING	ENT_DOMAIN_NAME
FAST_START_IO_TARGET	FAST_START_PARALLEL_ROLLBACK
HS_AUTOREGISTER	INSTANCE_NAME
JAVA_POOL_SIZE	LOG_ARCHIVE_DEST_n
LOG_ARCHIVE_DEST_STATE_n	LOG_ARCHIVE_MAX_PROCESSES
NLS_COMP	NLS_DUAL_CURRENCY
PARALLEL_AUTOMATIC_TUNING	PARALLEL_SERVER_INSTANCES
PARALLEL_THREADS_PER_CPU	PLSQL_LOAD_WITHOUT_RECOMPILE
QUERY_REWRITE_ENABLED	QUERY_REWRITE_INTEGRITY
RESOURCE_MANAGER_PLAN	SERVICE_NAMES
SORT_MULTIBLOCK_READ_COUNT	STANDBY_ARCHIVE_DEST

Initialization Parameters Renamed in Release 8.0 / 8.1

Pre-Release 8.0 Name	Release 8.0 Name
ASYNC_READ	DISK_ASYNCH_IO
ASYNC_WRITE	DISK_ASYNCH_IO
CCF_IO_SIZE *	DB_FILE_DIRECT_IO_COUNT *
DB_FILE_STANDBY_NAME_CONVERT	DB_FILE_NAME_CONVERT
DB_WRITERS	DBWR_IO_SLAVES
LOG_FILE_STANDBY_NAME_CONVERT	LOG_FILE_NAME_CONVERT
SNAPSHOT_REFRESH_INTERVAL	JOB_QUEUE_INTERVAL
MVIEW_REWRITE_ENABLED	QUERY_REWRITE_ENABLED
REWRITE_INTEGRITY	QUERY_REWRITE_INTEGRITY
NLS_UNION_CURRENCY	NLS_DUAL_CURRENCY
PARALLEL_TRANSACTION_RECOVERY	FAST_START_PARALLEL_ROLLBACK

Compatibility Issues with Initialization Parameters

Certain initialization parameter changes require special attention because they may raise compatibility issues for your database.

New Default Value for LOG_CHECKPOINT_TIMEOUT

Starting in release 8.1.6, the LOG_CHECKPOINT_TIMEOUT initialization parameter has a new default value. In previous releases, the default value was zero seconds, but in release 8.1.6 and higher, the default value is 1800 seconds.

Data Dictionary Protection

O7_DICTIONARY_ACCESSIBILITY is the initialization parameter switch that continues Oracle7 data dictionary behavior. It is only a temporary expedient and is not planned for future Oracle 8.1.6 releases.

The DML_LOCKS Parameter

Oracle 8.1.6 systems typically consume more DML locks while performing DDL operations than are required for Oracle7 systems. Nevertheless, the Oracle7 DML_LOCKS parameter default settings usually are adequate for Oracle 8.1.6 systems, even for DML-intensive applications.

The default value of DML_LOCKS is a multiple of the number of transactions, which is calculated from the number of rollback segments. However, in Oracle 8.1.6 fewer transactions are used per rollback segment than are used in Oracle7. Consequently, DML_LOCKS has a lower default value in Oracle 8.1.6. Under some extreme test conditions, an Oracle 8.1.6 system exceeded its (Oracle7) DML lock limit, and the DML_LOCKS parameter value had to be increased.

The DB_DOMAIN Parameter

Beginning with release 8.1, if the DB_DOMAIN initialization parameter is unset, it is set to NULL by default. In prior releases of Oracle, the default setting was the following: *.WORLD.*

A NULL setting for DB_DOMAIN may cause database **connection problems** in some environments. Before you migrate or upgrade to release 8.1, make sure the DB_DOMAIN parameter in your *INITSID.ORA* file is set to one of the following:

- .WORLD
- a valid domain setting for your environment

If DB_DOMAIN is not set in your current database, set it to *.WORLD* before you migrate or upgrade to release 8.1. If DB_DOMAIN is set to a valid domain for your environment in your current database, retain the setting in your INITSID.ORA file when you migrate to release 8.1.

Parallel Execution Allocated from Large Pool

Starting with release 8.1, parallel execution message buffers are allocated from the large pool whenever **PARALLEL_AUTOMATIC_TUNING**, a new initialization parameter, is set to TRUE. In past releases, this allocation was from the shared pool. If you are migrating or upgrading to release 8.1 and you choose to set PARALLEL_AUTOMATIC_TUNING to TRUE, you can **avoid problems** by modifying the settings for the following initialization parameters:

- SHARED_POOL_SIZE
- LARGE_POOL_SIZE

Typically, you should reduce the setting for SHARED_POOL_SIZE and raise the setting for LARGE_POOL_SIZE to avoid problems. Alternatively, you can reduce the setting for SHARED_POOL_SIZE and let Oracle calculate the setting for LARGE_POOL_SIZE. Oracle calculates a default LARGE_POOL_SIZE only if PARALLEL_AUTOMATIC_TUNING is set to TRUE and LARGE_POOL_SIZE is unset.

The calculation is based on the settings for the following parameters:

- PARALLEL_MAX_SERVERS
- PARALLEL_THREADS_PER_CPU
- PARALLEL_SERVER_INSTANCES
- MTS_DISPATCHERS
- DBWR_IO_SLAVES

If PARALLEL_AUTOMATIC_TUNING is unset or set to FALSE, and if LARGE_POOL_SIZE is unset, the value for LARGE_POOL_SIZE defaults to zero. The following scenarios illustrate the behavior that results from various initialization parameter settings when you migrate or upgrade to release 8.1.

Possible Scenarios

Retaining Parameter Settings without Modifications

Parameter	Setting
PARALLEL_AUTOMATIC_TUNING	Unset (defaults to FALSE).
SHARED_POOL_SIZE	Set to a large value, including the space required forparallel execution.
LARGE_POOL_SIZE	Unset (defaults to zero).

These settings are the most common scenario. In this case, you already have accounted for the space required for parallel execution in the shared pool.

Using PARALLEL_AUTOMATIC_TUNING

Parameter	Setting
PARALLEL_AUTOMATIC_TUNING	Set to TRUE.
SHARED_POOL_SIZE	Set to a small value that accounts for all clients except parallel execution.
LARGE_POOL_SIZE	Unset (defaults to a large value that includes the space required for parallel execution).

In this case, parallel execution allocates buffers from the large pool based on Oracle's automatic calculation. Buffer allocation is more efficient, and failures to allocate are isolated from the clients of the shared pool.

Using PARALLEL_AUTOMATIC_TUNING and Setting LARGE_POOL_SIZE

Parameter	Setting
PARALLEL_AUTOMATIC_TUNING	Set to TRUE.
SHARED_POOL_SIZE	Set to a small value that accounts for all clients except parallel execution.
LARGE_POOL_SIZE	Set to a value that includes the space required for parallel execution.

In this case, parallel execution allocates buffers from the large pool. After initial testing with LARGE_POOL_SIZE unset, you determined that the default calculation for LARGE_POOL_SIZE did not reflect your requirements for the large pool. Therefore, you decided to set LARGE_POOL_SIZE manually. After you set LARGE_POOL_SIZE properly, buffer allocation is more efficient, and failures to allocate are isolated from the clients of the shared pool.

Using PARALLEL_AUTOMATIC_TUNING without Modifying SHARED_POOL_SIZE

Parameter	Setting
PARALLEL_AUTOMATIC_TUNING	Set to TRUE.
SHARED_POOL_SIZE	Set to a large value, including the space required for parallel execution.
LARGE_POOL_SIZE	Unset (defaults to a large value that includes the space required for parallel execution).

In this case, parallel execution allocates buffers from the large pool, but because you did not modify SHARED_POOL_SIZE, it is likely that the SGA will be unnecessarily large, causing performance problems. Therefore, avoid setting PARALLEL_AUTOMATIC_TUNING to TRUE without modifying the settings of SHARED_POOL_SIZE and LARGE_POOL_SIZE appropriately.

Archive Log Destination Parameters

Release 8.1 supports new archive log destination parameters. After you migrate or upgrade to release 8.1, you can dynamically convert from the old pre-release 8.1 parameters (LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_DEST) to the new release 8.1 parameters (LOG_ARCHIVE_DEST_n and LOG_ARCHIVE_DEST_STATE_n).

Prepare Migration Environment

During migration from 7.3.4 to 8.1.6 a special environment is needed. ORA_NLS33 must point $ORACLE_HOME/migrate/nls/admin/data, where ORACLE_HOME is the Orcale7 Oracle Home. Check NLS_LANG as well, it must be correct for the characterset you are using.

Check NLS_LANG

```
SVRMGR> SELECT * FROM v$nls_parameters
        WHERE parameter = 'NLS_LANGUAGE'
           OR parameter = 'NLS_TERRITORY'
            OR parameter = 'NLS_CHARACTERSET';

PARAMETER          VALUE
------------------ ----------------
NLS_LANGUAGE       AMERICAN
NLS_TERRITORY      AMERICA
NLS_CHARACTERSET   WE8ISO8859P1
```

Example

```
ORACLE_HOME=/opt/oracle/product/7.3.4; export ORACLE_HOME
ORACLE_TERM=vt100; export ORACLE_TERM
TNS_ADMIN=$ORACLE_HOME/sqlnet/config; export TNS_ADMIN
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1; export NLS_LANG
ORA_NLS32=$ORACLE_HOME/ocommon/nls/admin/data; export ORA_NLS32
ORA_NLS33=$ORACLE_HOME/migrate/nls/admin/data; export ORA_NLS33
LD_LIBRARY_PATH=$ORACLE_HOME/lib:/usr/lib:/usr/openwin/lib; export LD_LIBRARY_PATH
```

Prepare Oracle 8.1.6 Environment

Prepare the new Oracle 8.1.6 environment. If you require /usr/ucb in your PATH, put it at the end of the PATH environment variable list; otherwise, it may cause installation problems. Make sure /usr/ccs/bin is before /usr/ucb in the PATH. If you have the ORA_NLS environment variable set in your .profile, remove this variable. ORA_NLS is wrong and should be removed.

Example

```
ORACLE_HOME=/opt/oracle/product/8.1.6; export ORACLE_HOME
ORACLE_TERM=vt100; export ORACLE_TERM
TNS_ADMIN=$ORACLE_HOME/sqlnet/config; export TNS_ADMIN
NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1; export NLS_LANG
ORA_NLS33=$ORACLE_HOME/ocommon/nls/admin/data; export ORA_NLS33
LD_LIBRARY_PATH=$ORACLE_HOME/lib:/usr/lib:/usr/openwin/lib; export LD_LIBRARY_PATH
```

Document your DB

Locate the database files for all databases. If these databasefiles are NOT located within the ORACLE_HOME tree, the migration is easier, because the database files must not be relocated (moved from old ORACLE_HOME to new ORACLE_HOME). Usually this is the true, if the operator have not created additional databasefiles in $ORACLE_HOME. To be sure, it‘s very important to document the database first, before the migration starts, this must be done for all databases.

Set environment for your «old» 7.3.4 database (check with env).

```
$ ORACLE_SID=SOL1
$ export ORACLE_SID

svrmgrl
CONNECT INTERNAL

SVRMGR> SELECT * FROM v$dbfile;
SVRMGR> SELECT * FROM v$logfile;
```

Backup the Oracle 7 Controlfile

Backup the Oracle7 Controlfile with the following Command. Check, that no database files are within $ORACLE_HOME.

```
SVRMGR> ALTER DATABASE BACKUP CONTROLFILE TO TRACE;
```

This file is saved in /var/opt/sol1/udmp, look up for the youngest file.

```
Dump file /var/opt/sol1/udmp/sol1_ora_11864.trc
Oracle7 Server Release 7.3.3.0.0 - Production Release
With the distributed, replication and parallel query options
PL/SQL Release 2.3.3.0.0 - Production
ORACLE_HOME = /opt/oracle/product/7.3.3
System name:SunOS
Node name:test
Release:5.6
Version:Generic_103640-24
Machine:sun4u
Instance name: SOL1
Redo thread mounted by this instance: 1
Oracle process number: 29
Unix process pid: 11864, image: oracleSOL1
Thu Mar  4 13:09:56 1999
Thu Mar  4 13:09:56 1999

*** SESSION ID:(29.4666) 1999.03.04.13.09.56.000

# The following commands will create a new control file and use it
# to open the database.
# No data other than log history will be lost. Additional logs may
# be required for media recovery of offline data files. Use this
# only if the current version of all online logs are available.
STARTUP NOMOUNT
CREATE CONTROLFILE REUSE DATABASE "SOL1"
    NORESETLOGS
    ARCHIVELOG
    MAXLOGFILES 62
    MAXLOGMEMBERS 3
    MAXDATAFILES 256
    MAXINSTANCES 4
    MAXLOGHISTORY 1600
LOGFILE
  GROUP 1 (
    '/data/sol1/db1/SOL1_log1A.rdo',
    '/data/sol1/db1/SOL1_log1B.rdo'
  ) SIZE 5M,
  GROUP 2 (
    '/data/sol1/db1/SOL1_log2A.rdo',
    '/data/sol1/db1/SOL1_log2B.rdo'
  ) SIZE 5M
DATAFILE
  '/data/sol1/db1/SOL1_sys1.dbf',
  '/data/sol1/db1/SOL1_rbs1.dbf',
  '/data/sol1/db1/SOL1_temp1.dbf',
  '/data/sol1/db1/SOL1_users1.dbf',
  '/data/sol1/db1/SOL1_tab1.dbf',
  '/data/sol1/db1/SOL1_cdr1.dbf',
  '/data/sol1/db1/SOL1_cre1.dbf',
  '/data/sol1/db1/SOL1_idx1.dbf'
;
# Recovery is required if any of the datafiles are restored
backups,
# or if the last shutdown was not normal or immediate.
RECOVER DATABASE
# All logs need archiving and a log switch is needed.
ALTER SYSTEM ARCHIVE LOG ALL;
# Database can now be opened normally.
ALTER DATABASE OPEN;
```

We can see, that NO databasefiles are within ORACLE_HOME, this is very good, therefore we don‘t have to relocate any files.

Reset Passwords for SYS and SYSTEM

Reset Passwords for SYS and SYSTEM to "manager", this must be done for all databases which will be migrated.

```
svrmgrl
SVRMGR> CONNECT INTERNAL
SVRMGR> STARTUP
SVRMGR> ALTER USER SYS IDENTIFIED BY manager;
SVRMGR> ALTER USER SYSTEM IDENTIFIED BY manager;
```

Document objects

Document all objects, check if they are valid.

```
sqlplus sol1/sol1@SOL1
SQL> SELECT DISTINCT (object_type) object, status, COUNT(*)
       FROM user_objects
     GROUP BY object_type, status;

OBJECT        STATUS    COUNT(*)
------------- ------- ----------
INDEX         VALID           95
PACKAGE       VALID            1
PACKAGE BODY  VALID            1
PROCEDURE     VALID            2
SEQUENCE      VALID           20
TABLE         VALID          132
TRIGGER       VALID           65
```

If one or more objects are in INVALID status, try to recompile them using the SQL script: ***compile_all.sql.***

Check constraints

Check that all constraints are enabled for the objects of the user SOL1, using the following SQL statement.

```
SQL> SELECT c.constraint_name,
            c.table_name,
            c.column_name,
            i.r_constraint_name,
            i.status
       FROM user_cons_columns c,
            user_constraints i
      WHERE i.table_name = c.table_name
        AND i.constraint_name = c.constraint_name
        AND i.owner = c.owner
        AND i.status = 'DISABLED'
   ORDER BY c.table_name,c.constraint_name,c.column_name;
```

Stop DBMS Jobs

```
SQL> SELECT SUBSTR(job,1,4) "Job",
     SUBSTR(schema_user,1,5) "User",
     SUBSTR(TO_CHAR(last_date,'DD.MM.YYYY HH24:MI'),1,16) "Last Date",
     SUBSTR(TO_CHAR(next_date,'DD.MM.YYYY HH24:MI'),1,16) "Next Date",
     SUBSTR(broken,1,2) "B",
     SUBSTR(failures,1,6) "Failed",
     SUBSTR(what,1,32) "Command"
     FROM dba_jobs;

Job  User  Last Date        Next Date        B Failed Command
---- ----- ---------------- ---------------- - ------ --------------------------------
1    SOL1  08.04.1999 02:00 09.04.1999 02:00 N 0      sol1.cleanup_partial_cdr(10);
2    SOL1  08.04.1999 03:00 09.04.1999 03:00 N 0      sys.dbms_utility.analyze_schema(
```

Login as the User listed in the column «User» (e.g. SOL1)

```
SQL>  execute dbms_job.broken(1,TRUE);
SQL>  execute dbms_job.broken(2,TRUE);
```

Check Size of SYSTEM Tablespace

Make sure you have enough space **(250MB)** in the SYSTEM tablespace. A common migration problem is running out of space in the SYSTEM tablespace during migration. The Migration utility will not complete the migration unless sufficient space is allocated in the SYSTEM tablespace.

To determine disk space requirements for a successful migration, run the following SQL statement **pointing to Oracle 7.3.4**

```
$ . $HOME/.profile-7.3.4
$ sqlplus system/manager
SQL> SELECT tablespace_name,
       ROUND(SUM(bytes)/1000000) "Size MB"
         FROM dba_data_files
    GROUP BY tablespace_name
    ORDER BY tablespace_name;

TABLESPACE_NAME                Size MB
------------------------------ ----------
CDR                                   744
CRE                                   220
IDX                                  1583
RBS                                   430
SYSTEM                                105
TAB                                   744
TEMP                                  157
USERS                                  10
```

If the the SYSTEM tablespace is less than 250MB, add more space to the SYSTEM tablespace, issue a command similar to the following, substituting the appropriate directory path and name for the new datafile and the amount of space you need to add:

```
SQL> SELECT tablespace_name,
            file_name
       FROM dba_data_files
   ORDER BY tablespace_name;

SQL> ALTER TABLESPACE system
     ADD DATAFILE '/u01/db/SOL3/sys/SOL3_sys2.dbf' SIZE 100M;
```

Do NOT set OPTIMAL for SYSTEM Rollback Segment

Make sure the SYSTEM rollback segment does not have an OPTIMAL setting. An OPTIMAL setting may cause errors during migration.

The Migration utility take all non-SYSTEM rollback segments offline and then freeze the size of the SYSTEM rollback segment by altering MAXEXTENTS to the number of extents currently allocated. This action prevents any space operations, such as an extent allocation, while the utility or assistant handles the space management tables.

If the SYSTEM rollback segment has an OPTIMAL setting, extents are de-allocated dynamically when their data is no longer needed for active transactions. The dynamic de-allocation may cause the number of currently allocated extents to be small when the SYSTEM rollback segment is frozen. Therefore, the SYSTEM rollback segment may not be large enough to handle the transactions involving the space management tables during migration.

To check the OPTIMAL setting for the SYSTEM rollback segment, issue the following SQL statement.

```
SQL> SELECT a.usn, a.name, b.optsize
       FROM v$rollname a, v$rollstat b
      WHERE a.usn = b.usn AND name = 'SYSTEM';
```

Your output should be similar to the following:

```
USN        NAME                           OPTSIZE
---------- ------------------------------ ----------
         0 SYSTEM
```

If there is a value in the OPTSIZE column, issue the following SQL statement to set optimal to NULL.

```
SQL> ALTER ROLLBACK SEGMENT system STORAGE (OPTIMAL NULL);
```

Size of SYSTEM Rollback Segment

Increase the maximum number of extents for your SYSTEM rollback segment by altering the MAXEXTENTS parameter in the STORAGE clause of the ALTER ROLLBACK SEGMENT statement.

```
SQL> ALTER ROLLBACK SEGMENT system
     STORAGE (MAXEXTENTS 249);
```

More SYSTEM rollback segment characteristics can be found with:

```
SQL> SELECT COUNT(*) FROM dba_extents
      WHERE segment_name = 'SYSTEM'
        AND segment_type = 'ROLLBACK';

SQL> SELECT EXTENTS FROM v$rollstat WHERE usn = 0;

SQL> SELECT max_extents FROM dba_rollback_segs
      WHERE segment_name = 'SYSTEM';

SQL> SELECT usn,extents,rssize,hwmsize,optsize
       FROM v$rollstat WHERE usn = 0;

USN        EXTENTS    RSSIZE     HWMSIZE    OPTSIZE
---------- ---------- ---------- ---------- ----------
         0          4     241664     241664
```

USN	Rollback Segment Number (0 = SYSTEM RS)
EXTENTS	Number of allocated Extents.
RSSIZE	Total size of the Rollback Segment in Bytes.
HWMSIZE	High water mark of Rollback Segment in Bytes
OPTSIZE	Optimal size of Rollback Segment (must be NULL for the migration from Oracle 7.3.4 to 8.1.6).

Check for any Symbolic Links

Attention: If you have any symbolic links for the database files, copy or move the database files to the real directory, or the migration will possibly fail.

```
$ find /data -type l -print
```

Check AUDIT_TRAIL

An incorrect AUDIT_TRAIL Parameter may cause the following error:

```
ORA-00604: error occurred at recursive SQL level num
ORA-01552: cannot use system rollback segment for non-system tablespace 'name'
ORA-02002: error while writing to audit trail
```

You will encounter these errors only under the following conditions:

- Either the initialization parameter AUDIT_TRAIL is set to DB or to TRUE.
- The SYS.AUD$ table is located in a tablespace other than SYSTEM.

To correct this problem, complete the following steps:

- Shutdown the database if it is open.
- Set the AUDIT_TRAIL parameter in the `init`*`SID`*`.ora` file in the following way:
  `AUDIT_TRAIL = NONE`

DROP some Views

Drop the following views in the Orcale 7 database if they exists.

```
SQL> DROP VIEW dba_histograms;
SQL> DROP VIEW v$bh;
```

Disable MTS

Disable Multithreaded Server MTS in the initSID.ora during the migration.

```
# mts_multiple_listeners = ...
# mts_listener_address   = ...
# mts_listener_address   = ...
# mts_service            = ...
# mts_dispatchers        = ...
# mts_dispatchers        = ...
# mts_max_dispatchers    = ...
# mts_servers            = ...
# mts_max_servers        = ...
```

Drop Replication

All snapshots on the replication database (SOL2) and all snapshot logs on the master database (SOL1) must be droped. After the successful migration the replication must be rebuild.

On the Replication Database

```
sqlplus sol1/sol1@SOL2
SQL> @drop_snapshotlogs.pnt
```

On the Master Database(s)

```
sqlplus sol1/sol1@SOL1
SQL> @drop_snapshots.pnt
```

Run Utility «migprep»

The *migprep* utility prepares the Oracle7 environment for migration by copying required migration files from the Oracle 8.1.6 Oracle home to the Oracle7 Oracle home. With your environment variables **pointing to Oracle 8.1.6**, run *migprep* in the following way:

```
$ . $HOME/.profile-8.1.6
$ migprep /opt/oracle/product/8.1.6 /opt/oracle/product/7.3.4
```

The utility *migprep* is a simple shell scripts which copies some files (*mig* utility, message files, NLS files, migrate.bsq) needed for the migration from the Oracle 8.1.6 Home to the Oracle 7.3.4 Home.

6. Migrate the Oracle 7 Source Database

Setup Oracle 7 Migration Environment

Set the Migration Environment you have already prepared, ORA_NLS33 must point to $ORACLE_HOME/migrate/nls/admin/data. Make sure, that the environment variable **TWO_TASK is NOT set.** Make sure, that your NLS_LANG is set to the correct value.

```
$ . $HOME/.profile-mig
```

Complete and verify Cold Backup

Stop and deactivate High Availability System, **make a full, complete Database Backup** of all Databases. No migration step will be further done without a certified database backup. **If something fails during the Oracle Migration from 7.3.4 to 8.1.6 this is the only way to migrate back to a Oracle 7.3.4 database**.

```
svrmgrl
SVRMGR> CONNECT INTERNAL;
SVRMGR> SHUTDOWN IMMEDIATE;
SVRMGR> EXIT;
```

Copy all Database-, Control-, Redolog, Configuration- (INITSID.ORA), Listener-, Password (if any) to a save place.

Stop CRON Jobs

Stop all Cron activities (Needs ROOT access)

```
$ telnet <SYSTEM-1>
$ /etc/init.d/cron stop
$ telnet <SYSTEM-2>
$ /etc/init.d/cron stop
```

Edit initSID.ora

Edit all parameter files for all databases and set the following parameters, this must be done in the Orcale 7.3.4 environment.

```
log_archive_start   =  false
audit_trail         =  none
job_queue_processes =  0
```

Make final checks on the Oracle 7 Database

Switch to NOARCHIVELOG

```
svrmgrl
SVRMGR> CONNECT INTERNAL
SVRMGR> STARTUP MOUNT EXCLUSIVE
SVRMGR> ALTER DATABASE NOARCHIVELOG;
SVRMGR> ARCHIVE LOG LIST;
```

Enable only restricted Logons

```
SVRMGR> ALTER SYSTEM ENABLE RESTRICTED SESSION;
```

Set all Tablespaces to Read-Only, except SYSTEM, TEMP and ROLLBACK.

```
SVRMGR> ALTER DATABASE OPEN;
SVRMGR> SELECT tablespace_name FROM dba_tablespaces;
SVRMGR> SELECT 'ALTER TABLESPACE '||lower(tablespace_name)||' READ ONLY;'
          FROM dba_tablespaces
         WHERE status != 'READ ONLY'
           AND tablespace_name NOT IN ('SYSTEM','RBS','TEMP');
SVRMGR> SELECT tablespace_name,status FROM dba_tablespaces;

TABLESPACE_NAME                STATUS
------------------------------ ---------
SYSTEM                         ONLINE
RBS                            ONLINE
TEMP                           ONLINE
USERS                          READ ONLY
TAB                            READ ONLY
CDR                            READ ONLY
CRE                            READ ONLY
IDX                            READ ONLY
```

Check pending Transactions

Make sure that no redo information and no uncommitted transaction are outstanding. The KTTVSTNM column of the table X$KTTVS corresponds <kttvstnm>, so you can SELECT name FROM ts$ where ts# = <kttvstnm> to find out what tablespaces have save undoto find out what tablespaces have save undo. Alter the tablespace online to eliminate the save undo. Then you can alter the tablespace READ-ONLY again.

```
SVRMGR> SELECT name
          FROM ts$ t1, x$kttvs t2
         WHERE t1.ts# = t2.kttvstnm;
```

Check the internal transaction tables for open transactions.

```
SVRMGR> SELECT * FROM sys.pending_trans$;
SVRMGR> SELECT * FROM sys.pending_sessions$;
SVRMGR> SELECT * FROM sys.pending_sub_sessions$;
SVRMGR> SELECT * FROM dba_2pc_pending;
```

All queries must retun with:

```
0 rows selected.
```

Check running Online Backup ?

Check, that we are NOT in «BEGIN BACKUP» Online Backup Mode.

```
SVRMGR> SELECT * FROM v$backup;

FILE#      STATUS             CHANGE#    TIME
---------- ------------------ ---------- ------------------
         1 NOT ACTIVE                  0 01/01/88 00:00:00
         2 NOT ACTIVE                  0 01/01/88 00:00:00
         3 NOT ACTIVE                  0 01/01/88 00:00:00
         4 NOT ACTIVE                  0 01/01/88 00:00:00
         5 NOT ACTIVE                  0 01/01/88 00:00:00
         6 NOT ACTIVE                  0 01/01/88 00:00:00
         7 NOT ACTIVE                  0 01/01/88 00:00:00
         8 NOT ACTIVE                  0 01/01/88 00:00:00
         9 NOT ACTIVE                  0 01/01/88 00:00:00
```

NOT ACTIVE meens we are NOT in Online Backup Mode !

Any Files needing a recovery ?

The view V$RECOVER_FILE shows if files needing media recovery.

```
SVRMGR> SELECT * FROM v$recover_file;

FILE#      ONLINE  ERROR              CHANGE#    TIME
---------- ------- ------------------ ---------- -----------
0 rows selected.
```

Clean Shutdown

Shutdown the Oracle7 database cleanly using the SHUTDOWN NORMAL or SHUTDOWN IMMEDIATE command; **do not use SHUTDOWN ABORT**. The Oracle7 source database must be shut down cleanly; therefore, no redo information or uncommitted transactions can remain.

```
SVRMGR> SHUTDOWN IMMEDIATE
```

Run *MIG* with CHECK_ONLY

To check disk space requirements for a successful migration, run the Oracle 8.1.6 Migration utility *MIG*, **pointing to Oracle 7.3.4,** with the CHECK_ONLY command-line option set to TRUE by entering the following at a system prompt:

```
$ . $HOME/.profile-mig
$ mig CHECK_ONLY=TRUE
```

The CHECK_ONLY command-line option causes the Migration utility to assess the amount of disk space required for migration, check the amount of space available, and issue an informational message about the disk space requirements. When the CHECK_ONLY command-line option is set to TRUE, the Migration utility does not build the Oracle 8.1.6 data dictionary or perform any other migration processing. After this check stop the database in a clean way.

```
SVRMGR> SHUTDOWN IMMEDIATE
```

Cleanup Trace-Files

Now cleanup the Tracefiles and Alertlog.

```
$ cd /var/opt/sol1
$ find admp bdmp cdmp udmp -name *.trc -exec rm {} \;
$ cd cdmp
$ :> alert_SOL1.log
```

Stop the Oracle 7.3 Listener

Now stop the Oracle 7.3.4 Listener

```
lsnrctl
LSNRCTL> stop <ListenerName>
```

Prepare Log-Files

Open two new windows and observe the logfiles during the migration.

In the first window:

```
$ cd /var/opt/sol1/bdmp
$ tail -f alert_SOL1.log
```

Start Migration Utility *MIG*

In the second window, you can now start the migration.

```
$ cd $HOME/work
$ :> migration.log
$ mig 1>migration.log 2>&1 &
$ tail -f migration.log
```

Check Log File

Check the results after running the Migration utility. The Migration utility generates informational messages and echoes its progress as it runs the *migrate.bsq* script. The Migration utility creates a convert file that contains the information of the Oracle7 control file. Later in the migration process, the convert file is used by ALTER DATABASE CONVERT to create a new control file in Oracle 8.1.6. The name of the convert file is *confSID.dbf*, the location is *$ORACLE_HOME/dbs* in the Oracle7 environment, where *SID* is the Oracle7 instance ID.

Do NOT start Oracle 7 database

Caution !

Do not open the Oracle7 database, which was shut down by the Oracle 8.1.6 Migration utility. To ensure datafile version integrity, the SCNs in the dictionary, the convert file, and file header must all be consistent when the database is converted to Oracle 8.1.6. If the Oracle7 database is opened after running the Migration utility, the SCN check will fail when the database is converted to Oracle 8.1.6, and an ORA-1211 error will be displayed, stating «Oracle7 datafile is not from migration to Oracle8».

Remove 7.3.4 Source Files

It's very important to remove or to rename the old source files for Oracle 7 now, so it's not possible to start the Oracle 7 database by mistake.

```
$ cd /opt/oracle/product
$ tar cvf 7.3.4.tar 7.3.4
$ gzip 7.3.4.tar
$ rm -r 7.3.4
```

7. Convert Oracle 7 database to Oracle 8.1.6

Setup Oracle 8.1.6 Environment

Set the Oracle 8.1.6 Environment for the «Oracle» user, which you have already prepared in the home directory for «Oracle».

```
$ cd $HOME
$ cp .profile-8.1.6 .profile
Logout and Login as Oracle
$ env
```

Check if environment is set correctly !

Activate initSID.ora for the Migration

During the Migration, some Parameters should not be activated in the *initSID.ora* file in the Oracle 8.1.6 Environment. Therefor activate the prepared ***initSIDmig.ora*** File. Be sure, that the following parameters are uncommented or set to the appropriate values:

```
# db_block_checking  = true
# db_block_checksum  = false

db_domain  =  WORLD
compatible  =  8.1.6.0

log_archive_start =  false

# mts_multiple_listeners = ....
# mts_listener_address   = ....
# mts_listener_address   = ....
# mts_service            = ....
# mts_dispatchers        = ....
# mts_dispatchers        = ....
# mts_max_dispatchers    = ....
# mts_servers            = ....
# mts_max_servers        = ....

job_queue_processes  =  0

# parallel_min_servers  = ....
# parallel_max_servers  = ....
# parallel_max_servers  = ....
# parallel_max_servers  = ....

# Sets integrity checking for tables
# event = "10210 trace name context forever, level 10"
# Sets integrity checking for indexes
# event = "10211 trace name context forever, level 10"
# Allows export to extract all data, except the corrupted blocks
# event = "10231 trace name context forever, level 10"

$ cd $ORACLE_HOME/dbs
$ cp initSIDmig.ora initSID.ora
```

Make sure, that the $ORACLE_HOME/dbs/initSID.ora points to the correct initSID.ora file, which you have prepared for the Oracle 8.1.6 database.

Example

```
$ cd $ORACLE_HOME/dbs
$ ls -l

initSOL1.ora -> /opt/oracle/config/8.1.6/initSOL1.ora
```

Move Oracle 7 Controlfiles

Either remove or rename the database's control files, or use the CONTROL_FILES initialization parameter to specify new control file names. The CONTROL_FILES initialization parameter typically is set in the *initSID.ora* file.

The ALTER DATABASE CONVERT command automatically creates new control files. If you do not use the CONTROL_FILES parameter, this command uses the control file names of your pre-migration database (derived from the CONVERT file) and returns an error if the control files already exist. Therefore, in this case, you must remove or rename the control file(s).

However, if you use the CONTROL_FILES parameter to specify new control file names, the ALTER DATABASE CONVERT command creates the new control file(s) with the names you specify, and you do not need to remove the old control files.

Control files are considerably larger in Oracle 8.1.6 than in Oracle7. For example, Oracle7 control files in the hundreds of kilobytes may expand into tens of megabytes in Oracle 8.1.6. The larger size in Oracle 8.1.6 results from the storage of more information in the control file, such as backup and tablespace records. This size increase could be important if a control file is on a raw device or if its available disk space is restricted.

```
$ cd <Location of the Oracle7 control files>
  see: CONTROL_FILES in initSID.ora
$ mv <CTRL-File-Ora7> <CTRL-File-Ora7>.old
```

Move Convert File

Move the convert file from the Oracle7 Oracle home directory to the Oracle 8.1.6 Oracle home directory. The convert file, *convSID.dbf* (where SID is the Oracle 8.1.6 database name), reside in $ORACLE_HOME/dbs in both the Oracle7 and the Oracle 8.1.6 environment.

```
$ cd /opt/oracle/product/7.3.4/dbs
$ mv convSID.ora $ORACLE_HOME/dbs
```

Convert the Oracle 7 Database to Oracle 8.1.6

Make sure all online database files are accessible and in the correct directories. You must be in the Oracle 8.1.6 Oracle home. Create a new Oracle 8.1.6 database control file and convert the file headers of all online tablespaces to Oracle 8.1.6 format.

```
$ cd $ORACLE_HOME/rdbms/admin
svrmgrl
SVRMGR> CONNECT INTERNAL
SVRMGR> STARTUP NOMOUNT;
SVRMGR> ALTER DATABASE CONVERT;
SVRMGR> ALTER DATABASE OPEN RESETLOGS;
SVRMGR> SELECT * FROM dual;
SVRMGR> SPOOL catoutm.log
SVRMGR> @u0703040.sql
SVRMGR> @catrep.sql
SVRMGR> @r0703040.sql
SVRMGR> @utlrp.sql
SVRMGR> SPOOL OFF;
SVRMGR> SHUTDOWN IMMEDIATE;
SVRMGR> STARTUP;
```

Execute additional Scripts

Execute the following additional Scripts, to perform the full accessibility to the converted Database.

```
# Parallel-Server specific views for performance queries
@$ORACLE_HOME/rdbms/admin/catparr.sql

# Collect I/O per table (actually object) statistics by
# statistical  sampling
@$ORACLE_HOME/rdbms/admin/catio.sql

# This package creates a table into which references to
# the chained rows for an IOT (Index-Only-Table) can be
# placed using the ANALYZE command.
@$ORACLE_HOME/rdbms/admin/dbmsiotc.sql

# Wrap Package which creates IOTs (Index-Only-Table)
@$ORACLE_HOME/rdbms/admin/prvtiotc.plb

# This package allows you to display the sizes of objects in the
# shared pool, and mark them for keeping or unkeeping in order to
# reduce memory fragmentation.
@$ORACLE_HOME/rdbms/admin/dbmspool.sql

# Creates the default table for storing the output
# of the ANALYZE LIST CHAINED ROWS command
@$ORACLE_HOME/rdbms/admin/utlchain.sql

# Creates the EXCEPTION table
@$ORACLE_HOME/rdbms/admin/utlexcpt.sql

# Grant public access to all views used by TKPROF
# with verbose=y option
@$ORACLE_HOME/rdbms/admin/utltkprf.sql

# Create table PLAN_TABLE that is used by the EXPLAIN PLAN
# statement. The explain statement requires the presence of this
# table in order to store the descriptions of the row sources.
@$ORACLE_HOME/rdbms/admin/utlxplan.sql

# Create performance tuning views
@$ORACLE_HOME/rdbms/admin/catperf.sql

# Create v7 style export/import views against the v8 RDBMS
# so that EXP/IMP v7 can be used to read out data in a v8 RDBMS.
# These views are necessary if you want to export from Oracle8
# and import in an Oracle7 database.
@$ORACLE_HOME/rdbms/admin/catexp7.sql

# Create views of oracle locks
@$ORACLE_HOME/rdbms/admin/catblock.sql

# Print out the lock wait-for graph in a tree structured fashion
@$ORACLE_HOME/rdbms/admin/utllockt.sql

# Creates the default table for storing the output of the
# analyze validate command on a partitioned table
@$ORACLE_HOME/rdbms/admin/utlvalid.sql

# PL/SQL Package of utility routines for raw datatypes
@$ORACLE_HOME/rdbms/admin/utlraw.sql
@$ORACLE_HOME/rdbms/admin/prvtrawb.plb

# Contains the PL/SQL interface to the cryptographic toolkit
@$ORACLE_HOME/rdbms/admin/dbmsoctk.sql
@$ORACLE_HOME/rdbms/admin/prvtoctk.plb
```

```
# This package provides a built-in random number generator. It is
# faster than generators written in PL/SQL because it calls Oracle's
# internal random number generator.
@$ORACLE_HOME/rdbms/admin/dbmsrand.sql

# DBMS package specification for Oracle8 Large Object
# This package provides routines for operations on BLOB
# and CLOB datatypes.
@$ORACLE_HOME/rdbms/admin/dbmslob.sql

# Procedures for instrumenting database applications
# DBMS_APPLICATION_INFO package spec.
@$ORACLE_HOME/rdbms/admin/dbmsapin.sql

# Recompile INVALID objects
@utlrp.sql
```

Set all Tablespaces READ WRITE

Now all Tablespaces can be set to READ WRITE again.

```
SVRMGR> SELECT tablespace_name FROM dba_tablespaces;
SVRMGR> SELECT 'ALTER TABLESPACE '||lower(tablespace_name)||' READ WRITE;'
         FROM dba_tablespaces
        WHERE status = 'READ ONLY'
          AND tablespace_name NOT IN ('SYSTEM','RBS','TEMP');
SVRMGR> SELECT tablespace_name,status FROM dba_tablespaces;

TABLESPACE_NAME                STATUS
------------------------------ ---------
SYSTEM                         ONLINE
RBS                            ONLINE
TEMP                           ONLINE
USERS                          ONLINE
TAB                            ONLINE
CDR                            ONLINE
CRE                            ONLINE
IDX                            ONLINE
```

Activate initSID.ora for Oracle 8.1.6

Now activate the real, prepared *initSID.ora* for Oracle 8.1.6, with all the necessary Parameters enabled.

Enable Online Backup again for all Instances

This must be done for all databases.

Edit initSOL1.ora as follows:

```
log_archive_start = true
```

Set database in ARCHIVELOG Mode

```
svrmgrl
SVRMGR> CONNECT INTERNAL
SVRMGR> SHUTDOWN IMMEDIATE
SVRMGR> STARTUP MOUNT
SVRMGR> ALTER DATABASE ARCHIVELOG;
SVRMGR> ARCHIVE LOG LIST;
SVRMGR> ALTER DATABASE OPEN;
SVRMGR> ALTER SYSTEM SWITCH LOGFILE;
```

Enable all Logons

After performing all migration tasks, issue the following statement to allow any user with CREATE SESSION system privilege to log on.

```
SVRMGR> ALTER SYSTEM DISABLE RESTRICTED SESSION;
```

Check ALERT.LOG for any errors !

It is very important to check the ALERT.LOG file for the specified instance. Does ARCHIVER works correctly after a log switch ? Check any TRACE files for errors.

Start the Oracle 8.1.6 Listener

Start the Net8 Listener for the migrated database.

```
lsnrctl
LSNRCTL> start <ListenerName>
```

8. Post Migration Tasks

Different Tasks are necessary after completing to Oracle 8.1.6.

Recreate BITMAP Indexes

All BITMAP Indexes should be recreated. First lookup the data dictionary for bitmap indexes using the following SQL statement:

```
sqlplus sol1/sol1@SOL1
SQL> SELECT index_name,index_type
       FROM dba_indexes
      WHERE index_type = 'BITMAP'
        AND status = 'UNUSABLE';

INDEX_NAME                     INDEX_TYPE
------------------------------ ------------
IDX_BM_ORDER_REQ_1             BITMAP
IDX_BM_CREDIT_1                BITMAP
IDX_BM_CCRSUBREQ_1             BITMAP
IDX_BM_CCRSUBPEND_1            BITMAP
IDX_BM_CCRREQUEST_1            BITMAP
```

Now recreate this bitmap indexes again.

```
SQL> set feed off;
SQL> set pagesize 10000;
SQL> set heading off;
SQL> set linesize 200;
SQL> set wrap off;
SQL> spool recreate_bitmap_indexes.sql
SQL> SELECT 'ALTER INDEX ' || index_name || ' REBUILD;'
       FROM dba_indexes
      WHERE owner = 'SOL1'
        AND index_type = 'BITMAP';
SQL> set feed on;
SQL> @recreate_bitmap_indexes.sql;
```

Check objects

Check if all objects are valid.

```
sqlplus sol1/sol1@SOL1
SQL> SELECT DISTINCT (object_type) object, status, COUNT(*)
       FROM user_objects
      GROUP BY object_type, status;
```

Check constraints

Check that all constraints are enabled.

```
SQL> SELECT c.constraint_name,
        c.table_name,
        c.column_name,
        i.r_constraint_name,
        i.status
   FROM user_cons_columns c,
        user_constraints i
  WHERE i.table_name = c.table_name
    AND i.constraint_name = c.constraint_name
    AND i.owner = c.owner
    AND i.status = 'DISABLED'
ORDER BY c.table_name,c.constraint_name,c.column_name;
```

Recompile INVALID objects

Use the script *compile_all.sql* to recompile all objects as user SYS.

```
sqlplus sys/manager@SOL1
SQL> @compile_all.sql
```

Check for Bad Date Constraints

A bad date constraint involves invalid date manipulation, which is a date manipulation that implicitly assumes the century in the date, causing problems at the year 2000. The *utlconst.sql* script runs through all of the check constraints in the database and sets constraints as bad if they include any invalid date manipulation. This script selects all the bad constraints at the end. After you run the script, the *utlresult.log* log file includes all the constraints that have invalid date constraints. The *utlconst.sql* script does not correct bad constraints, but instead disables them. You should either drop the bad constraints or recreate them after you make the necessary changes.

```
$ cd $ORACLE_HOME/rdbms/admin
svrmgrl
SVRMGR> spool utlresult.log
SVRMGR> @utlconst.sql

Server Output ON
Statement processed.
Statement processed.
 Checking for bad date constraints
 Finished checking -- All constraints OK!
```

Start CRON Jobs

Start all Cron activities (Needs ROOT access)

```
$ telnet <SYSTEM-1>
$ /etc/init.d/cron start
$ telnet <SYSTEM-2>
$ /etc/init.d/cron start
```

Reactivate DBMS Jobs

Check that DBMS Jobs works correctly.

```
sqlplus sol1/sol1@SOL1
SQL> SELECT SUBSTR(job,1,4) "Job",
     SUBSTR(schema_user,1,5) "User",
     SUBSTR(TO_CHAR(last_date,'DD.MM.YYYY HH24:MI'),1,16) "Last Date",
     SUBSTR(TO_CHAR(next_date,'DD.MM.YYYY HH24:MI'),1,16) "Next Date",
     SUBSTR(broken,1,2) "B",
     SUBSTR(failures,1,6) "Failed",
     SUBSTR(what,1,32) "Command"
     FROM dba_jobs;
```

Now run the job manually to check if every thing is ok, connect as the user listed above and execute the following Statement. If there is an error, lookup the file alertSID.log or the generated Tracefile for more details.

```
SQL> EXECUTE DBMS_JOB.RUN(<Enter Job-Nr from above>);
```

Change the Password for the OUTLN User

The OUTLN user is created automatically during installation of Oracle 8.1.6. This user has DBA privileges. Use the ALTER USER command to change the password for this user. Oracle 8.1.6 adds the OUTLN user schema to support Plan Stability. The OUTLN user acts as a place to centrally manage metadata associated with stored outlines.

```
sqlplus sys/manager@SOL1
SQL> ALTER USER outln IDENTIFIED BY <password>;
```

Check new Environment in all $HOME/.profiles

Check, that $ORACLE_HOME, $TNS_ADMIN and all the other Environment Variables point to the correct location.

```
cd /export/home
find . -exec grep -l '7.3.4' {} \;
```

Check Redolog and DB-Files

Check, if your Redolog Files are big enough (15-20 MB) and if there are about 6-10 Groups. Then force some Logswitches.

```
sqlplus sys/manager@SOL1
SQL> SELECT * FROM v$log;
SQL> SELECT * FROM v$logfile ORDER BY group#;
SQL> SELECT file#,ts#,status,enabled
     FROM v$datafile
     ORDER BY file#;
```

Update your HA startup / shutdown scripts

If you use the High Availability System HA, you must update your scripts. Usually the only thing you have to do is the replacment of the string 7.3.4 with 8.1.6 in the following scripts on both systems.

Reset Passwords for SYSTEM ans SYS

Reset Passwords for the Users SYSTEM and SYS. Do NOT leave them as the very common and known value «MANAGER».

```
svrmgrl
SVRMGR> CONNECT INTERNAL
SVRMGR> ALTER USER sys IDENTIFIED BY ....;
SVRMGR> ALTER USER system IDENTIFIED BY ....;
```

9. Troubleshooting

INVALID Objects in SYS Schema

If you have any INVALID objects in the schema SYS or SYSTEM after the whole migration, then restart the Procedure *catproc.sql* and possibly *catrep.sql* for Replication Packages in *$ORACLE_HOME/rdbms/admin*. This Procedures recreate most of the needed packages. However, this will last about 15 to 30 minutes.

```
cd $ORACLE_HOME/rdbms/admin
svrmgrl
SVRMGR> CONNECT INTERNAL
SVRMGR> @catproc.sql;
SVRMGR> @catrep.sql;
SVRMGR> @utlrp.sql

sqlplus sys/manager@SOL1
SQL> start compile_all
SQL> start utlrp_simple.sql
```

You should not find any objects if you execute the following Statements:

```
sqlplus sys/manager@SOL1
SQL> SELECT owner, object_name,
       TO_CHAR(created,'DD.MM.YYYY:HH24:MI:SS'), status
      FROM dba_objects
     WHERE status != 'VALID';

no rows selected

SQL> SELECT COUNT(*)
 FROM obj$ o, user$ u
WHERE u.user# = o.owner#
  AND o.remoteowner is NULL
  AND o.status IN (4,5,6)
  AND o.type# in (4, 7, 8, 9, 11, 12, 13, 14);

 COUNT(*)
---------
         0
```

utlrp.sql

This Utility from Oracle recompiles invalid PL/SQL modules.

Description (Oracle):

This is a fairly general script that can be used at any time to recompile all existing invalid PL/SQL modules in a database.

If run as one of the last steps during migration/upgrade/downgrade (see the README notes for your current release and the Oracle Migration book), this script will validate all PL/SQL modules (procedures, functions, packages, triggers, types, views) during the migration step itself.

Although invalid PL/SQL modules get automatically recompiled on use, it is useful to run this script ahead of time (e.g. as one of the last steps in your migration), since this will either eliminate or minimize subsequent latencies caused due to on-demand automatic recompilation at runtime.

Oracle highly recommends running this script towards the end of of any migration/upgrade/downgrade.