Oracle Database 10*g* Migration to Automatic Storage Management

An Oracle White Paper
August 2005

ORACLE

Oracle Database 10*g* Migration to Automatic Storage Management

Oracle Database 10*g* Migration to Automatic Storage Management

INTRODUCTION

In Oracle Database 10*g*, a new feature called Automatic Storage Management (ASM) simplifies storage management and provisioning for the database. ASM provides file system and volume manager capabilities as a built-in feature of the database. With this capability, ASM simplifies storage management tasks, such as managing the underlying database files during database creation and disk space management when the database is in use. Since ASM allows disk management to be done using familiar SQL statements such as CREATE, ALTER, and DROP, database administrators (DBAs) do not need to learn a new skill set or make crucial decisions about provisioning.

ASM is optimized for Oracle files. DBAs can use ASM on a piecemeal basis by allocating new Oracle database files into ASM as the database grows. However, to get all of the benefits of ASM, the entire database should be migrated to ASM. For example, when the entire database is in ASM, DBAs can add storage to or remove storage from the database configuration with automated balancing of the distribution of the database files without downtime.

This whitepaper describes two methods of migrating an existing Oracle database from legacy storage to ASM. This whitepaper addresses both primary databases and standby databases.

DATABASE MIGRATION TO ASM OVERVIEW

There are two migration methods, hot migration and cold migration, described in this whitepaper. Both of these migration methods use Recovery Manager (RMAN) because only RMAN can back up, restore, and recover a database that is on an ASM disk group. Consider the following factors when deciding which migration method to use:

- The amount of storage capacity available
- The amount of database downtime that can be tolerated

It is possible to migrate an existing primary database to ASM by issuing a Data Guard switchover command to a Redo Apply database to reduce downtime. This method is outside the scope of this white paper.

Hot Migration Overview

Use the hot migration method when there is sufficient, unallocated disk space that can be used for ASM. The amount of disk space required for hot migration depends on the backup strategy and the amount of disk space used by disk-based backups. The minimum amount of unallocated disk space required is equivalent to the size of the database. The hot migration method consists of two phases:

- The preparation phase builds and configures ASM storage while the database remains online.
- The short outage phase switches the database to ASM storage while the database is offline.

Cold Migration Overview

Use the cold migration method when there is insufficient unallocated disk space available to hold a full copy of the database. The cold migration method requires longer downtime than the hot migration method. The cold migration method consists of two phases:

- The first phase migrates the flash recovery area, which contains recovery-related files such as archived redo logs and database backups.
- The second phase migrates the data and the redo log files.

Both phases of the cold migration may be completed during a single maintenance window or spread over the course of multiple days or even weeks.

Choosing a Migration Method

Use the following table to determine the migration method you should use based on the amount of disk space available for the migration and the outage time allotted. The database size is the sum of the sizes of the data files, temporary files, control files, and redo log files. The flash recovery area utilization can be determined by querying V$RECOVERY_FILE_DEST.

Disk Space Available for Migration	**Downtime Allotted**	**Migration Method**
More than database size plus flash recovery area contents	Minimal – single outage only	Hot migration
More than database size plus flash recovery area contents	Multiple outages allowed	Hot migration or cold migration
Less than database size plus flash recovery area contents	Minimal – single outage only	N/A *

Less than database size plus flash recovery area contents	Multiple outages allowed	Cold migration

* - A minimal outage migration to ASM cannot be completed without sufficient disk space available.

Hot Migration

Hot migration is possible when there is enough disk space available during the course of the migration.

Hot Migration with Additional Storage

Hot migration with additional storage requires the permanent allocation of additional disk space. This method is ideal when the added storage contains the database, and the current storage remains to hold disk backups.

Hot Migration with Current Storage

Hot migration with current storage uses temporary additional storage. This method can be used when the newly allocated disk space is available for a limited amount of time, and the ultimate goal is to switch back to the original storage. This method requires additional steps, which are documented in Appendix 1 of the white paper.

Cold Migration

Cold migration is required when there is not enough disk space to make an ASM disk group large enough to contain a copy of the database and the flash recovery area files during the course of the migration. During cold migration, RMAN can use disk or tape for the backup.

Cold Migration using Disk

Cold migration using disk requires a full RMAN backup in the flash recovery area. This backup reduces the amount of time required to restore the database during the second outage.

Cold Migration using Tape

Cold migration using tape is the slowest migration method and requires tape devices to hold the backup of the database. The method described in this white paper assumes that the default device type for backups has been set to tape using the RMAN CONFIGURE command.

DATABASE MIGRATION TO ASM

This section describes migrating an Oracle Database 10*g* database from UNIX file systems to ASM. In the examples in this section, the database is a single instance database running on the Redhat Linux operating system, but the procedure is identical when migrating from a cluster file system or RAW partitions, regardless of the operating system.

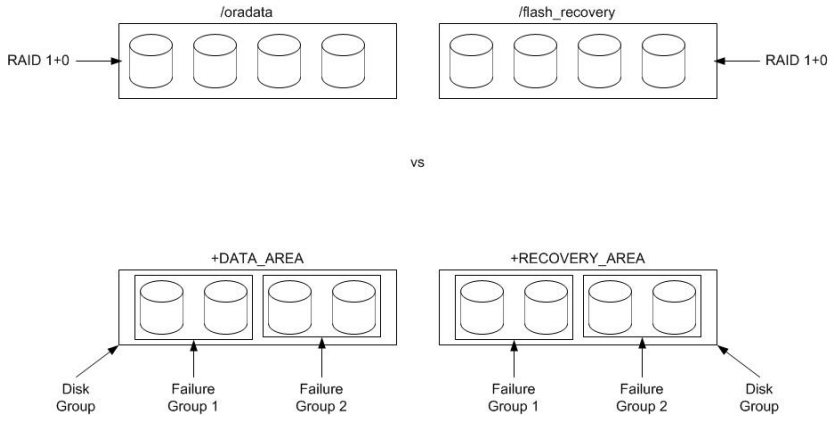
The same procedure can be used to migrate a primary database or a logical standby database, but a different procedure migrates a physical standby database. Both procedures are documented in this white paper.

Assumptions

This white paper makes the following assumptions about the environment that is being migrated:

- The database uses Oracle Managed Files (OMF) by setting the DB_CREATE_FILE_DEST and DB_RECOVERY_FILE_DEST initialization parameters.
- The database area specified by the DB_CREATE_FILE_DEST initialization parameter resides on the UNIX file system in the /oradata directory and is striped and mirrored over four individual disks.
- The flash recovery area specified by the DB_RECOVERY_FILE_DEST initialization parameter for the database resides on the Unix file system in the /flash_recovery directory, and is striped and mirrored over 4 individual disks.
- The database has at least two redo log members for each group: one on the /oradata file system and the other on the /flash_recovery file system.
- The database has at least two control files: one on the /oradata file system and the other on the /flash_recovery file system.
- There are eight additional disks available for the Hot Migration.

The following illustration shows the disk layout before and after the migration.



Follow the instructions in the appropriate section:

Migrating a Primary or Logical Standby Database to ASM
Migrating a Physical Standby Database to ASM

MIGRATING A PRIMARY OR LOGICAL STANDBY DATABASE TO ASM

Since the standby database is shutdown during the course of the migration, the protection level cannot be maximum protection mode unless there is another standby database available that supports maximum protection mode.

The following procedures describe migrating a primary database to ASM, but the procedure also applies to a logical standby database.

Follow the instructions in the appropriate section:

Primary Database – Hot Migration to ASM
Primary Database – Cold Migration to ASM

Primary Database – Hot Migration to ASM

There are three phases in the process of migrating a database from UNIX file systems to ASM.

Hot Migration - Before Migration
Hot Migration - The Switch
Hot Migration - After Migration

Optionally, if the additional storage required by the hot migration needs to be removed, perform the tasks described in "Appendix 1: Migrating ASM Disk Groups to Original Database Storage."

Hot Migration - Before Migration

During this phase of the migration, there is no outage to the primary database. Complete the following steps:

Step 1: Create the Initialization Parameter File for the ASM Instance
Step 2: Create the Server Parameter File for the ASM Instance
Step 3: Start the ASM Instance
Step 4: Prepare the Disks for Use by ASM
Step 5: Change the Permissions on the Disk Device Files
Step 6: Create the ASM Disk Groups
Step 7: Prepare the Primary Database for ASM Disk Group Usage
Step 8: Migrate the Current RMAN Backups to the Flash Recovery Area

Primary Database – Hot Migration to ASM

Step 9: Migrate the Block Change Tracking File to the Data Area
Step 10: Make the Initial Copy of the Oracle Data Files
Step 11: Migrate the Oracle Redo Log and Standby Redo Log Files to ASM Disk Groups
Step 12: Migrate the Temporary Files to ASM Disk Groups
Step 13: Refresh the Previous Copy of the Oracle Data Files

Step 1: Create the Initialization Parameter File for the ASM Instance

The ASM instance has its own initialization parameter file. A typical ASM initialization parameter file has the following parameter settings:

```
*.instance_type='asm'
*.remote_login_passwordfile='SHARED'
*.large_pool_size=12M
*.asm_power_limit=10
*.background_dump_dest='/u01/app/oracle/admin/+ASM/bdump'
*.core_dump_dest='/u01/app/oracle/admin/+ASM/cdump'
*.user_dump_dest='/u01/app/oracle/admin/+ASM/udump'
*.asm_diskstring='/dev/raw/raw*'
```

The Database Configuration Assistant (DBCA) defaults the Oracle SID for the ASM instance to '+ASM' for a single instance and '+ASM*n*' for RAC instances.

Step 2: Create the Server Parameter File for the ASM Instance

It is considered best practice to create a server parameter file (SPFILE) for the ASM instance so that the ASM_DISKGROUPS initialization parameter is updated automatically when a new disk group is created. With this parameter set, the proper ASM disk groups are mounted when the ASM instance is started. Create the SPFILE using the following commands:

```
$ export ORACLE_SID=+ASM
$ sqlplus "/as sysdba"
Connected to an idle instance.

SQL> CREATE SPFILE FROM PFILE;

File created.
```

Step 3: Start the ASM Instance

Start the ASM instance with the NOMOUNT option in the same way that any other Oracle instance can be started.

```
SQL> STARTUP NOMOUNT
ASM instance started

Total System Global Area  100663296 bytes
Fixed Size                   783672 bytes
Variable Size              99879624 bytes
Database Buffers                  0 bytes
Redo Buffers                      0 bytes
```

Note: If you are performing a cold migration of a primary or logical standby database, then proceed to "Step 2: Save Backups to Tape" on page 15. Otherwise, continue with the next step in this procedure.

Primary Database – Hot Migration to ASM

Step 4: Prepare the Disks for Use by ASM

This step varies depending on the operating system. For example, to prepare the disks on Linux, remove the device on which the original file system was built, stop the RAID device, and create a RAW device created on the disks over the block device.

Step 5: Change the Permissions on the Disk Device Files

The ASM instance runs as the Oracle user. Therefore, change the permissions on the disk device file for each disk that will be used by ASM to give the ASM instance has write access. For example, the following commands change the permissions on a disk device file:

```
# chown oracle:dba /dev/raw/raw_device
# chmod 640 /dev/raw/raw_device
```

Replace *raw_device* with the name of the device.

Note: If you are performing a cold migration of a primary or logical standby database, then proceed to "Step 4: Create the Flash Recovery Area Disk Group" on page 18 or return to "Step 6: Prepare the Disk for ASM Usage" on page 21, depending on your current progress with that operation. Otherwise, continue with the next step in this procedure.

Step 6: Create the ASM Disk Groups

Create the two ASM disk groups for the data area and flash recovery area. The following example creates two disk groups with normal redundancy.

```
SQL> CREATE DISKGROUP data_area NORMAL REDUNDANCY
FAILGROUP controller1 DISK '/dev/raw/raw9','/dev/raw/raw10'
FAILGROUP controller2 DISK '/dev/raw/raw11','/dev/raw/raw12';

SQL> CREATE DISKGROUP recovery_area normal REDUNDANCY
FAILGROUP controller1 DISK '/dev/raw/raw13','/dev/raw/raw14'
FAILGROUP controller2 DISK '/dev/raw/raw15','/dev/raw/raw16';
```

If you are migrating a logical standby database, then the following statements are also required.

```
SQL> ALTER DISKGROUP recovery_area ADD DIRECTORY
'+RECOVERY_AREA/ORCL';
SQL> ALTER DISKGROUP recovery_area ADD DIRECTORY
'+RECOVERY_AREA/ORCL/ARCHIVELOG';
SQL> ALTER DISKGROUP recovery_area ADD DIRECTORY
'+RECOVERY_AREA/ORCL/ARCHIVELOG/FAL';
SQL> ALTER DISKGROUP recovery_area ADD DIRECTORY
'+RECOVERY_AREA/ORCL/ARCHIVELOG/SRL';
```

These additional statements are not required for a primary database.

The destinations for the STANDBY_ARCHIVE_DEST and LOG_ARCHIVE_DEST_*n* parameters for the database's standby redo log files must be created prior to the start of the Oracle instance.

Step 7: Prepare the Primary Database for ASM Disk Group Usage

Change the DB_CREATE_FILE_DEST and DB_RECOVERY_FILE_DEST initialization parameters to point to the data area and flash recovery area disk groups, respectively.

```
SQL> ALTER SYSTEM SET DB_CREATE_FILE_DEST='+DATA_AREA' SCOPE=BOTH;
```

```
SQL> ALTER SYSTEM SET DB_RECOVERY_FILE_DEST='+RECOVERY_AREA'
SCOPE=BOTH;
```

Step 7 assumes that initialization parameter LOG_ARCHIVE_DEST_*n* is configured to archive the standby redo log files locally.

If you are migrating a logical standby database, then change the LOG_ARCHIVE_DEST_*n* and STANDBY_ARCHIVE_DEST initialization parameters to point to the flash recovery area disk group.

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_n=
'+RECOVERY_AREA/ORCL/ARCHIVELOG/SRL/
VALID_FOR=(STANDBY_LOGFILES,STANDBY_ROLE) DB_UNIQUE_NAME=ORCL'
SCOPE=SPFILE;
SQL> ALTER SYSTEM SET STANDBY_ARCHIVE_DEST=
'+RECOVERY_AREA/ORCL/ARCHIVELOG/FAL/' SCOPE=SPFILE;
```

Replace *n* in LOG_ARCHIVE_DEST_*n* with the appropriate value. These additional statements are not required for a primary database.

Step 8: Migrate the Current RMAN Backups to the Flash Recovery Area

Move the current backup sets, data file copies, and archived log files to the ASM disk groups by running the following RMAN commands.

```
RMAN> BACKUP BACKUPSET ALL DELETE INPUT;
RMAN> BACKUP AS COPY DATAFILECOPY ALL DELETE INPUT;
RMAN> BACKUP AS COPY ARCHIVELOG ALL DELETE INPUT;
```

Step 9: Migrate the Block Change Tracking File to the Data Area

If block change tracking has been enabled previously, then recreate the file in ASM.

```
SQL> ALTER DATABASE DISABLE BLOCK CHANGE TRACKING;
SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;
```

If block change tracking is not enabled, then enabled it for the duration of the ASM migration.

```
SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;
```

Step 10: Make the Initial Copy of the Oracle Data Files

Using RMAN, back up the database using the AS COPY option.

```
RMAN> BACKUP DEVICE TYPE DISK INCREMENTAL LEVEL 0 AS COPY TAG
'ASM_Migration' DATABASE FORMAT '+DATA_AREA';
```

Note: If you are performing a hot migration of a physical standby database, then proceed to "Step 2: Migrate the Oracle Redo Log and Standby Redo Log Files to ASM Disk Groups" on page 24. Otherwise, continue with the next step in this procedure.

Step 11: Migrate the Oracle Redo Log and Standby Redo Log Files to ASM Disk Groups

In order to delete the logfile members, log switches might be required. Additionally, for SQL Apply and standby log files, log groups might need to be cleared, and log transport services might need to be deferred temporarily.

First, identify the current redo log members by running the following query:

```
SQL> COLUMN GROUP# HEADING 'Group' FORMAT 99999
COLUMN MEMBER HEADING 'Member' FORMAT A60

SELECT GROUP#, MEMBER FROM V$LOGFILE ORDER BY GROUP#;

Group  Member
------ ------------------------------------------------------------
     1 /oradata/ORCL/onlinelog/o1_mf_1_0fs38tdh_.log
     1 /flash_recovery/ORCL/onlinelog/o1_mf_1_0fs38tyq_.log
     2 /oradata/ORCL/onlinelog/o1_mf_2_0fs38vmw_.log
```

```
2 /flash_recovery/ORCL/onlinelog/o1_mf_2_0fs393bj_.log
3 /oradata/ORCL/onlinelog/o1_mf_3_0fs3942r_.log
3 /flash_recovery/ORCL/onlinelog/o1_mf_3_0fs39c12_.log
```

It might be necessary to stop SQL Apply temporarily during this step.

Next, for each redo log group, drop one of the two current redo log members, and add the two new redo log members before finally dropping the other original redo log member. For example, for group 1 in the query example, run the following statements:

```
SQL> ALTER DATABASE DROP LOGFILE MEMBER
'/flash_recovery/ORCL/onlinelog/o1_mf_1_0fs38tyq_.log';
SQL> ALTER DATABASE ADD LOGFILE MEMBER
'+DATA_AREA','+RECOVERY_AREA' TO GROUP 1;
SQL> ALTER DATABASE DROP LOGFILE MEMBER
'/oradata/ORCL/onlinelog/o1_mf_1_0fs38tdh_.log';
```

Step 12: Migrate the Temporary Files to ASM Disk Groups

It might be necessary to stop SQL Apply temporarily during this step.

Identify each current temporary file and its size, and add an equivalent temporary file of the same size before removing the original temporary file. Perform this action for each temporary file.

For example, to identify each current temporary file, run the following query:

```
SQL> SELECT BYTES, NAME FROM V$TEMPFILE;

     BYTES NAME
---------- ----------------------------------------------------------
  20971520 /oradata/ORCL/datafile/o1_mf_temp_0fs3bq8w_.tmp
```

The example query returned one temporary file that is approximately 20 megabytes in size. In this case, run the following statements to create a new temporary file and drop the original temporary file.

```
SQL> ALTER TABLESPACE temp ADD TEMPFILE SIZE 20M;
SQL> ALTER DATABASE TEMPFILE
'/oradata/ORCL/datafile/o1_mf_temp_0fs3bq8w_.tmp' DROP;
```

This step identifies those blocks that were modified during the course of the previous incremental backup and can be repeated in the run up to the database outage. By repeatedly executing this step, the outage described in the "Hot Migration – The Switch" section would need to recover only one or two archived log files, reducing the outage window.

Step 13: Refresh the Previous Copy of the Oracle Data Files

Using RMAN, roll forward the data file copies in the DATA_AREA by creating an incremental level 1 backup of the database and merging this backup with the earlier level 0 backup.

```
RMAN> RUN {
BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY WITH TAG
'ASM_Migration' database;
RECOVER COPY OF DATABASE WITH TAG 'ASM_Migration';
}
```

Note: If you are performing a hot migration of a physical standby database, then proceed to "Step 1: Prepare the ASM Disk Groups to Store the Control Files" on page 25. Otherwise, continue with the next section in this procedure.

Hot Migration - The Switch

This phase is the start of the outage. Database downtime is determined by the length of time it takes to complete the steps in this section.

Step 1: Prepare the ASM Disk Groups to Store the Control Files
Step 2: Switch the Database Data Files to ASM Disk Groups
Step 3: Move the Flashback Database Log Files

Step 1: Prepare the ASM Disk Groups to Store the Control Files

Ensure that the directory structure exists in the new ASM disk groups for the control files by creating control file backups in both disk groups, and update the CONTROL_FILES parameter in the SPFILE.

```
SQL> ALTER DATABASE BACKUP CONTROLFILE TO '+DATA_AREA';
SQL> ALTER DATABASE BACKUP CONTROLFILE TO '+RECOVERY_AREA';
SQL> ALTER SYSTEM SET CONTROL_FILES =
'+DATA_AREA/ORCL/CONTROLFILE/mycontrol.ctl',
'+RECOVERY_AREA/ORCL/CONTROLFILE/mycontrol.ctl' SCOPE=SPFILE;
```

Determine the current control file names by running the following query:

```
SQL> SELECT NAME FROM V$CONTROLFILE;

NAME
---------------------------------------------------------------------
/oradata/ORCL/controlfile/o1_mf_0fs38sx3_.ctl
/flash_recovery/ORCL/controlfile/o1_mf_0fs38t2w_.ctl
```

Step 2: Switch the Database Data Files to ASM Disk Groups

To prepare for restoring one of the control files into the new control file locations, shut down the database cleanly, restart the instance to a nomount state, and mount the database.

```
RMAN> SHUTDOWN IMMEDIATE;
RMAN> STARTUP NOMOUNT
RMAN> RESTORE CONTROLFILE FROM
'/oradata/ORCL/controlfile/o1_mf_0fs38sx3_.ctl';
RMAN> ALTER DATABASE MOUNT;
```

If you are migrating a logical standby database, then use RMAN to create an incremental level 1 backup of the database, and restore the backup into the database copy with the database mounted.

```
RMAN> RUN {
BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY WITH TAG
'ASM_Migration' DATABASE;
RECOVER COPY OF DATABASE WITH TAG 'ASM_Migration';
}
```

Switch the database to use the backup data file copies created previously, and recover any changes that might have occurred between the last incremental backup and the database being shut down.

```
RMAN> SWITCH DATABASE TO COPY;
RMAN> RECOVER DATABASE;
```

These additional statements are not required for a primary database.

Primary Database – Hot Migration to ASM

Step 3: Move the Flashback Database Log Files

If Flashback Database is enabled, then disable it. Next, enable Flashback Database to re-create the flashback log files in the ASM flash recovery area disk group, and open the database.

For a SQL Apply database, use ALTER DATABASE OPEN NORESETLOGS.

```
SQL> ALTER DATABASE FLASHBACK OFF;
SQL> ALTER DATABASE FLASHBACK ON;
SQL> ALTER DATABASE OPEN;
```

If you are migrating a logical standby database, then start logical standby apply:

```
SQL> ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;
```

This additional statement is not required for a primary database.

Hot Migration - After Migration

To complete the migration, remove the block change-tracking file and validate that all files have been moved to the ASM disk groups.

Step 1: Disable Block Change Tracking
Step 2: Verify File Placement

Step 1: Disable Block Change Tracking

If block change tracking was enabled during the migration, then disable it.

```
SQL> ALTER DATABASE DISABLE BLOCK CHANGE TRACKING;
```

Step 2: Verify File Placement

Run the following query and make sure all files reside in either the DATA_AREA or RECOVERY_AREA ASM disk group.

```
SQL> SELECT NAME FROM V$CONTROLFILE
UNION
SELECT NAME FROM V$DATAFILE
UNION
SELECT NAME FROM V$TEMPFILE
UNION
SELECT MEMBER FROM V$LOGFILE
UNION
SELECT FILENAME FROM V$BLOCK_CHANGE_TRACKING
UNION
SELECT NAME FROM V$FLASHBACK_DATABASE_LOGFILE;

NAME
----------------------------------------------------------------------
+DATA_AREA/orcl/changetracking/ctf.256.1
+DATA_AREA/orcl/controlfile/mycontrol.ctl
+DATA_AREA/orcl/datafile/sysaux.259.1
+DATA_AREA/orcl/datafile/system.257.1
+DATA_AREA/orcl/datafile/undotbs1.258.1
+DATA_AREA/orcl/datafile/users.260.1
+DATA_AREA/orcl/onlinelog/group_1.263.1
+DATA_AREA/orcl/onlinelog/group_2.264.1
+DATA_AREA/orcl/onlinelog/group_3.265.1
+DATA_AREA/orcl/tempfile/temp.266.1
+RECOVERY_AREA/orcl/controlfile/mycontrol.ctl
+RECOVERY_AREA/orcl/flashback/log_1.276.1
+RECOVERY_AREA/orcl/onlinelog/group_1.265.1
+RECOVERY_AREA/orcl/onlinelog/group_2.266.1
+RECOVERY_AREA/orcl/onlinelog/group_3.267.1
```

Primary Database – Hot Migration to ASM

Step 3: Optionally Migrate the ASM Disk Groups Back to the Original Database Storage

Optionally, if the additional storage required by the hot migration must be removed, then complete the procedure in "Appendix 1: Migrating ASM Disk Groups to Original Database Storage" starting on page 33.

Primary Database – Cold Migration to ASM

There are five phases in the process of migrating a database from UNIX file systems to ASM.

Cold Migration – Before Migration
Cold Migration – First Outage – Flash Recovery Area Moves to ASM
Cold Migration - Database Backup
Cold Migration - Second Outage – Data Area Moves to ASM
Cold Migration - After Migration

Cold Migration – Before Migration

During this phase of the migration, there is no database outage.

Step 1: Create the ASM Instance
Step 2: Save Backups to Tape

Step 1: Create the ASM Instance

Perform "Step 1: Create the Initialization Parameter File for the ASM Instance" through "Step 3: Start the ASM Instance" on page 8 to create the ASM instance.

Step 2: Save Backups to Tape

Existing backups are not used to aid the migration of the database to ASM. However, it is strongly recommended that these backups be saved to an alternate location such as tape, because the contents of the current flash recovery area will be destroyed during this migration.

Prior to the start of the first outage, any backups that must be saved should be copied to tape or to an alternate location. The following example assumes that RMAN is being used to transfer the backups to tape.

Save current backup sets, data file copies, and archived log files to tape.

```
RMAN> BACKUP DEVICE TYPE SBT BACKUPSET ALL DELETE INPUT;
RMAN> BACKUP DEVICE TYPE SBT DATAFILECOPY ALL DELETE INPUT;
RMAN> BACKUP DEVICE TYPE SBT ARCHIVELOG ALL DELETE INPUT;
```

Note: If you are performing a cold migration of a physical standby database, then proceed to "Step 1: Clear the Old Flash Recovery Area" on page 28. Otherwise, continue with the next section in this procedure.

Cold Migration – First Outage – Flash Recovery Area Moves to ASM

During this phase, the database must be shut down so that the storage currently used for the flash recovery area can be reformatted for use by ASM.

Step 1: Prepare Standby Database for ASM Disk Group Usage (Logical Standby Database Only)
Step 2: Clear the Old Flash Recovery Area
Step 3: Prepare the Disks for Use by ASM
Step 4: Create the Flash Recovery Area Disk Group
Step 5: Prepare the Primary Database to Use the ASM Disk Group
Step 6: Restart SQL Apply (Logical Standby Database Only)

Step 1: Prepare Standby Database for ASM Disk Group Usage (Logical Standby Database Only)

Decide on a location where the archived redo log files can be temporarily stored for the SQL Apply database. For this example, /tmp/ is the assumed location.

This step is required only when a logical standby database is being migrated to ASM. If a primary database is being migrated to ASM, then this step is not required, and you can proceed to "Step 2: Clear the Old Flash Recovery Area".

Change the settings for the LOG_ARCHIVE_DEST_*n* and STANDBY_ARCHIVE_DEST initialization parameters that archive the standby redo log files on the logical standby database. These parameters should specify a temporary location during the migration process.

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_n='LOCATION=/tmp/
VALID_FOR=(STANDBY_LOGFILES,STANDBY_ROLE) DB_UNIQUE_NAME=ORCL'
SCOPE=both;
SQL> ALTER SYSTEM SET STANDBY_ARCHIVE_DEST='/tmp/' SCOPE=both;
```

Replace *n* in LOG_ARCHIVE_DEST_*n* with the appropriate value.

On the primary database, force one or more log switches, and when the files have been transferred successfully, stop the log transport service.

```
SQL> ALTER SYSTEM SWITCH LOGFILE;
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_n=defer;
```

Replace *n* in LOG_ARCHIVE_DEST_STATE_*n* with the appropriate value.

On the logical standby database, stop SQL Apply, change the hidden SQL Apply parameter, and restart SQL Apply.

```
SQL> ALTER DATABASE STOP LOGICAL STANDBY APPLY;
SQL> EXECUTE DBMS_LOGSTDBY.APPLY_SET('_max_log_lookback',0.1);
SQL> ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;
```

Run the following query on the logical standby database.

```
SQL> SELECT FILE_NAME
FROM DBA_LOGSTDBY_LOG dll, DBA_LOGSTDBY_PROGRESS dlp
WHERE dll.NEXT_CHANGE# >= dlp.READ_SCN;
```

Primary Database – Cold Migration to ASM

When the only file names returned reside in the temporary location or no records are returned, stop SQL Apply.

```
SQL> ALTER DATABASE STOP LOGICAL STANDBY APPLY;
```

Step 2: Clear the Old Flash Recovery Area

Remove the redo log members from the /flash_recovery file system. To determine the redo log member files residing in the /flash_recovery file system, you can query the V$LOGFILE view. Drop the redo log member for both online and standby redo log files. For example, the following statement drops a redo log member.

```
SQL> ALTER DATABASE DROP LOGFILE MEMBER
'/flash_recovery/ORCL/onlinelog/o1_mf_1_0fpqygx6_.log';
```

Shut down the database, remount the instance, and remove any archived log files that might have been generated.

```
RMAN> SHUTDOWN IMMEDIATE;
RMAN> STARTUP MOUNT
RMAN> BACKUP DEVICE TYPE SBT ARCHIVELOG ALL DELETE INPUT;
```

If flashback database is enabled, then disable it.

```
SQL> ALTER DATABASE FLASHBACK OFF;
```

Query the V$CONTROLFILE view, and redefine the CONTROL_FILE initialization parameter in the SPFILE so that the database will not reference the /flash_recovery file system copy.

```
SQL> SELECT NAME FROM V$CONTROLFILE;

NAME
------------------------------------------------------------
/oradata/ORCL/controlfile/o1_mf_0fpqyfw7_.ctl
/flash_recovery/ORCL/controlfile/o1_mf_0fpqyg20_.ctl

SQL> ALTER SYSTEM SET CONTROL_FILES=
'/oradata/ORCL/controlfile/o1_mf_0fpqyfw7_.ctl' SCOPE=spfile;
```

Change the DB_RECOVERY_FILE_DEST initialization parameter to point to the RECOVERY_AREA disk group.

```
SQL> ALTER SYSTEM SET DB_RECOVERY_FILE_DEST='+RECOVERY_AREA'
SCOPE=spfile;
```

If you are migrating a logical standby database, then change the STANDBY_ARCHIVE and LOG_ARCHIVE_DEST_*n* initialization parameters to point to the RECOVERY_AREA disk group.

```
SQL> ALTER SYSTEM SET STANDBY_ARCHIVE_DEST=
'+RECOVERY_AREA/ORCL/ARCHIVELOG/FAL' SCOPE=spfile;

SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_n=
'location=+RECOVERY_AREA/ORCL/ARCHIVELOG/SRL/
VALID_FOR=(STANDBY_LOGFILES,STANDBY_ROLE) DB_UNIQUE_NAME=ORCL'
SCOPE=spfile;
```

Replace *n* in LOG_ARCHIVE_DEST_*n* with the appropriate value.

Shut down the database, and unmount the flash recovery area file system.

```
SQL> SHUTDOWN IMMEDIATE;
# umount /flash_recovery
```

Primary Database – Cold Migration to ASM

Step 3: Prepare the Disks for Use by ASM

Perform "Step 4: Prepare the Disks for Use by ASM" and "Step 5: Change the Permissions on the Disk Device Files" starting on page 9.

Step 4: Create the Flash Recovery Area Disk Group

Create the ASM disk group for the flash recovery area. The following example creates a disk group with normal redundancy and two failure groups.

```
SQL> CREATE DISKGROUP RECOVERY_AREA NORMAL REDUNDANCY
FAILGROUP controller1 DISK '/dev/raw/raw13','/dev/raw/raw14'
FAILGROUP controller2 DISK '/dev/raw/raw15','/dev/raw/raw16';
```

If you are migrating a logical standby database, then the following statements are also required.

```
SQL> ALTER DISKGROUP RECOVERY_AREA ADD DIRECTORY
'+RECOVERY_AREA/ORCL';
SQL> ALTER DISKGROUP RECOVERY_AREA ADD DIRECTORY
'+RECOVERY_AREA/ORCL/ARCHIVELOG';
SQL> ALTER DISKGROUP RECOVERY_AREA ADD DIRECTORY
'+RECOVERY_AREA/ORCL/ARCHIVELOG/FAL';
SQL> ALTER DISKGROUP RECOVERY_AREA ADD DIRECTORY
'+RECOVERY_AREA/ORCL/ARCHIVELOG/SRL';
```

Note: If you are performing a cold migration of a physical standby database, then proceed to "Step 2: Prepare the Standby Database to Use the ASM Disk Group" on page 28. Otherwise, continue with the next step in this procedure.

Step 5: Prepare the Primary Database to Use the ASM Disk Group

Mount the database instance, reenable flashback database if required, and reestablish the redo log file members and standby redo log file members for all the redo log groups.

```
SQL> STARTUP MOUNT
SQL> ALTER DATABASE FLASHBACK ON;
SQL> ALTER DATABASE ADD LOGFILE MEMBER '+RECOVERY_AREA' TO GROUP 1;
SQL> ALTER DATABASE ADD STANDBY LOGFILE MEMBER '+RECOVERY_AREA' TO
GROUP 4;
```

Reestablish the second control file of the database by editing the CONTROL_FILES initialization parameter.

```
SQL> SELECT NAME FROM V$CONTROLFILE;

NAME
------------------------------------------------------------
/oradata/ORCL/controlfile/o1_mf_0fpqyfw7_.ctl

SQL> ALTER SYSTEM SET CONTROL_FILES=
'/oradata/ORCL/controlfile/mycontrol.ctl',
'+RECOVERY_AREA/mycontrol.ctl' SCOPE=spfile;

SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP NOMOUNT;
```

Use RMAN to restore the original control file into the new location.

```
RMAN> RESTORE CONTROLFILE FROM
'/oradata/ORCL/controlfile/o1_mf_0fpqyfw7_.ctl';
```

Primary Database – Cold Migration to ASM

Mount and open the database.

```
SQL> ALTER DATABASE MOUNT;
SQL> ALTER DATABASE OPEN;
```

Step 6: Restart SQL Apply (Logical Standby Database Only)

On the primary database, reenable log transport service

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_n=enable;
```

Replace *n* in LOG_ARCHIVE_DEST_STATE_*n* with the appropriate value.

On the standby database, unset the hidden SQL Apply parameter before restarting SQL apply.

```
SQL> EXECUTE DBMS_LOGSTDBY.APPLY_UNSET('_max_log_lookback');
SQL> ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;
```

This concludes the first outage. The database is now available and recovery related files will be written to the ASM managed storage.

Cold Migration - Database Backup

The database is available. The Oracle instance is using the original storage for the database area and the ASM disk group (+RECOVERY_AREA) for flash recovery area files. This section describes the third phase of the migration.

Step 1: Make the Initial Database Backup to the ASM Disk Group
Step 2: Remove the Redo Log Members
Step 3: Create an Incremental Backup of the Database to the ASM Disk Group

Step 1: Make the Initial Database Backup to the ASM Disk Group

If block change tracking is not enabled, then enabled it, and use RMAN to make the initial backup set of the Oracle database.

```
SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;

RMAN> BACKUP INCREMENTAL LEVEL 0 DATABASE TAG 'ASM_Migration';
```

Step 2: Remove the Redo Log Members

Remove the redo log members from the /oradata file system. To determine the redo log members in the /oradata file system, query the V$LOGFILE view. Drop both the online and standby redo log members.

```
SQL> ALTER DATABASE DROP LOGFILE MEMBER
'/oradata/ORCL/onlinelog/o1_mf_3_0fpqyp56_.log';

SQL> ALTER DATABASE DROP STANDBY LOGFILE MEMBER
'/oradata/ORCL/onlinelog/o1_mf_4_0fpqyp57_.log';
```

The incremental backup identifies those blocks that were modified during the course of the level 0 backup and is intended to reduce the actual outage time.

Step 3: Create an Incremental Backup of the Database to the ASM Disk Group

Using RMAN, create an incremental level 1 backup of the database.

```
RMAN> BACKUP INCREMENTAL LEVEL 1 DATABASE TAG 'ASM_Migration';
```

Primary Database – Cold Migration to ASM

Cold Migration - Second Outage – Data Area Moves to ASM

This phase starts the second and longer outage.

Step 1: Prepare the Control File for ASM Disk Groups
Step 2: Drop the Temporary Files on the File System
Step 3: Configure the Database to Use the Data Area Disk Group
Step 4: Create an Incremental Level 1 Database Backup (Logical Standby Database Only)
Step 5: Shut Down the Database and Unmount the File System
Step 6: Prepare the Disk for ASM Usage
Step 7: Migrate the Control Files to the Proper Locations on Both ASM Disk Groups
Step 8: Restore the Database Files from the Backup and Open the Database
Step 9: Restore the Temporary Files
Step 10: Restart SQL Apply
Step 11: Create the Redo Log File Members

Step 1: Prepare the Control File for ASM Disk Groups

Determine the current control file name.

```
SQL> SELECT NAME FROM V$CONTROLFILE;

NAME
---------------------------------------------------------------------
/oradata/ORCL/controlfile/mycontrol.ctl
+RECOVERY_AREA/mycontrol.ctl
```

Ensure that the directory structure exists in the new ASM disk group for the control files, and update the CONTROL_FILES initialization parameter in the SPFILE.

```
SQL> ALTER DATABASE BACKUP CONTROLFILE TO '+RECOVERY_AREA';
SQL> ALTER SYSTEM SET CONTROL_FILES =
'+RECOVERY_AREA/mycontrol.ctl' SCOPE=spfile;
```

Step 2: Drop the Temporary Files on the File System

Restart the database to a mount state.

```
SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP MOUNT;
```

Determine the name, location, and size of the current temporary files, and make a note of the size of each file.

```
SQL> SELECT BYTES, NAME FROM V$TEMPFILE;

     BYTES NAME
---------- --------------------------------------------------
  20971520 /oradata/ORCL/datafile/o1_mf_temp_0fpr0dbs_.tmp
```

Primary Database – Cold Migration to ASM

Drop all of the current temporary files. The following is an example that drops a temporary file.

```
SQL> ALTER DATABASE TEMPFILE
'/oradata/ORCL/datafile/o1_mf_temp_0fpr0dbs_.tmp' DROP;
```

Step 3: Configure the Database to Use the Data Area Disk Group

Update the DB_CREATE_FILE_DEST initialization parameter in the SPFILE.

```
SQL> ALTER SYSTEM SET DB_CREATE_FILE_DEST='+DATA_AREA'
SCOPE=spfile;
```

Note: If you are performing a cold migration of a physical standby database, then proceed to "Step 2: Shut Down the Database and Unmount the File System" on page 30. Otherwise, continue with the next step in this procedure.

Step 4: Create an Incremental Level 1 Database Backup (Logical Standby Database Only)

Using RMAN, create an incremental level 1 backup of the database.

```
RMAN> BACKUP INCREMENTAL LEVEL 1 DATABASE TAG 'ASM_Migration';
```

Step 5: Shut Down the Database and Unmount the File System

To prepare for the creation of the ASM DATA_AREA disk group, shut down the database, and unmount the data area file system.

```
SQL> SHUTDOWN IMMEDIATE
# umount /oradata
```

Step 6: Prepare the Disk for ASM Usage

Perform "Step 4: Prepare the Disks for Use by ASM" and "Step 5: Change the Permissions on the Disk Device Files" starting on page 9 for the new disks that will be added to the data area disk group.

When those steps are complete, create the ASM disk group.

```
SQL> CREATE DISKGROUP DATA_AREA NORMAL REDUNDANCY
FAILGROUP controller1 DISK '/dev/raw/raw9','/dev/raw/raw10'
FAILGROUP controller2 DISK '/dev/raw/raw11','/dev/raw/raw12';
```

Step 7: Migrate the Control Files to the Proper Locations on Both ASM Disk Groups

Mount the database and ensure that the directory structure exists in the newly created ASM disk group for the control files.

```
SQL> STARTUP MOUNT
SQL> ALTER DATABASE BACKUP CONTROLFILE TO '+DATA_AREA';
```

Update the CONTROL_FILE initialization parameter in the SPFILE, stop and restart the database cleanly, and use RMAN to restore the original control file into the new control file locations.

```
SQL> ALTER SYSTEM SET CONTROL_FILES =
'+DATA_AREA/ORCL/CONTROLFILE/mycontrol.ctl',
'+RECOVERY_AREA/ORCL/CONTROLFILE/mycontrol.ctl' SCOPE=spfile;

SQL> SHUTDOWN IMMEDIATE
SQL> STARTUP NOMOUNT

RMAN> RESTORE CONTROLFILE FROM '+RECOVERY_AREA/mycontrol.ctl';
```

Primary Database – Cold Migration to ASM

Note: If you are performing a cold migration of a physical standby database, then proceed to "Step 4: Restore the Database Files from the Backup" on page 30. Otherwise, continue with the next step in this procedure.

Step 8: Restore the Database Files from the Backup and Open the Database

Bring the database to a mounted state, and restore the database to the DATA_AREA disk group. The block change tracking file must either be dropped or relocated to ASM storage before opening the database.

```
RMAN> ALTER DATABASE MOUNT;
RMAN> RUN
{
RESTORE DATABASE;
SWITCH DATAFILE ALL;
RECOVER DATABASE;
}

SQL> ALTER DATABASE DISABLE BLOCK CHANGE TRACKING;
SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;
```

If the database is a primary database, then open the database with the ALTER DATABASE OPEN statement.

```
SQL> ALTER DATABASE OPEN;
```

If the database is a logical standby database, then open the database with the ALTER DATABSE OPEN RESETLOGS statement.

```
SQL> ALTER DATABASE OPEN RESETLOGS;
```

Step 9: Restore the Temporary Files

The alert log contains a message about temporary tablespaces that contain no data files. The following is an example of the message.

```
*************************************************************
WARNING: The following temporary tablespaces contain no files.
         This condition can occur when a backup controlfile has
         been restored.  It may be necessary to add files to these
         tablespaces.  That can be done using the SQL statement:

         ALTER TABLESPACE <tablespace_name> ADD TEMPFILE

         Alternatively, if these temporary tablespaces are no
longer
         needed, then they can be dropped.
           Empty temporary tablespace: TEMP
*************************************************************
```

Add the temporary file to each tablespace listed in the message.

```
SQL> ALTER TABLESPACE TEMP ADD TEMPFILE;
```

Step 10: Restart SQL Apply (Logical Standby Database Only)

If the database being migrated is a logical standby database, then run the following statements on the primary database.

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_n=ENABLE;
SQL> ALTER SYSTEM SWITCH LOGFILE;
```

Replace *n* in LOG_ARCHIVE_DEST_STATE_*n* with the appropriate value.

Primary Database – Cold Migration to ASM

If the database being migrated is a logical standby database, then run the following statements on the standby database.

```
SQL> ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;
```

Step 11: Create the Redo Log File Members

Reestablish the second redo log member on the DATA_AREA disk group for all online and standby redo log groups.

```
SQL> ALTER DATABASE ADD LOGFILE MEMBER '+DATA_AREA' TO GROUP 1;
SQL> ALTER DATABASE ADD STANDBY LOGFILE MEMBER '+DATA_AREA' TO
GROUP 4;
```

Cold Migration - After Migration

Finish the migration by validating that all files have been moved to the ASM disk groups.

Step 1: Verify File Placement

Step 1: Verify File Placement

Query the database and ensure that all files reside in either the DATA_AREA or RECOVERY_AREA ASM disk group.

```
SQL> SELECT NAME FROM V$DATAFILE
UNION
SELECT NAME FROM V$TEMPFILE
UNION
SELECT MEMBER FROM V$LOGFILE
UNION
SELECT NAME FROM V$CONTROLFILE
UNION
SELECT FILENAME FROM V$BLOCK_CHANGE_TRACKING
UNION
SELECT NAME FROM V$FLASHBACK_DATABASE_LOGFILE;

NAME
--------------------------------------------------------------------------------
+DATA_AREA/orcl/changetracking/ctf.262.1
+DATA_AREA/orcl/controlfile/mycontrol.ctl
+DATA_AREA/orcl/datafile/sysaux.260.1
+DATA_AREA/orcl/datafile/system.258.1
+DATA_AREA/orcl/datafile/undotbs1.259.1
+DATA_AREA/orcl/datafile/users.261.1
+DATA_AREA/orcl/onlinelog/group_1.264.1
+DATA_AREA/orcl/onlinelog/group_2.265.1
+DATA_AREA/orcl/onlinelog/group_3.266.1
+DATA_AREA/orcl/tempfile/temp.263.1
+RECOVERY_AREA/orcl/controlfile/mycontrol.ctl
+RECOVERY_AREA/orcl/flashback/log_1.256.1
+RECOVERY_AREA/orcl/onlinelog/group_1.257.1
+RECOVERY_AREA/orcl/onlinelog/group_2.258.1
+RECOVERY_AREA/orcl/onlinelog/group_3.259.1
```

MIGRATING A PHYSICAL STANDBY DATABASE TO ASM

Follow the instructions in the appropriate section:

Physical Standby Database – Hot Migration to ASM
Physical Standby Database - Cold Migration to ASM

Physical Standby Database – Hot Migration to ASM

There are three phases in the process of migrating a physical standby database from UNIX file systems to ASM.

Hot Migration - Before Migration
Hot Migration - The Switch
Hot Migration - After Migration

Hot Migration - Before Migration

During this phase of the migration, there is no outage to the primary database.

Step 1: Prepare the ASM Instance and Physical Standby Database
Step 2: Migrate the Oracle Redo Log and Standby Redo Log Files to ASM Disk Groups
Step 3: Migrate the Temporary Files to ASM Disk Groups
Step 4: Refresh the Previous Copy of the Oracle Data Files

Step 1: Prepare the ASM Instance and Physical Standby Database

Perform "Step 1: Create the Initialization Parameter File for the ASM Instance" through "Step 10: Make the Initial Copy of the Oracle Data Files" starting on page 8.

Step 2: Migrate the Oracle Redo Log and Standby Redo Log Files to ASM Disk Groups

Stop managed standby recovery, and clear all the redo log file groups.

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
SQL> SELECT GROUP# FROM V$LOG;
SQL> SELECT GROUP# FROM V$STANDBY_LOG;
SQL> ALTER DATABASE CLEAR LOGFILE GROUP group#;
```

In the final ALTER DATABASE statement, replace *group#* with a redo log group number, and repeat this statement for each redo log file group returned by the queries.

Physical Standby Database – Hot Migration to ASM

Step 3: Migrate the Temporary Files to ASM Disk Groups

When the database is opened in READ ONLY mode, the temporary files are recreated on the ASM disk group.

```
SQL> ALTER DATABASE OPEN READ ONLY;
```

Identify the current temporary files and their sizes, add an equivalent number of temporary files with the same sizes, and remove the original temporary files.

```
SQL> SELECT BYTES, NAME FROM V$TEMPFILE;

     BYTES NAME
---------- -------------------------------------------------------
  20971520 /oradata/ORCL/datafile/o1_mf_temp_0fs3bq8w_.tmp

SQL> alter tablespace temp add tempfile size 20m;
SQL> alter database tempfile
'/oradata/ORCL/datafile/o1_mf_temp_0fs3bq8w_.tmp' drop;
```

Return the database to managed recovery mode.

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT
LOGFILE DISCONNECT;
```

Step 4: Refresh the Previous Copy of the Oracle Data Files

Perform "Step 13: Refresh the Previous Copy of the Oracle Data Files" on page 11.

Hot Migration - The Switch

This phase is the start of the outage. Database downtime is determined by the length of time it takes to complete the steps in this section.

Step 1: Prepare the ASM Disk Groups to Store the Control Files
Step 2: Switch the Database Data Files to ASM Disk Groups
Step 3: Move the Flashback Database Log Files and Restart Managed Recovery

Step 1: Prepare the ASM Disk Groups to Store the Control Files

Ensure that the directory structure exists in the new ASM disk groups for the control files by creating control file backups in both disk groups, and update the CONTROL_FILES parameter in the SPFILE.

```
SQL> ALTER DATABASE BACKUP CONTROLFILE TO '+DATA_AREA';
SQL> ALTER DATABASE BACKUP CONTROLFILE TO '+RECOVERY_AREA';
SQL> ALTER SYSTEM SET CONTROL_FILES =
'+DATA_AREA/ORCL/CONTROLFILE/mycontrol.ctl',
'+RECOVERY_AREA/ORCL/CONTROLFILE/mycontrol.ctl' scope=spfile;
```

Determine the current control file names.

```
SQL> SELECT NAME FROM V$CONTROLFILE;

NAME
---------------------------------------------------------------------
/oradata/ORCL/controlfile/o1_mf_0fs38sx3_.ctl
/flash_recovery/ORCL/controlfile/o1_mf_0fs38t2w_.ctl
```

Physical Standby Database – Hot Migration to ASM

Step 2: Switch the Database Data Files to ASM Disk Groups

To prepare for restoring one of the control files into the new control file locations, shut down the database cleanly, restart the instance to a nomount state, and mount the database.

```
RMAN> SHUTDOWN IMMEDIATE;
RMAN> STARTUP NOMOUNT
RMAN> RESTORE CONTROLFILE FROM
'/oradata/ORCL/controlfile/o1_mf_0fs38sx3_.ctl';
RMAN> ALTER DATABASE MOUNT;
```

Switch the database to use the backup data file copies created previously.

```
RMAN> SWITCH DATABASE TO COPY;
```

Step 3: Move the Flashback Database Log Files and Restart Managed Recovery

If Flashback Database is enabled, then disable it. Next, enable Flashback Database to re-create the flashback log files in the ASM flash recovery area disk group, and start managed recovery.

```
SQL> ALTER DATABASE FLASHBACK OFF;
SQL> ALTER DATABASE FLASHBACK ON;
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT
LOGFILE DISCONNECT;
```

Hot Migration - After Migration

To complete the migration, remove the block change-tracking file, and validate that all files have been moved to the ASM disk groups. Optionally, you can move the ASM disk group back to the original database storage.

Step 1: Disable Block Change Tracking
Step 2: Verify file placement
Step 3: Optionally Migrate the ASM Disk Groups Back to the Original Database Storage

Step 1: Disable Block Change Tracking

If block change tracking was enabled during the migration, then disable it.

```
SQL> ALTER DATABASE DISABLE BLOCK CHANGE TRACKING;
```

Step 2: Verify file placement

Query the database to ensure that all files reside in either the DATA_AREA or RECOVERY_AREA ASM disk group.

```
SQL> SELECT NAME FROM V$CONTROLFILE
UNION
SELECT NAME FROM V$DATAFILE
UNION
SELECT NAME FROM V$TEMPFILE
UNION
SELECT MEMBER FROM V$LOGFILE
UNION
SELECT FILENAME FROM V$BLOCK_CHANGE_TRACKING
UNION
SELECT NAME FROM V$FLASHBACK_DATABASE_LOGFILE;
```

```
NAME
-------------------------------------------------------------------
+DATA_AREA/orcl/changetracking/ctf.256.1
+DATA_AREA/orcl/controlfile/mycontrol.ctl
+DATA_AREA/orcl/datafile/sysaux.259.1
+DATA_AREA/orcl/datafile/system.257.1
+DATA_AREA/orcl/datafile/undotbs1.258.1
+DATA_AREA/orcl/datafile/users.260.1
+DATA_AREA/orcl/onlinelog/group_1.263.1
+DATA_AREA/orcl/onlinelog/group_2.264.1
+DATA_AREA/orcl/onlinelog/group_3.265.1
+DATA_AREA/orcl/tempfile/temp.266.1
+RECOVERY_AREA/orcl/controlfile/mycontrol.ctl
+RECOVERY_AREA/orcl/flashback/log_1.276.1
+RECOVERY_AREA/orcl/onlinelog/group_1.265.1
+RECOVERY_AREA/orcl/onlinelog/group_2.266.1
+RECOVERY_AREA/orcl/onlinelog/group_3.267.1
```

Step 3: Optionally Migrate the ASM Disk Groups Back to the Original Database Storage

Optionally, if the additional storage required by the hot migration must be removed, then complete the procedure in "Appendix 1: Migrating ASM Disk Groups to Original Database Storage" starting on page 33.

Physical Standby Database - Cold Migration to ASM

There are five phases in the process of migrating a physical standby database from UNIX file systems to ASM.

Cold Migration - Before Migration
Cold Migration - First Outage – Move the Flash Recovery Area to ASM
Cold Migration - Database Backup
Cold Migration - Second Outage – Move Data Area to ASM
Cold Migration - After Migration

Cold Migration - Before Migration

During this phase of the migration, there is no outage to the database.

Step 1: Prepare for Cold Migration

Step 1: Prepare for Cold Migration

Perform "Step 1: Create the ASM Instance" and "Step 2: Save Backups to Tape" starting on page 15.

Cold Migration - First Outage – Move the Flash Recovery Area to ASM

During this phase, the database must be shut down so that the storage currently used for the recovery area may be reformatted for use by ASM.

Step 1: Clear the Old Flash Recovery Area
Step 2: Prepare the Standby Database to Use the ASM Disk Group

Step 1: Clear the Old Flash Recovery Area

Stop managed recovery.

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```

Perform "Step 2: Clear the Old Flash Recovery Area" through "Step 4: Create the Flash Recovery Area Disk Group" starting on page 17.

Step 2: Prepare the Standby Database to Use the ASM Disk Group

Mount the database instance, reenable flashback database if required, and reestablish the redo log file members and standby redo log file members for all the redo log groups.

```
SQL> STARTUP MOUNT
SQL> ALTER DATABASE FLASHBACK ON;
SQL> SELECT GROUP# FROM V$LOG;
SQL> SELECT GROUP# FROM V$STANDBY_LOG;
```

```
SQL> ALTER DATABASE CLEAR LOGFILE GROUP group#;
```

In the final ALTER DATABASE statement, replace *group#* with a redo log group number, and repeat this statement for each redo log file group returned by the queries.

Reestablish the second control file of the database by editing the CONTROL_FILES initialization parameter.

```
SQL> SELECT NAME FROM V$CONTROLFILE;

NAME
---------------------------------------------------------------------
/oradata/ORCL/controlfile/o1_mf_0fpqyfw7_.ctl

SQL> ALTER SYSTEM SET CONTROL_FILES=
'/oradata/ORCL/controlfile/mycontrol.ctl',
'+RECOVERY_AREA/mycontrol.ctl' SCOPE=spfile;

SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP NOMOUNT;
```

Use RMAN to restore the original control file into the new location.

```
RMAN> RESTORE CONTROLFILE FROM
'/oradata/ORCL/controlfile/o1_mf_0fpqyfw7_.ctl';
```

Mount the database, and restart managed recovery.

```
SQL> ALTER DATABASE MOUNT;
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT
LOGFILE DISCONNECT;
```

This concludes the first outage. The database is now available, and the flash recovery area resides in ASM managed storage.

Cold Migration - Database Backup

The Oracle instance is using the original storage for the database area and the ASM disk group (+RECOVERY_AREA) for flash recovery area files. This section describes the third phase of the migration

Step 1: Make the Initial Database Backup to the ASM Disk Group
Step 2: Create an Incremental Backup of the Database to the ASM Disk Group

Step 1: Make the Initial Database Backup to the ASM Disk Group

If block change tracking is not enabled, then enable it, and use RMAN to make the initial backup set of the Oracle database.

```
SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;

RMAN> BACKUP INCREMENTAL LEVEL 0 DATABASE TAG ‘ASM_Migration’;
```

Step 2: Create an Incremental Backup of the Database to the ASM Disk Group

Using RMAN, create an incremental, level 1 backup of the database.

```
RMAN> BACKUP INCREMENTAL LEVEL 1 DATABASE TAG 'ASM_Migration';
```

Physical Standby Database - Cold Migration to ASM

Cold Migration - Second Outage – Move Data Area to ASM

This phase starts the second and longer outage.

Step 1: Prepare the Database for ASM Disk Groups
Step 2: Shut Down the Database and Unmount the File System
Step 3: Prepare the Disk for ASM Usage
Step 4: Restore the Database Files from the Backup
Step 5: Restore the Temporary Files
Step 6: Create the Redo Log File Members

Step 1: Prepare the Database for ASM Disk Groups

Perform "Step 1: Prepare the Control File for ASM Disk Groups" and "Step 3: Configure the Database to Use the Data Area Disk Group" starting on page 20.

Step 2: Shut Down the Database and Unmount the File System

To prepare for the creation of the ASM DATA_AREA disk group, stop Redo Apply, shut down the database, and unmount the data area file system.

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
SQL> SHUTDOWN IMMEDIATE
# umount /oradata
```

Step 3: Prepare the Disk for ASM Usage

Perform "Step 6: Prepare the Disk for ASM Usage" and "Step 7: Migrate the Control Files to the Proper Locations on Both ASM Disk Groups" starting on page 21.

Step 4: Restore the Database Files from the Backup

Use RMAN to bring the database to a mounted state, and restore the database to the DATA_AREA disk group.

```
RMAN> ALTER DATABASE MOUNT;
RMAN> RUN {
RESTORE DATABASE;
SWITCH DATAFILE ALL;
RECOVER DATABASE NOREDO;
}
```

Either drop the block change tracking file or relocate it to ASM storage before restarting managed recovery.

```
SQL> ALTER DATABASE DISABLE BLOCK CHANGE TRACKING;
SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT
LOGFILE DISCONNECT;
```

Physical Standby Database - Cold Migration to ASM

Step 5: Restore the Temporary Files

Stop managed recovery, and open the database in READ ONLY mode.

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
SQL> ALTER DATABASE OPEN READ ONLY;
```

Using the previously identified temporary file sizes, recreate the temporary files.

```
SQL> ALTER TABLESPACE temp ADD TEMPFILE SIZE 20m;
```

Step 6: Create the Redo Log File Members

Reestablish the second redo log member on the DATA_AREA disk group for all online and standby redo log groups.

```
SQL> ALTER DATABASE CLEAR LOGFILE GROUP group#;
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT
LOGFILE DISCONNECT;
```

Cold Migration - After Migration

Complete the migration by validating that all files have been moved to the ASM disk groups.

Step 1: Verify File Placement

Step 1: Verify File Placement

Query the database to ensure that all files reside in either the DATA_AREA or RECOVERY_AREA ASM disk group.

```
SQL> SELECT NAME FROM V$DATAFILE
UNION
SELECT NAME FROM V$TEMPFILE
UNION
SELECT MEMBER FROM V$LOGFILE
UNION
SELECT NAME FROM V$CONTROLFILE
UNION
SELECT FILENAME FROM V$BLOCK_CHANGE_TRACKING
UNION
SELECT NAME FROM V$FLASHBACK_DATABASE_LOGFILE;

NAME
----------------------------------------------------------------------
+DATA_AREA/orcl/changetracking/ctf.262.1
+DATA_AREA/orcl/controlfile/mycontrol.ctl
+DATA_AREA/orcl/datafile/sysaux.260.1
+DATA_AREA/orcl/datafile/system.258.1
+DATA_AREA/orcl/datafile/undotbs1.259.1
+DATA_AREA/orcl/datafile/users.261.1
+DATA_AREA/orcl/onlinelog/group_1.264.1
+DATA_AREA/orcl/onlinelog/group_2.265.1
+DATA_AREA/orcl/onlinelog/group_3.266.1
+DATA_AREA/orcl/tempfile/temp.263.1
+RECOVERY_AREA/orcl/controlfile/mycontrol.ctl
+RECOVERY_AREA/orcl/flashback/log_1.256.1
+RECOVERY_AREA/orcl/onlinelog/group_1.257.1
+RECOVERY_AREA/orcl/onlinelog/group_2.258.1
+RECOVERY_AREA/orcl/onlinelog/group_3.259.1
```

CONCLUSION

This white paper has described two different methods for moving an existing Oracle Database 10*g* database from a traditional UNIX file system configuration into Automatic Storage Management. The best method to use depends on the amount of storage available for the migration and the amount of database down time that can be tolerated. Use of ASM is considered best practice for Oracle Database 10*g* databases because it provides the agility to dynamically change your storage configuration to further optimize your use of storage resources. ASM greatly reduces the management overhead required to provision and maintain the storage resources for your database. This white paper provides several options for migrating your data files into ASM so that you can leverage the automation and virtualization of ASM.

APPENDIX 1: MIGRATING ASM DISK GROUPS TO ORIGINAL DATABASE STORAGE

After completing a hot migration to ASM, the database and recovery area are in the new storage. If you must move the database back to the original storage, then complete the steps in this appendix. These steps can be performed while the database is online. You might need to move to original storage if you borrowed storage to accomplish the migration, and the borrowed storage must be returned.

Step1: Identify the Temporary Disks to Remove
Step 2: Add the Original Storage and Drop the Temporary Storage
Step 3: Verify That the Rebalance Operation
Step 4: Remove Temporary Storage

Step1: Identify the Temporary Disks to Remove

Connect to the ASM instance, and identify the ASM disk name and disk group name of the disks to be removed.

```
$ export ORACLE_SID=+ASM
$ sqlplus "/as sysdba"
SQL> SELECT FAILGROUP, NAME FROM V$ASM_DISK;

FAILGROUP                      NAME
------------------------------ ------------------------------
CONTROLLER2                    RECOVERY_AREA_0003
CONTROLLER2                    RECOVERY_AREA_0002
CONTROLLER1                    RECOVERY_AREA_0001
CONTROLLER1                    RECOVERY_AREA_0000
CONTROLLER2                    DATA_AREA_0003
CONTROLLER2                    DATA_AREA_0002
CONTROLLER1                    DATA_AREA_0001
CONTROLLER1                    DATA_AREA_0000
```

Step 2: Add the Original Storage and Drop the Temporary Storage

Modify the disk group by removing the temporary storage and adding the original storage back into the ASM disk group. Before the disks can be added into the appropriate disk group, format the original storage for use by ASM.

```
SQL> ALTER DISKGROUP DATA_AREA
ADD FAILGROUP controller1 DISK '/dev/raw/raw1','/dev/raw/raw2'
    FAILGROUP controller2 DISK '/dev/raw/raw3','/dev/raw/raw4'
DROP DISK data_area_0000, data_area_0001,
          data_area_0002, data_area_0003;

SQL> ALTER DISKGROUP RECOVERY_AREA
ADD FAILGROUP controller1 DISK '/dev/raw/raw5','/dev/raw/raw6'
    FAILGROUP controller2 DISK '/dev/raw/raw7','/dev/raw/raw8'
DROP DISK recovery_area_0000, recovery_area_0001,
          recovery_area_0002, recovery_area_0003;
```

Step 3: Verify That the Rebalance Operation Is Complete

Check the status of the rebalance operation, which occurs in the background, by querying the V$ASM_OPERATION view. The rebalance operation is complete when a query on this view returns no rows.

```
SQL> SELECT * FROM V$ASM_OPERATION;
```

```
GROUP_NUMBER OPERA STAT  POWER ACTUAL SOFAR EST_WORK EST_RATE
EST_MINUTES
------------ ----- ---- ------ ------ ----- -------- -------- -----
------
           1 REBAL RUN      10     10    72     2114     1254
1

SQL> SELECT * FROM V$ASM_OPERATION;

no rows selected
```

When the V$ASM_OPERATION view returns no rows, query the V$ASM_DISK view again to verify that the disk group is using the original disks.

```
SQL> SELECT PATH,FAILGROUP,NAME FROM V$ASM_DISK;

PATH            FAILGROUP       NAME
--------------- --------------- --------------------
/dev/raw/raw9
/dev/raw/raw10
/dev/raw/raw11
/dev/raw/raw12
/dev/raw/raw13
/dev/raw/raw14
/dev/raw/raw15
/dev/raw/raw16
/dev/raw/raw8   CONTROLLER2     RECOVERY_AREA_0007
/dev/raw/raw7   CONTROLLER2     RECOVERY_AREA_0006
/dev/raw/raw6   CONTROLLER1     RECOVERY_AREA_0005
/dev/raw/raw5   CONTROLLER1     RECOVERY_AREA_0004
/dev/raw/raw4   CONTROLLER2     DATA_AREA_0007
/dev/raw/raw3   CONTROLLER2     DATA_AREA_0006
/dev/raw/raw2   CONTROLLER1     DATA_AREA_0005
/dev/raw/raw1   CONTROLLER1     DATA_AREA_0004
```

Do not remove the storage until the rebalance operation has completed, which is indicated by no rows being returned by the V$ASM_OPERATION query.

Step 4: Remove Temporary Storage

The temporary storage can be removed from the system.

APPENDIX 2: MIGRATING A TABLESPACE TO ASM

This appendix describes the following ways to migrate a tablespace to ASM:

- Data file copy only
- Data file copy and incremental backup

The data file copy only method is recommended for relatively small tablespaces that can be taken offline for the duration of the data file copy operation. The data file copy and incremental backup method is recommended for larger tablespaces that cannot be offline for the duration of the data file copy operation.

The instructions in this appendix can be used for a tablespace in a primary, logical standby, or physical standby database. The steps provided assume that the ASM instances have already been configured and started and there is an existing ASM disk group named +DATA with sufficient free space to hold the tablespace being migrated.

Follow the instructions in the appropriate section:

Migrating a Tablespace to ASM with Data File Copy
Migrating a Tablespace to ASM with Data File Copy and Incremental Backup

Migrating a Tablespace to ASM with Data File Copy

Complete the following steps to migrate a tablespace to ASM with data file copy.

Step 1: Stop Apply (Standby Database Only)
Step 2: Take the Tablespace Offline
Step 3: Back Up the Tablespace Data Files
Step 4: Switch the Tablespace to the New Data Files
Step 5: Bring the Tablespace Online
Step 6: Start Apply (Standby Database Only)

Step 1: Stop Apply (Standby Database Only)

This step is required only when a tablespace in a logical standby database or physical standby database is being migrated to ASM. If a primary database is being migrated to ASM, then this step is not required, and you can proceed to "Step 2: Take the Tablespace Offline."

If the tablespace is in a logical standby database, then run the following statement to stop SQL Apply:

```
SQL> ALTER DATABASE STOP LOGICAL STANDBY APPLY;
```

If the tablespace is in a physical standby database, then run the following statement to stop Redo Apply:

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```

Step 2: Take the Tablespace Offline

If the tablespace is in a primary or logical standby database, then take the tablespace offline. If the tablespace is in a physical standby database, then this step is not required, and you can proceed to "Step 3: Back Up the Tablespace Data Files."

The following example takes a tablespace named users offline.

```
SQL> ALTER TABLESPACE users OFFLINE;
```

Step 3: Back Up the Tablespace Data Files

Use RMAN to back up the tablespace and specify ASM format for the data files.

```
RMAN> BACKUP AS COPY TABLESPACE users FORMAT '+DATA';
```

Step 4: Switch the Tablespace to the New Data Files

Use RMAN to switch the tablespace to use the new data files.

```
RMAN> SWITCH TABLESPACE users TO COPY;
```

Step 5: Bring the Tablespace Online

If the tablespace is in a primary or logical standby database, then bring the tablespace online. If the tablespace is in a physical standby database, then this step is not required, and you can proceed to "Step 6: Start Apply (Standby Database Only)."

The following example brings a tablespace named users online.

```
SQL> ALTER TABLESPACE users ONLINE;
```

Step 6: Start Apply (Standby Database Only)

This step is required only when a tablespace in a logical standby database or physical standby database is being migrated to ASM.

If the tablespace is in a logical standby database, then run the following statement to start SQL apply:

```
SQL> ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;
```

If the tablespace is in a physical standby database, then run the following statement to start Redo Apply:

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT
LOGFILE DISCONNECT;
```

Migrating a Tablespace to ASM with Data File Copy and Incremental Backup

Complete the following steps to migrate a tablespace to ASM with data file copy and incremental backup.

Step 1: Stop Apply (Standby Database Only)
Step 2: Enable Block Change Tracking
Step 3: Make the Initial Backup of the Oracle Data Files
Step 4: Take the Tablespace Offline
Step 5: Refresh the Previous Copy of the Oracle Data Files
Step 6: Switch the Tablespace to the New Data Files
Step 7: Bring the Tablespace Online
Step 8: Start Apply (Standby Database Only)

Step 1: Stop Apply (Standby Database Only)

This step is required only when a tablespace in a logical standby database or physical standby database is being migrated to ASM. If a primary database is being migrated to ASM, then this step is not required, and you can proceed to "Step 2: Enable Block Change Tracking."

If the tablespace is in a logical standby database, then run the following statement to stop SQL Apply:

```
SQL> ALTER DATABASE STOP LOGICAL STANDBY APPLY;
```

If the tablespace is in a physical standby database, then run the following statement to stop Redo Apply:

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```

Step 2: Enable Block Change Tracking

Enable block change tracking if it is not enabled.

```
SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;
```

Step 3: Make the Initial Backup of the Oracle Data Files

Use RMAN to back up the database with the AS COPY option. The following example makes a backup of a tablespace named users.

```
RMAN> BACKUP INCREMENTAL LEVEL 0 AS COPY TAG 'ASM_Migration'
TABLESPACE users FORMAT '+DATA';
```

Step 4: Take the Tablespace Offline

If the tablespace is in a primary or logical standby database, then take the tablespace offline. If the tablespace is in a physical standby database, then this step is not

required, and you can proceed to "Step 5: Refresh the Previous Copy of the Oracle Data Files."

The following example takes a tablespace named users offline.

```
SQL> ALTER TABLESPACE users OFFLINE;
```

Step 5: Refresh the Previous Copy of the Oracle Data Files

Using RMAN, roll forward the data file copies by creating an incremental level 1 backup of the tablespace and merging this backup with the earlier level 0 backup.

```
RMAN> RUN {
BACKUP INCREMENTAL LEVEL 1 FOR RECOVER OF COPY WITH TAG
'ASM_Migration' TABLESPACE users;
RECOVER COPY OF TABLESPACE users WITH TAG 'ASM_Migration';
}
```

Step 6: Switch the Tablespace to the New Data Files

Use RMAN to switch the tablespace to use the new data files

```
RMAN> SWITCH TABLESPACE users TO COPY;
```

Step 7: Bring the Tablespace Online

If the tablespace is in a primary or logical standby database, then bring the tablespace online. If the tablespace is in a physical standby database, then this step is not required, and you can proceed to "Step 8: Start Apply (Standby Database Only)."

The following example brings a tablespace named users online.

```
SQL> ALTER TABLESPACE users ONLINE;
```

Step 8: Start Apply (Standby Database Only)

This step is required only when a tablespace in a logical standby database or physical standby database is being migrated to ASM.

If the tablespace is in a logical standby database, then run the following statement to start SQL apply:

```
SQL> ALTER DATABASE START LOGICAL STANDBY APPLY IMMEDIATE;
```

If the tablespace is in a physical standby database, then run the following statement to start Redo Apply:

```
SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT
LOGFILE DISCONNECT;
```

Oracle Database 10*g* Migration to Automatic Storage Management
August 2005
Author: Andrew Babb, High Availability Systems Group, Server Technologies
Contributing Authors: Michael Hallas, Susan Kornberg, Paul Manning, Lawrence To, Doug Utzig, Nitin Vengurlekar

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com