Oracle Database 11*g*: Administration Workshop II

Volume III • Student Guide

D50079GC20

Edition 2.0

September 2010

D62545

Author

Maria Billings

Technical Contributors and Reviewers

Christian Bauwens
Yanti Chang
Timothy Chien
Joe Fong
Andy Fortunak
Gerlinde Frenzen
Mark Fuller
Peter Fusek
Joel Goodman
Vimala Jacob
Dominique Jeunot
Pete Jones
Fukue Kawabe
Donna Keesling
Sean Kim
Achiel Langers
Gwen Lazenby
Jerry Lee
Deidre Matishak
Bill Millar
Lakshmi Naraparreddi
Ira Singer
Ranbir Singh
James Spiller
Matt Taylor
Branislav Valny
Jean-Francois Verrier

Editors

Nita Pavitran
Raj Kumar

Graphic Designer

Satish Bettegowda

Publisher

Jayanthy Keshavamurthy

Contents

11 Using Flashback Technology II

12 Performing Flashback Database

13 Managing Memory

14 Managing Database Performance

20 Duplicating a Database

Appendix A: Practices and Solutions

Appendix B: Performing Tablespace Point-in-Time Recovery

Appendix A
Practices and Solutions

Table of Contents

Practices for Lesson 1

The paper practices for this lesson are embedded in the lesson itself, to facilitate reviewing the database architecture for yourself and in class. Below are possible answers.

Practice 1-1: Reviewing the Database Architecture

For page 1-3: **Naming the Core Components of an Oracle Database Server**

1) The two main components of a basic Oracle Database system: instance and database.
2) The Instance consists of memory structures and background processes.
3) The three major structures in Oracle Database server architecture are: memory structures, process structures and storage structures.
4) A session is a connection between the user login and the database instance.
5) The graphic on slide 1-4 is the answer to the task on slide 1-3

For page 1-7: **Naming the Memory Structures of an Oracle Database**

1) Which are the components of the PGA: stack space and user global area.
2) Name the main components of the SGA:
   - Shared pool
   - Database buffer cache
   - Redo log buffer
   - Large pool
   - Java pool
   - Streams pool
   - Keep buffer pool
   - Recycle buffer pool
   - nK buffer pool
3) The graphic on slide 1-8 is the answer to the task on slide 1-7.

For page 1-14: **Adding Process Names**

1) The DBWn process writes the dirty buffers to the data files.
2) The LGWR process writes the redo entries to the online redo log files.
3) The CKPT process writes checkpoint information in the control file and each data file header.
4) The SMON process performs recovery on instance startup.
5) The PMON process performs process recovery when a user process fails.
6) The RECO process resolves in-doubt distributed transactions.
7) The ARCn processes copy redo log files to a designated storage device.

Practices for Lesson 2

General Practice Note: To reduce clutter in the output, product banners, space lines and other repeating information is only shown when they first appear. Later, they are removed to not distract from the essence of a practice.

Practice 2-1: Configuring `ARCHIVELOG` Mode

In this practice, you configure your database to archive redo logs before reusing them.

1) Set up the environment for the `ORCL` database, using the `oraenv` script, and enter `orcl` when prompted for `ORACLE_SID`. If it is already set up for `orcl` (that is, you see `orcl` in the brackets), press Enter.

   Note: This command is a period, followed by a space, and then the `oraenv` script name.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
```

2) Use `labs` as your working directory. Unless otherwise indicated, always work from that directory.

```
$ cd ~/labs
```

3) Determine the archive mode your database is running in now.

```
$ sqlplus / as sysdba
SQL*Plus: Release 11.2.0.1.0 Production on Tue Jul 21 14:32:29
2009

Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> archive log list
Database log mode              No Archive Mode
Automatic archival             Disabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     6
Current log sequence           8
SQL>
```

   Note that it is running in `NOARCHIVELOG` mode.

Practice 2-1: Configuring ARCHIVELOG Mode (continued)

4) Because you can change the `ARCHIVELOG` mode only when the database is mounted, shut down the database.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
```

5) Mount the database.

```
SQL> startup mount
ORACLE instance started.

Total System Global Area  481259520 bytes
Fixed Size                  1337324 bytes
Variable Size             339740692 bytes
Database Buffers          134217728 bytes
Redo Buffers                5963776 bytes
Database mounted.
SQL>
```

6) Alter the database to use `ARCHIVELOG` mode.

```
SQL> alter database archivelog;

Database altered.

SQL>
```

7) Open the database.

```
SQL> alter database open;

Database altered.

SQL>
```

Practice 2-2: Resizing the Flash Recovery Area

In this practice, you enlarge the Flash Recovery Area.

1) Determine how big the Flash Recovery Area is now.

```
SQL> show parameter recovery_file_dest_size

NAME                                 TYPE        VALUE
------------------------------------ ----------- -----
db_recovery_file_dest_size           big integer 4062M
SQL>
```

Note that it is 4062 MB or 4 GB in size.

2) How big *could* the Flash Recovery Area be?

a) Determine where the Flash Recovery Area is located.

```
SQL> show parameter recovery_file_dest

NAME                                 TYPE        VALUE
------------------------------------ ----------- -------------
db_recovery_file_dest                string      +FRA
db_recovery_file_dest_size           big integer 4062
SQL>
```

Note that the Flash Recovery Area is in the `FRA` disk group.

Do not exit from your SQL*Plus session.

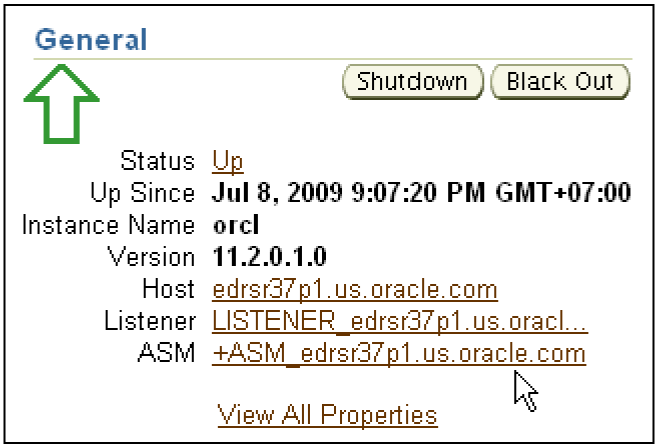
3) Use the `https://<hostname>:1158/em` URL to log into Enterprise Manager, as the `SYS` user with the `oracle_4U` password, and connect as `SYSDBA`.

Note: The first time that you are using a newly configured machine, you may be asked to add a security exception. The exact formulation and number of pages depends on your web browser. Follow the prompts and add the exception in the training environment.

Note 2: Also if Enterprise Manager shows, that the database is down, wait a minute and refresh the page. (This can occur due to the time intervals in which the database agent works.)

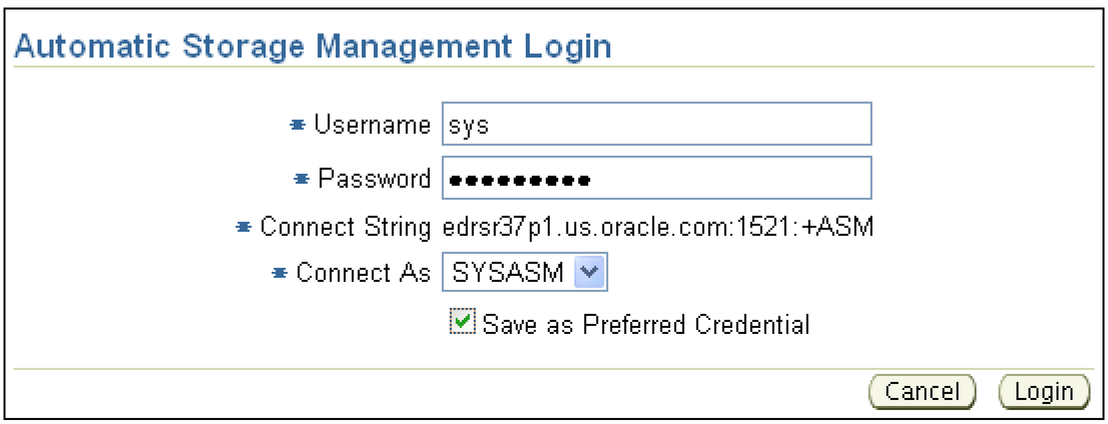
4) On the database home page, click the +ASM_<hostname> link.

Practice 2-2: Resizing the Flash Recovery Area (continued)

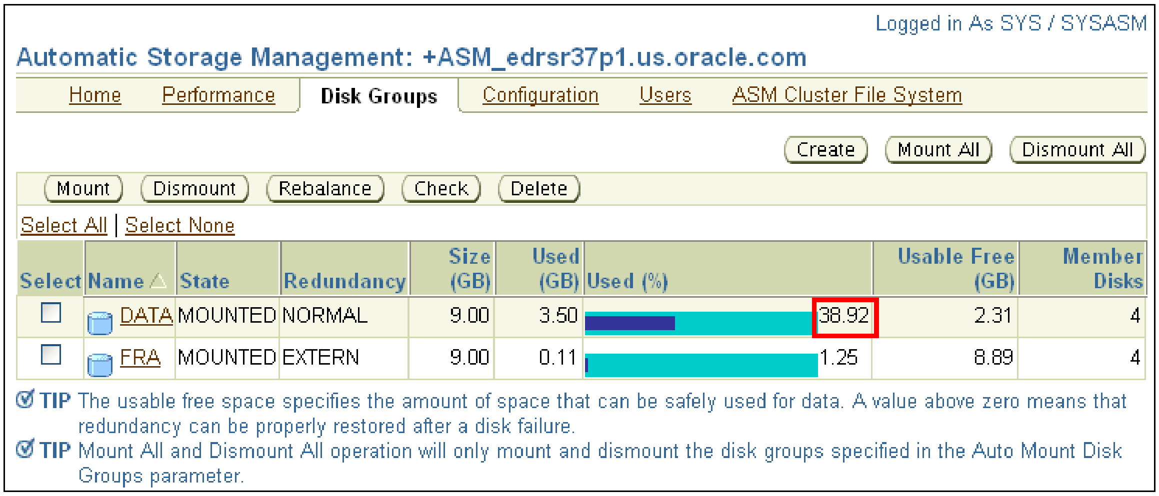


5) Click the Disk Groups tab.

6) If prompted for ASM login credentials, enter `sys` as username and `oracle_4U` as password. Connect as SYSASM, click "Save as Preferred Credential", and then click Login.



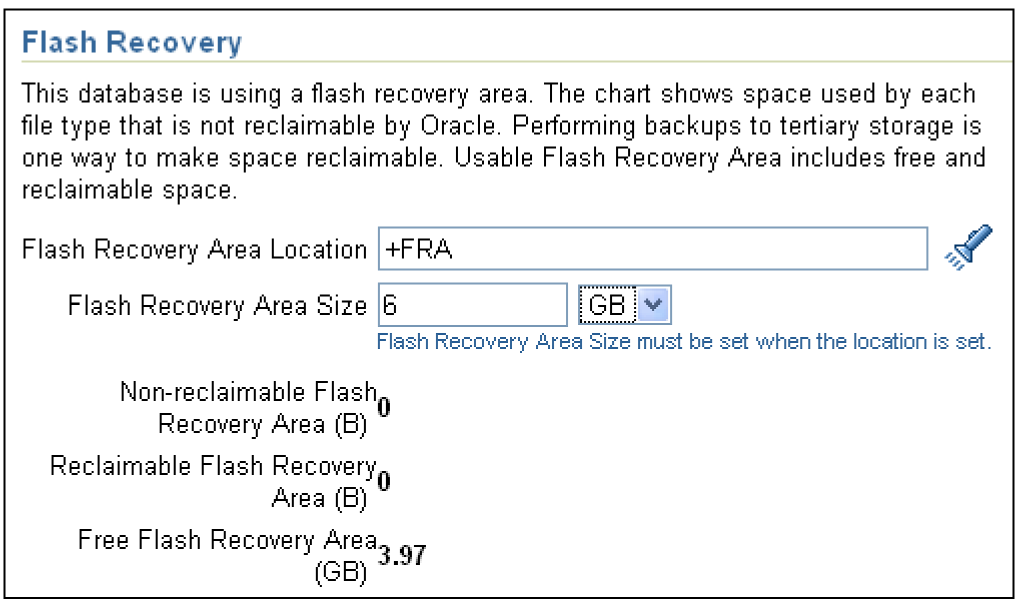
7) When the disk group list appears, note the Usable Free column for the FRA disk group.



Note that the FRA disk group has enough usable free space to enlarge the Flash Recovery Area considerably, although your displayed values may be a little different.

8) Change the Flash Recovery Area size to 6 GB.

   a) Navigate to Database > Availability > Recovery Settings and change the Flash Recovery Area Size to 6 GB.



   b) Click Show SQL, and note the SQL that will be run. This is important to know because if the Flash Recovery Area is having sizing problems, you may not be able to run Enterprise Manager to change it.

```
ALTER SYSTEM SET db_recovery_file_dest_size = 6442450944 SCOPE=BOTH
```

   c) Click Return, and then click Apply.

Practice 2-2: Resizing the Flash Recovery Area (continued)

9) Verify the size of the Flash Recovery Area by using SQL*Plus. Then exit your SQL*Plus session.

```
SQL> show parameter recovery_file_dest_size

NAME                                 TYPE        VALUE
------------------------------------ ----------- -------------
db_recovery_file_dest_size           big integer 6G

SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining
and Real Application Testing options
```

Practice 2-3: Verifying the Backup Destination

In this practice, you test the backup destination to see where backups are written.

Use the `oraenv` script to ensure you are still using the `orcl` instance in your terminal session.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
```

1) Start RMAN.

```
$ rman target /

Recovery Manager: Release 11.2.0.1.0 - Production on Tue Jul
21 14:52:20 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates.  All
rights reserved.

connected to target database: ORCL (DBID=1220535480)
RMAN>
```

2) See if the control files are automatically backed up.

```
RMAN> show CONTROLFILE AUTOBACKUP;

using target database control file instead of recovery catalog
RMAN configuration parameters for database with db_unique_name
ORCL are:
CONFIGURE CONTROLFILE AUTOBACKUP OFF; # default

RMAN>
```

Note that automatic backup is not enabled.

3) Configure RMAN to automatically back up the control file when any backups are done.

```
RMAN> configure controlfile autobackup on;

new RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters are successfully stored

RMAN>
```

4) List the database files in your schema, to understand which file you back up in the next task.

```
RMAN> report schema;

Report of database schema for database with db_unique_name
ORCL

List of Permanent Datafiles
```

```
===========================
File Size(MB) Tablespace           RB segs Datafile Name
---- -------- -------------------- ------- -------------------
-----
1    680      SYSTEM               YES
+DATA/orcl/datafile/system.256.692754557
2    590      SYSAUX               NO
+DATA/orcl/datafile/sysaux.257.692754559
3    100      UNDOTBS1             YES
+DATA/orcl/datafile/undotbs1.258.692754561
4    5        USERS                NO
+DATA/orcl/datafile/users.259.692754561
5    100      EXAMPLE              NO
+DATA/orcl/datafile/example.265.692754837

List of Temporary Files
=======================
File Size(MB) Tablespace     Maxsize(MB) Tempfile Name
---- -------- -------------- ----------- ---------------------
1    28       TEMP           32767       +DATA/orcl/temp01.dbf

RMAN>
```

Note: If you are not connected to the recovery catalog, the `RB segs` column contains `***` as a value. It contains the `YES` and `NO` values when you are connected to the recovery catalog.

5) Which tablespace is stored in file 5?

Answer: The `EXAMPLE` tablespace

6) Take a backup of data file 5, and note where the backup is written, then exit.

```
RMAN> backup datafile 5;

Starting backup at 21-JUL-09
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=55 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00005
name=+DATA/orcl/datafile/example.265.692754837
channel ORA_DISK_1: starting piece 1 at 21-JUL-09
channel ORA_DISK_1: finished piece 1 at 21-JUL-09
piece
handle=+FRA/orcl/backupset/2009_07_21/nnndf0_tag20090721t14535
8_0.260.692808839 tag=TAG20090721T145358 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time:
00:00:15
```

Practice 2-3: Verifying the Backup Destination (continued)

```
Finished backup at 21-JUL-09

Starting Control File and SPFILE Autobackup at 21-JUL-09
piece
handle=+FRA/orcl/autobackup/2009_07_21/s_692808854.261.6928088
57 comment=NONE
Finished Control File and SPFILE Autobackup at 21-JUL-09
RMAN>
RMAN> exit
```

Note that the backup file is written to the FRA.

Practice 2-4: Configuring the Retention Policy

In this practice, you change the retention policy for backups.
Note that the default retention policy in Oracle Database 11*g* Release 2 is already set to redundancy 1. So this practice is for training purposes only.

1) Use the `oraenv` script to ensure that you are still using the `ORCL` instance in your terminal session.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
```

2) Return to the RMAN session (or if you closed it, start a new one) and show the current retention policy setting.

```
$ rman target /
RMAN> show retention policy;

RMAN configuration parameters for database with db_unique_name
ORCL are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
RMAN>
```

3) If your redundancy is 1, go to step 5 and exit. If your redundancy had another value, change it to ensure that one copy of each file is backed up.

```
RMAN> configure retention policy to redundancy 1;

old RMAN configuration parameters:
CONFIGURE RETENTION POLICY TO NONE;
new RMAN configuration parameters:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1;
new RMAN configuration parameters are successfully stored

RMAN>
```

4) Check the retention policy setting again.

```
RMAN> show retention policy;

RMAN configuration parameters for database with db_unique_name
ORCL are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1;
```

5) Exit RMAN.

```
RMAN> exit

Recovery Manager complete.
```

Practice 3-1: Creating a Recovery Catalog

In this practice, you create a recovery catalog.

1) Use DBCA to start the process of creating a recovery catalog database.

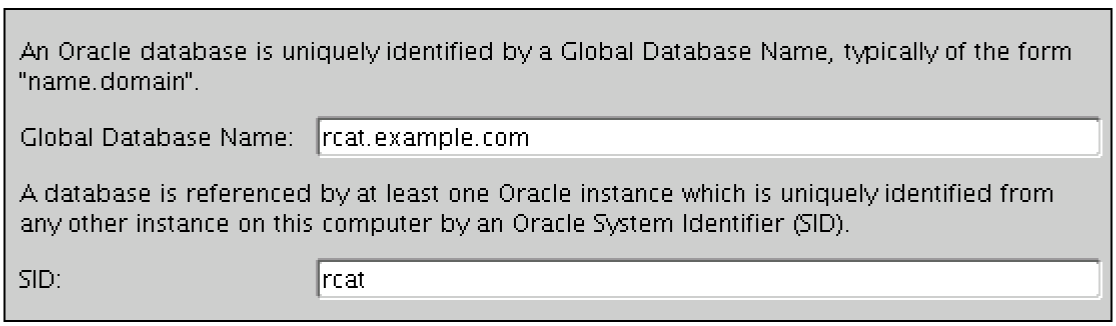
```
$ dbca
```

2) On the Welcome page, click Next.

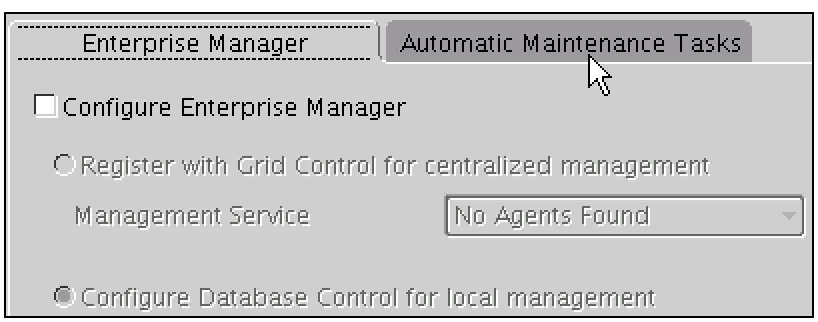
3) On the Operations page, select the Create a Database option, and then click Next.

4) On the Database Templates page, leave the default of General Purpose or Transaction Processing, and then click Next.

5) On the Database Identification page, enter the name of the recovery catalog database. Use **`rcat.example.com`** as the Global Database Name. Then click Next.

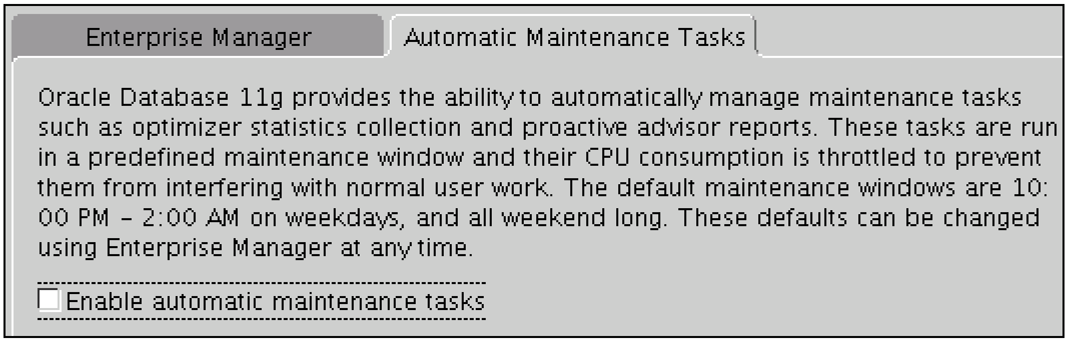


6) On the Management Options page, deselect the Configure Enterprise Manager option, and then click the Automatic Maintenance Tasks tab.



7) On the Automatic Maintenance Tasks page, deselect “Enable automatic maintenance tasks” and then click Next.

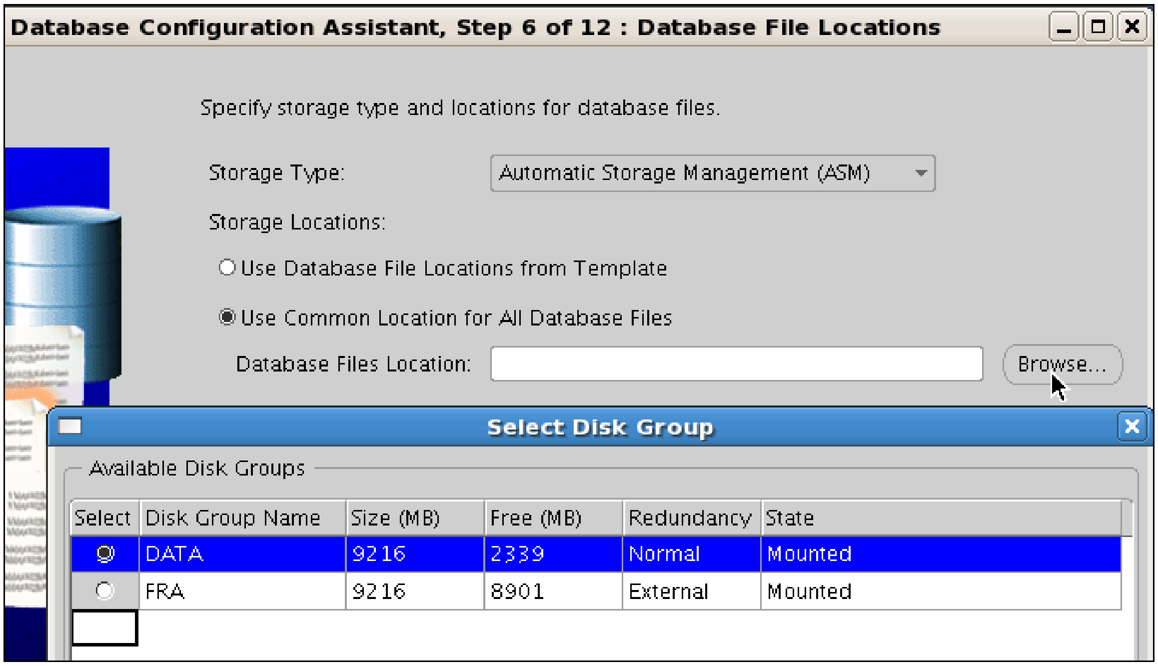
Practice 3-1: Creating a Recovery Catalog (continued)



8) On the Database Credentials page, select the option Use the Same Administrative Password for All Accounts. Then enter **`oracle_4U`** in the password fields and click Next.

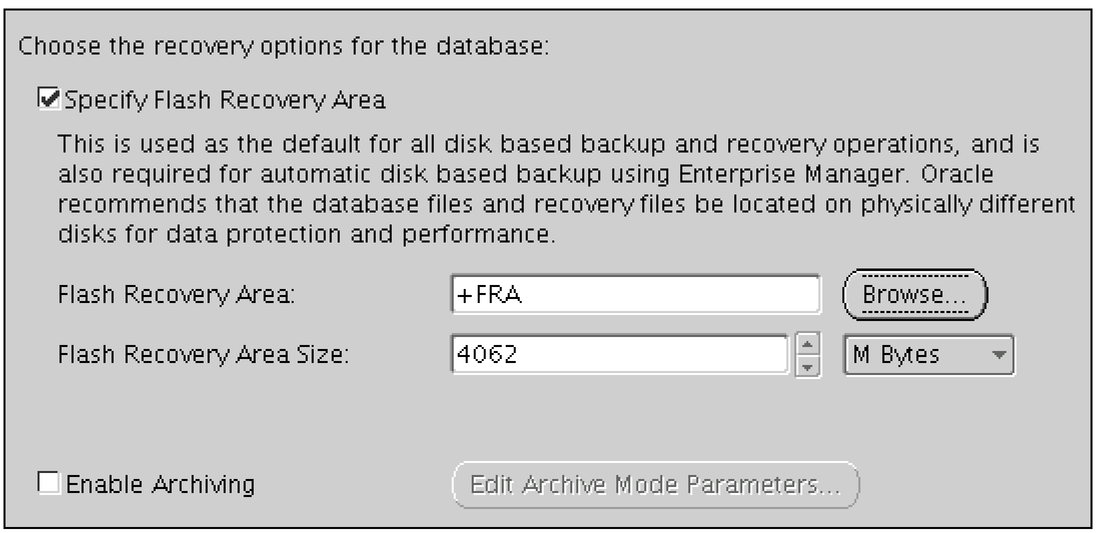
9) On the Database File Locations Page, select **Automatic Storage Management (ASM)** as Storage Type. Click Use Common Location for All Database Files and then use the Browse button to select the +**DATA** disk group, and click OK.

Practice 3-1: Creating a Recovery Catalog (continued)



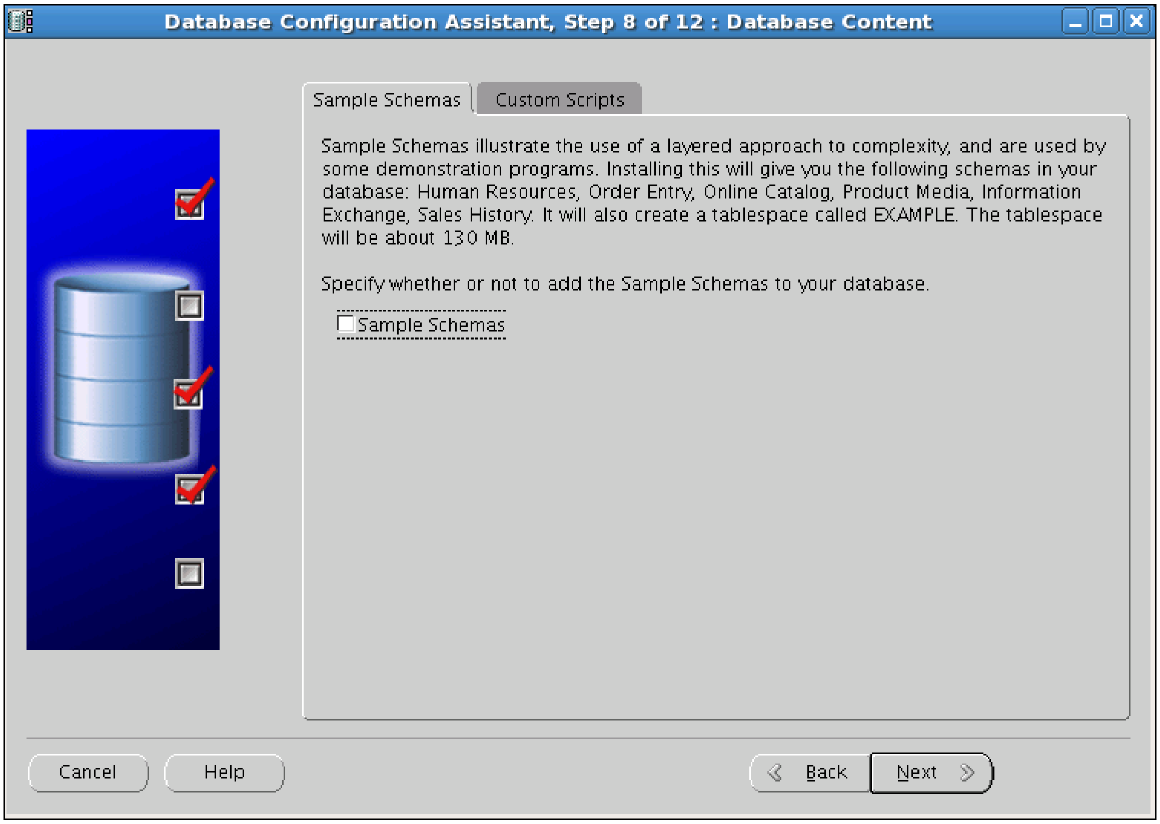
10) Then click Next.

11) On the Recovery Configuration page, select **Specify Flash Recovery Area**, use the Browse button to select the **+FRA** diskgroup (click OK) and ensure that the Flash Recovery Area Size is **4062 M Bytes**. Then click Next.

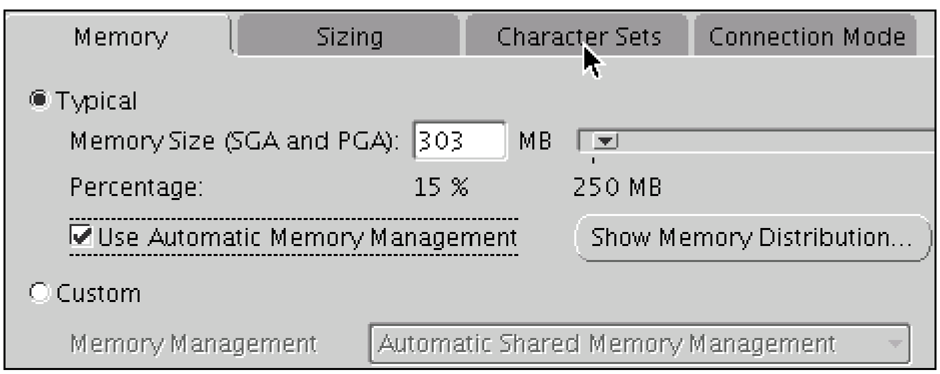


12) On the Database Content page, confirm that the Sample Schemas option is not selected, and then click Next.

Practice 3-1: Creating a Recovery Catalog (continued)

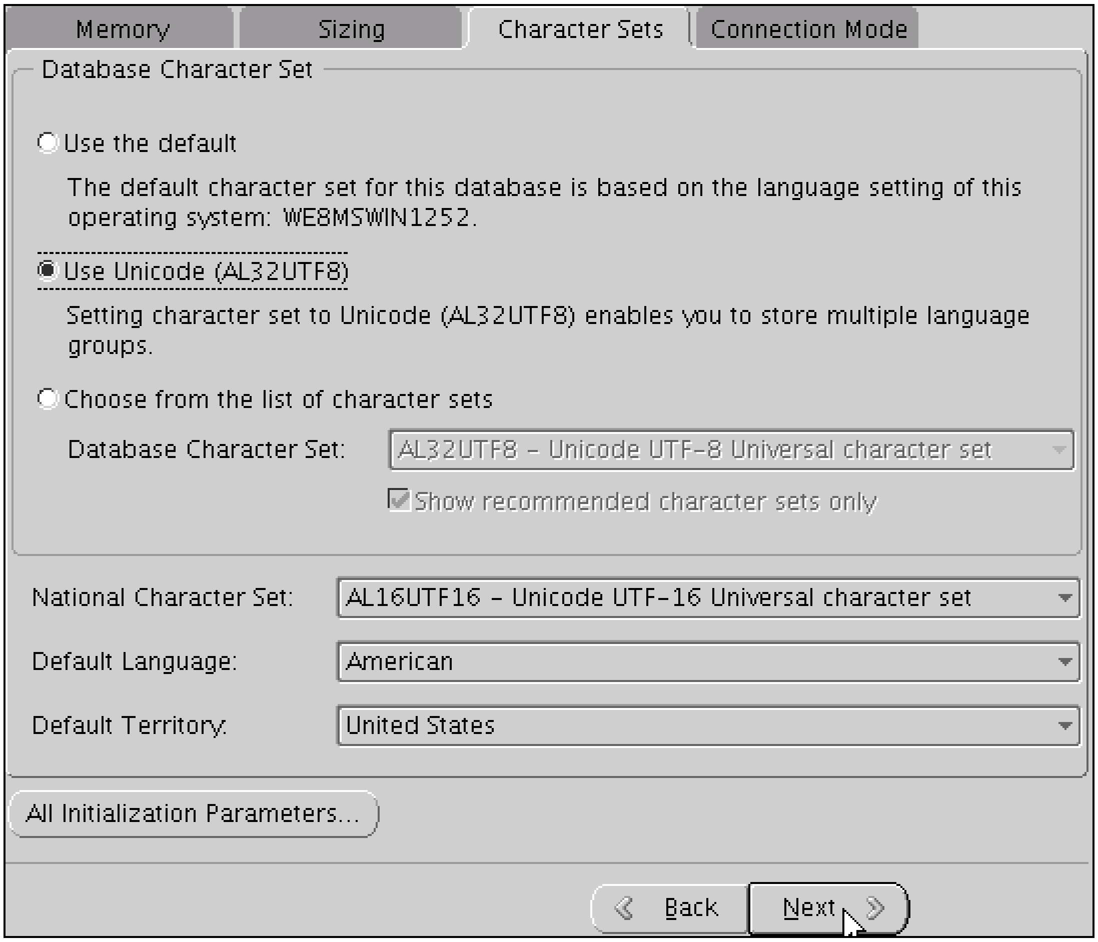


13) On the Initialization Parameter page, set the memory size to **303 MB**, click **Use Automatic Memory Management**, and then click the Character Sets tab.



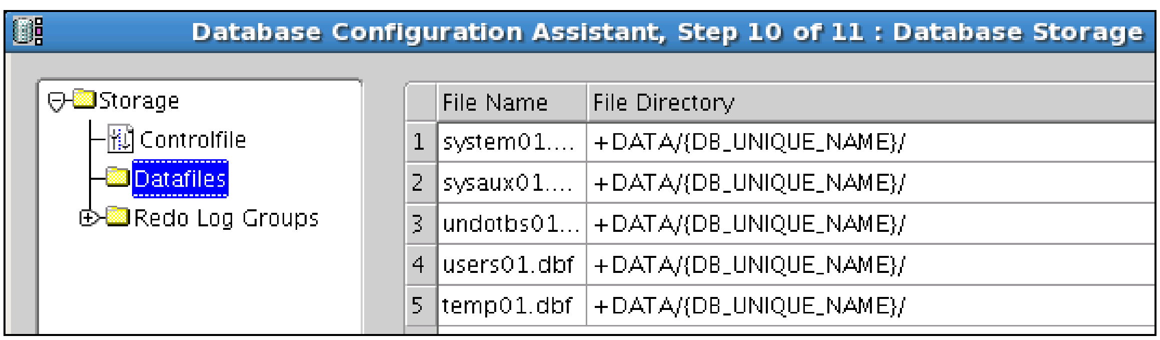
14) Click **Use Unicode (AL32UTF8)** and then click Next.

Practice 3-1: Creating a Recovery Catalog (continued)



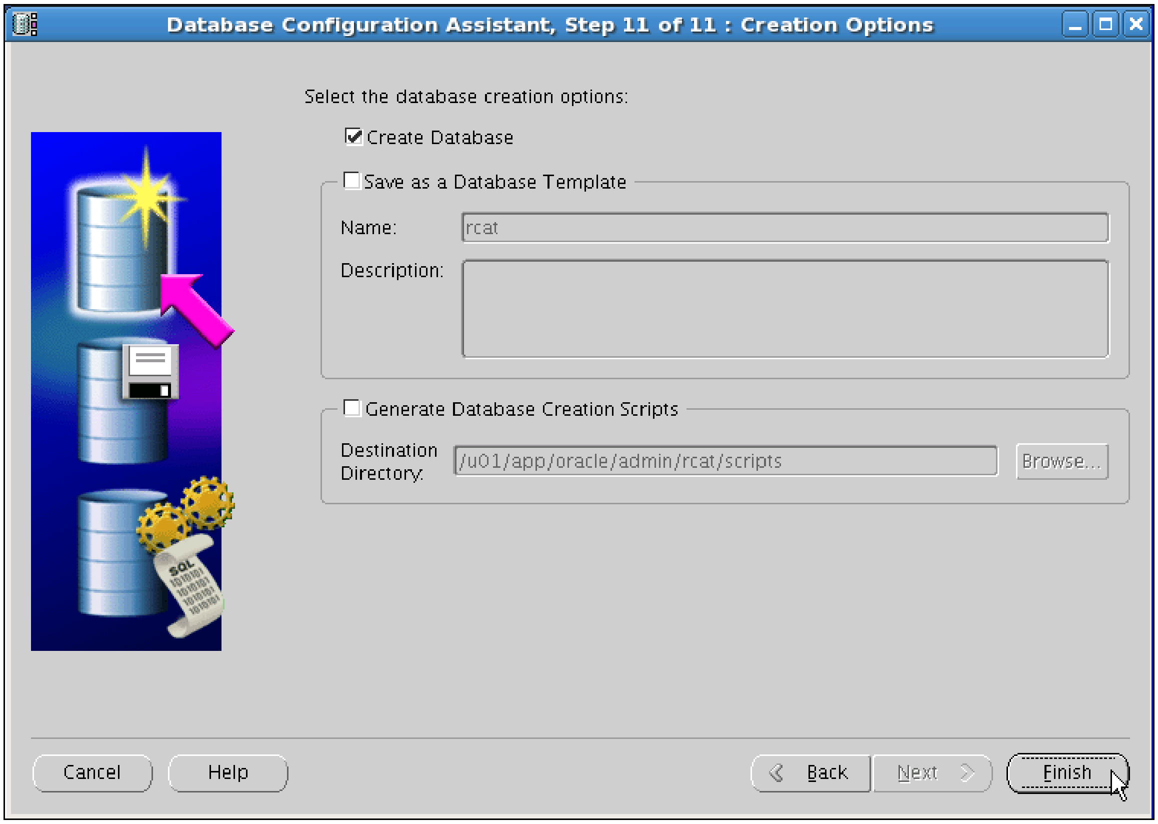
15) If the Security Settings page appears, keep the defaults, and then click Next.

16) On the Database Storage page, review the configuration, and then click Next.

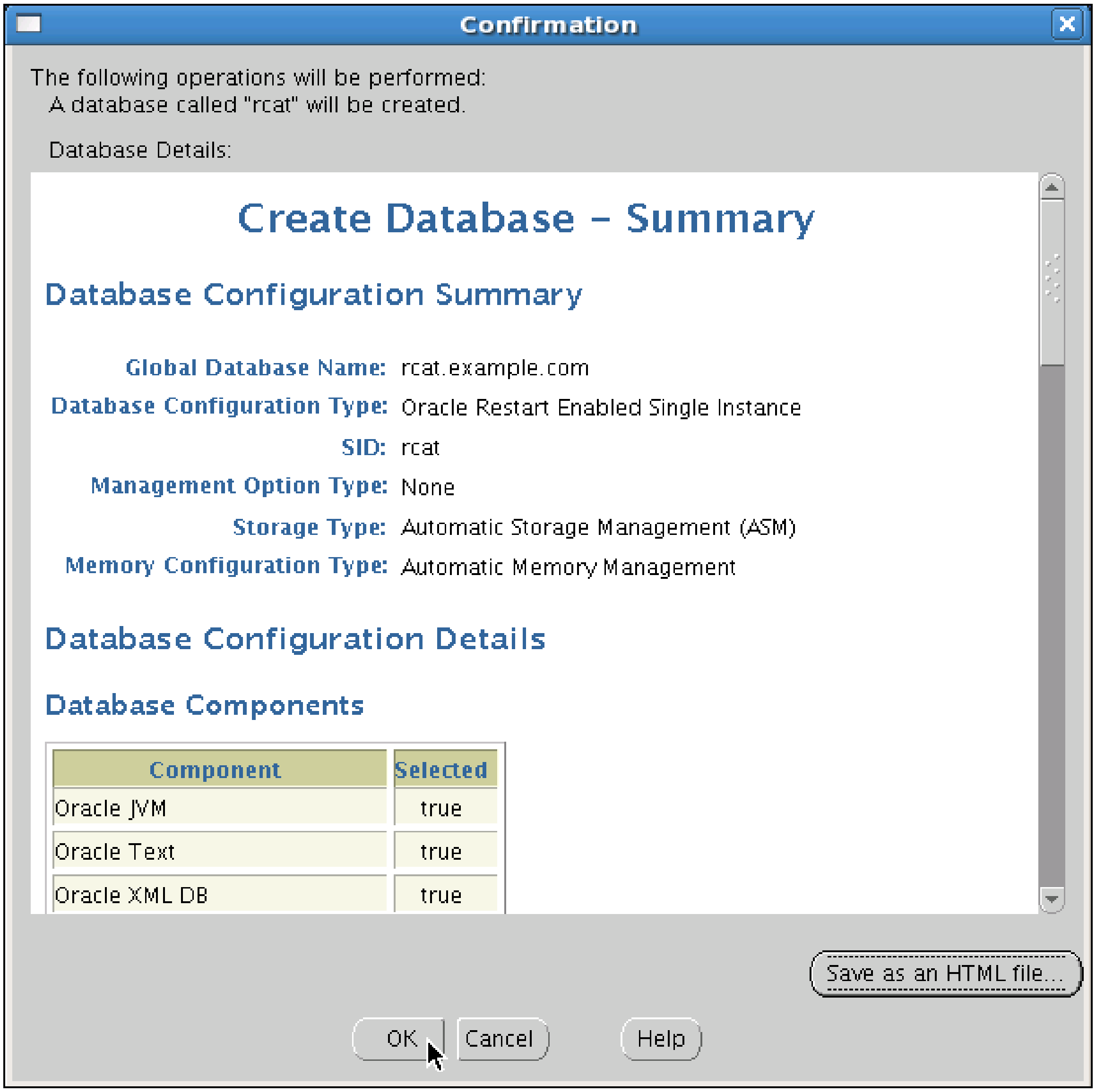


17) On the Creation Options page, make sure that Create Database is the only option selected, and then click Finish.

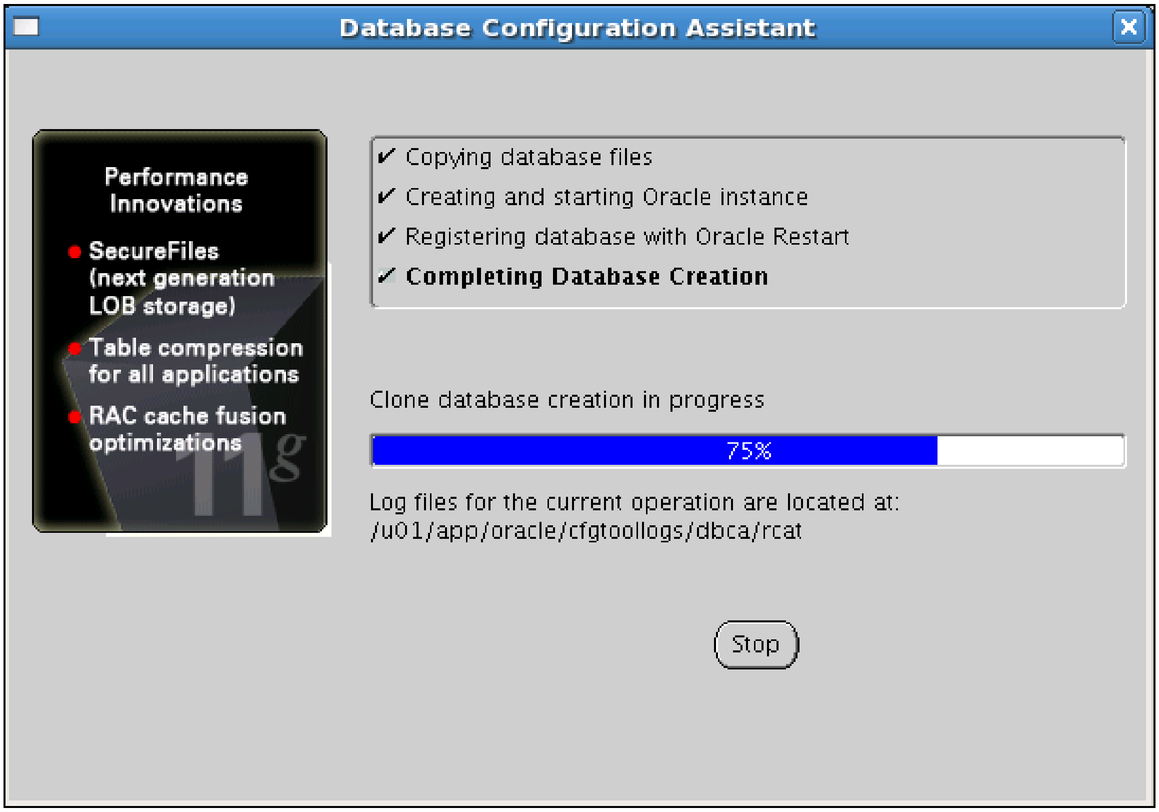
Practice 3-1: Creating a Recovery Catalog (continued)



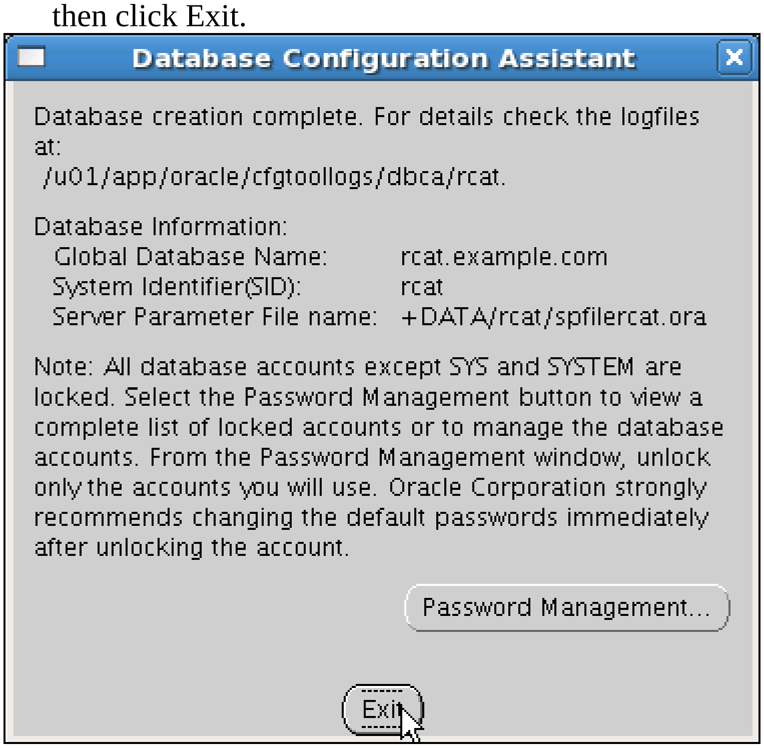
18) Review the Confirmation page, and then click OK to start the database creation process.



19) The Database Creation Assistant now creates the database.



20) When the Database Configuration Assistant finishes running, view the final page, and then click Exit.



Practice 3-2: Creating the Recovery Catalog Owner

In this practice, you configure the recovery catalog database with a user ID and appropriate privileges, and register a database.

1) Use SQL*Plus to configure the recovery catalog database. Connect to it as SYS to the RCAT database.

```
$ sqlplus sys as sysdba

Enter password: oracle_4U@rcat   <<<not displayed

SQL>
```

2) Create a tablespace named RCAT to hold the repository data. Make it 15 MB in size.

```
SQL> CREATE TABLESPACE rcat DATAFILE '+DATA/rcat01.dbf' SIZE 15M;

Tablespace created.

SQL>
```

3) Create a user who will own the recovery catalog data. Name the user RCATOWNER. The default tablespace should be the RCAT tablespace, and the user should have unlimited quota on that tablespace.

```
SQL> CREATE USER rcatowner IDENTIFIED BY "oracle_4U"
  2  DEFAULT TABLESPACE rcat QUOTA unlimited on rcat;

User created.

SQL>
```

4) Grant the RECOVERY_CATALOG_OWNER role to the RCATOWNER user.

```
SQL> GRANT recovery_catalog_owner to rcatowner;

Grant succeeded.
SQL> exit
```

.

Practice 3-3: Creating the Recovery Catalog

In this practice, you create the recovery catalog inside the recovery catalog database you have prepared.

1) Connect to the recovery catalog database using RMAN. Log in as the recovery catalog owner you just created.

```
$ rman catalog rcatowner@rcat

Recovery Manager: Release 11.2.0.1.0 - Production on Fri Jul
10 19:54:04 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates.  All
rights reserved.

recovery catalog database Password:oracle_4U <<<not displayed
connected to recovery catalog database

RMAN>
```

2) Create the recovery catalog. This command may take several minutes to complete.

```
RMAN> create catalog;

recovery catalog created

RMAN> exit
```

Practice 3-4: Registering a Database in the Recovery Catalog

In this practice, you register the ORCL database in the recovery catalog that you have just created.

1) Set up the environment for the ORCL database.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
```

2) Connect to the target database (to be registered) and the recovery catalog database using RMAN.

```
$ rman target / catalog rcatowner@rcat

Recovery Manager: Release 11.2.0.1.0 - Production on Tue Jul
21 15:30:32 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates.  All
rights reserved.

connected to target database: ORCL (DBID=1220535480)
recovery catalog database Password:oracle_4U <<<not displayed
connected to recovery catalog database

RMAN>
```

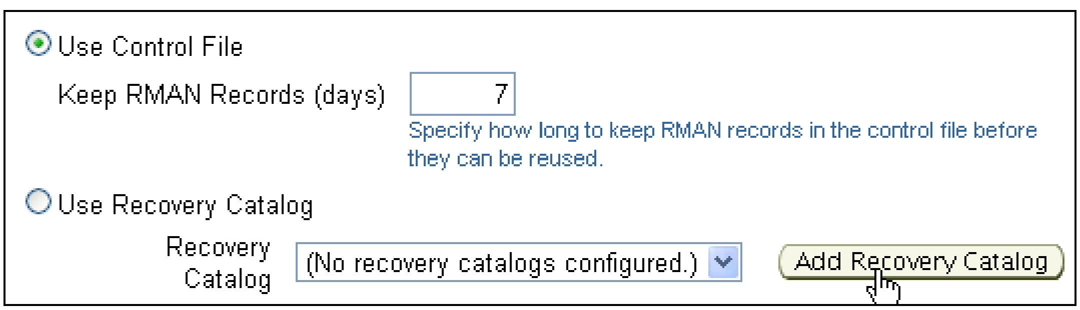
3) Register the database in the catalog.

```
RMAN> register database;

database registered in recovery catalog
starting full resync of recovery catalog
full resync complete

RMAN> exit
```

4) Register the recovery catalog to be used by Enterprise Manager.

a) In EM, navigate to Availability > Recovery Catalog Settings.

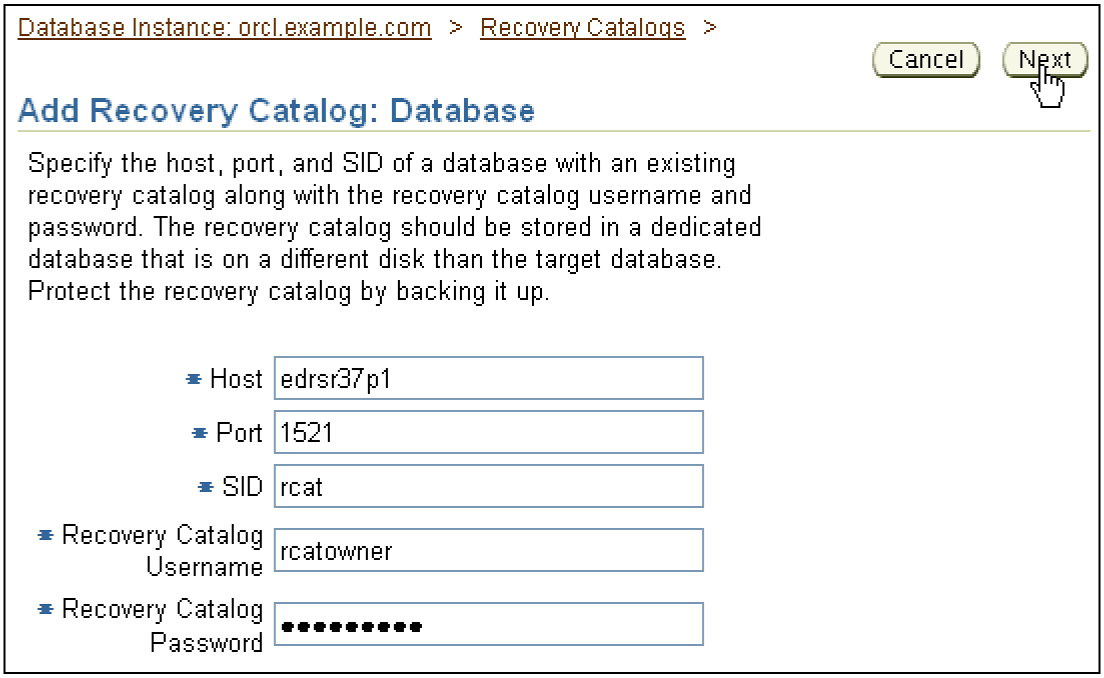
b) Click Add Recovery Catalog.



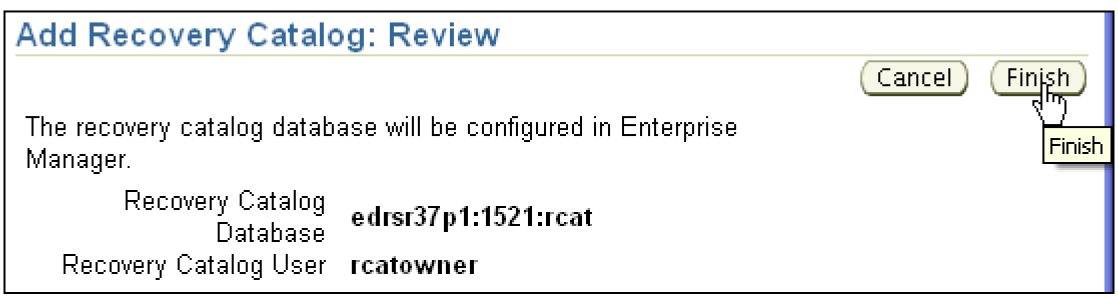
c) Enter the following information, then click Next:

Practice 3-4: Registering a Database in the Recovery Catalog (continued)

Host:	*<your_hostname>*
Port:	**1521**
SID:	**rcat**
Recovery Catalog Username:	**rcatowner**
Recovery Catalog Password:	**oracle_4U**



d) On the Review page, click Finish.

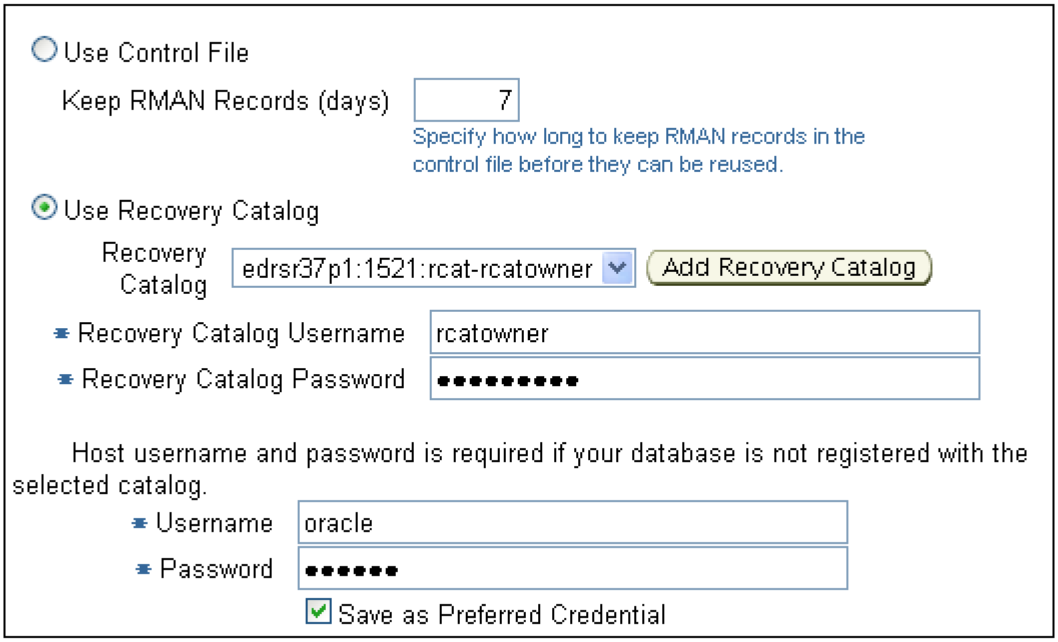


e) Back on the Recovery Catalog Settings page, select Use Recovery Catalog. Make sure that the recovery catalog you just registered is selected in the drop-down list, and enter the following values:

Recovery Catalog Username:	**rcatowner**
Recovery Catalog Password:	**oracle_4U**

Username:	**oracle**
Password:	**oracle**

Practice 3-4: Registering a Database in the Recovery Catalog (continued)



f) Click Save as Preferred Credential and then click OK.

Practice 3-5: Backing up the Recovery Catalog

In this practice, you configure the retention policy for the recovery catalog and back up your recovery catalog itself.

1) Set up the environment for the `ORCL` database.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
```

2) Start RMAN, use the recovery catalog database as the target, with no catalog specified.

```
$ rman target sys@rcat

target database Password: oracle_4U <<< not displayed
connected to target database: RCAT (DBID= 464959795)

RMAN>
```

3) Make sure that the retention policy for the recovery catalog is set to redundancy greater than 1. If it is not, set it to at least 2.

```
RMAN> show retention policy;

using target database control file instead of recovery catalog
RMAN configuration parameters for database with db_unique_name
RCAT are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default

RMAN> configure retention policy to redundancy 2;

new RMAN configuration parameters:
CONFIGURE RETENTION POLICY TO REDUNDANCY 2;
new RMAN configuration parameters are successfully stored

RMAN>
```

4) Try to back up the database.

```
RMAN> backup database;

Starting backup at 18-JUL-07
ORACLE error from target database:
ORA-00258: manual archiving in NOARCHIVELOG mode must identify
log

using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=137 device type=DISK
specification does not match any archived log in the recovery
catalog
backup cancelled because all files were skipped
```

```
Finished backup at 18-JUL-07

Starting backup at 18-JUL-07
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
RMAN-00571:
===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS
===============
RMAN-00571:
===========================================================
RMAN-03002: failure of backup plus archivelog command at
07/18/2007 22:55:45
ORA-19602: cannot backup or copy active file in NOARCHIVELOG
mode

RMAN>
```

Note that you cannot back up an open database that is not in ARCHIVELOG mode. The recovery catalog database should be run in ARCHIVELOG mode for maximum availability.

5) Exit RMAN.

```
RMAN> exit
```

6) First, set up the environment for the RCAT database and then run the rcat_to_archivelog.sh script to change the recovery catalog database to run in ARCHIVELOG mode.

```
$ . oraenv
ORACLE_SID = [orcl] ? rcat

$ cd ~/labs
$ ./rcat_to_archivelog.sh
Database closed.
Database dismounted.
ORACLE instance shut down.
ORACLE instance started.

Total System Global Area  263639040 bytes
Fixed Size                  1299164 bytes
Variable Size             230690084 bytes
Database Buffers           25165824 bytes
Redo Buffers                6483968 bytes
Database mounted.

Database altered.
Database altered.
$
```

7) Set up the environment for the ORCL database.

```
$ . oraenv
ORACLE_SID = [rcat] ? orcl
```

8) Log in to RMAN again, as in the preceding step.

```
$ rman target sys@rcat

target database Password: oracle_4U <<< not displayed
connected to target database: RCAT (DBID= 464959795)

RMAN>
```

9) Back up the recovery catalog database. This time the operation should be successful.

```
RMAN> backup database;

Starting backup at 21-JUL-09
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=40 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001 name=+DATA/rcat/system01.dbf
input datafile file number=00002 name=+DATA/rcat/sysaux01.dbf
input datafile file number=00003 name=+DATA/rcat/undotbs01.dbf
input datafile file number=00005 name=+DATA/rcat01.dbf
input datafile file number=00004 name=+DATA/rcat/users01.dbf
channel ORA_DISK_1: starting piece 1 at 21-JUL-09
channel ORA_DISK_1: finished piece 1 at 21-JUL-09
piece
handle=+FRA/rcat/backupset/2009_07_21/nnndf0_tag20090721t15532
5_0.263.692812405 tag=TAG20090721T155325 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time:
00:02:37
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current control file in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 21-JUL-09
channel ORA_DISK_1: finished piece 1 at 21-JUL-09
piece
handle=+FRA/rcat/backupset/2009_07_21/ncsnf0_tag20090721t15532
5_0.264.692812569 tag=TAG20090721T155325 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time:
00:00:03
Finished backup at 21-JUL-09

RMAN> exit
```

Practice 3-5: Backing up the Recovery Catalog (continued)

10) Run the `disable_asynch_io.sh` script to disable asynchronous input/output (I/O). After altering the system, the database is restarted. This improves performance on the following labs:

```
$ ~/labs/disable_asynch_io.sh
Setting up ORCL environment.
ORACLE_SID = [orcl] ? The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle

SQL*Plus: Release 11.2.0.1.0 Production on Tue Jul 21 15:58:51
2009

Copyright (c) 1982, 2009, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining
and Real Application Testing options

SQL>
System altered.

SQL>
System altered.

SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> ORACLE instance started.

Total System Global Area  481259520 bytes
Fixed Size                  1337324 bytes
Variable Size             360712212 bytes
Database Buffers          113246208 bytes
Redo Buffers                5963776 bytes
Database mounted.
Database opened.
SQL> Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining
and Real Application Testing options
```

Practice 4-1: Setting the Date and Time Format for RMAN

In this practice, you set the date/time format that RMAN uses for displaying timestamps.

1) Set the `NLS_LANG` and `NLS_DATE_FORMAT` variables such that RMAN includes time information in any timestamp values. Add the following two lines to the `~oracle/.bashrc` file. Then exit all of your terminal windows. This ensures that when you create new ones, these settings will be in effect.

```
export NLS_LANG=american_america.al32utf8
export NLS_DATE_FORMAT="yyyy-mm-dd:hh24:mi:ss"
```

2) Start a new terminal window, and verify the settings by starting RMAN and listing the backups of the recovery catalog database.

   a) Set up the environment for the `ORCL` database.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
```

   b) Log in to RMAN.

```
$ rman target sys@rcat

target database Password: oracle_4U <<< not displayed
connected to target database: RCAT (DBID= 464959795)
```

   c) List the backups, and note the timestamp format.

```
RMAN> list backup;
using target database control file instead of recovery catalog

List of Backup Sets
===================

BS Key  Type LV Size       Device Type Elapsed Time Completion
Time
------- ---- -- ---------- ----------- ------------ ----------
---------
1       Full    9.36M      DISK        00:00:06     2009-07-
21:15:42:05
        BP Key: 1   Status: AVAILABLE  Compressed: NO  Tag:
TAG20090721T154159
        Piece Name:
+FRA/rcat/backupset/2009_07_21/ncsnf0_tag20090721t154159_0.262
.692811721
  SPFILE Included: Modification time: 2009-07-21:15:32:47
  SPFILE db_unique_name: RCAT
```

Practice 4-1: Setting the Date and Time Format for RMAN (continued)

```
  Control File Included: Ckp SCN: 803770       Ckp time: 2009-
07-21:15:41:59

.
.
.
RMAN>
```

Note: Because the output of the RMAN commands can be quite long, consider using the RMAN `SPOOL LOG` command to direct the output to your specified file.

Practice 4-2: Enabling Control File Autobackup

In this practice, you configure RMAN to back up the control file and SPFILE each time it takes a backup of anything in the RCAT database.

1) In the same recovery catalog RMAN session, make sure that control file autobackup is enabled.

```
RMAN> show controlfile autobackup;

RMAN configuration parameters for database with db_unique_name
RCAT are:
CONFIGURE CONTROLFILE AUTOBACKUP OFF; # default

RMAN>
```

2) Enable control file autobackup.

```
RMAN> configure controlfile autobackup on;

new RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters are successfully stored

RMAN>
```

3) Verify that it is enabled by backing up the archive logs for the recovery catalog. Then exit RMAN.

```
RMAN> backup archivelog all;

Starting backup at 2009-07-21:16:22:10
current log archived
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=47 device type=DISK
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=5 RECID=1 STAMP=692814135
channel ORA_DISK_1: starting piece 1 at 2009-07-21:16:22:18
channel ORA_DISK_1: finished piece 1 at 2009-07-21:16:22:21
piece
handle=+FRA/rcat/backupset/2009_07_21/annnf0_tag20090721t16221
6_0.266.692814139 tag=TAG20090721T162216 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time:
00:00:04
Finished backup at 2009-07-21:16:22:21

Starting Control File and SPFILE Autobackup at 2009-07-
21:16:22:21
piece
handle=+FRA/rcat/autobackup/2009_07_21/s_692814141.267.6928141
45 comment=NONE
Finished Control File and SPFILE Autobackup at 2009-07-
21:16:22:28
```

Practice 4-2: Enabling Control File Autobackup (continued)

```
RMAN> exit
```

Note that the control file and SPFILE are automatically backed up now.

Practice 4-3: Configuring Devices for Backup

In this practice, you configure a tape device for use in making backups.
Note: This channel definition is used in the RMAN monitoring and tuning practice. Therefore, steps 1-4 are mandatory; step 5 (performing a backup) is recommended, but optional.

1) Make sure the SID variable is set to `ORCL`.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
```

2) Start RMAN by connecting to the `ORCL` as the target database and using the recovery catalog database.

```
$ rman target / catalog rcatowner@rcat

recovery catalog database Password: oracle_4U <<<not displayed
RMAN>
```

3) Show all configuration settings to see whether there are any tape devices defined.

```
RMAN> show all;

RMAN configuration parameters for database with db_unique_name
ORCL are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP ON;
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK
TO '%F'; # default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO
BACKUPSET; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; #
default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1;
# default
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE OFF; # default
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE
'DEFAULT' OPTIMIZE FOR LOAD TRUE ; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO
'/u01/app/oracle/product/11.2.0/dbhome_1/dbs/snapcf_orcl.f'; #
default

RMAN>
```

Note that there are no tape devices.

4) Define a channel for a tape device that uses the test interface. This actually writes to disk in the `/tape` directory.

Practice 4-3: Configuring Devices for Backup (continued)

```
RMAN> configure channel device type sbt
2> parms='SBT_LIBRARY=oracle.disksbt,ENV=(BACKUP_DIR=/tape)';

new RMAN configuration parameters:
CONFIGURE CHANNEL DEVICE TYPE 'SBT_TAPE' PARMS
'SBT_LIBRARY=oracle.disksbt,ENV=(BACKUP_DIR=/tape)';
new RMAN configuration parameters are successfully stored
starting full resync of recovery catalog
full resync complete

RMAN>
```

5) Perform a backup to the tape device to make sure it works. Back up the USERS tablespace. Then exit RMAN.

```
RMAN> backup device type sbt tablespace users;

Starting backup at 2009-07-21:16:32:09
allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=53 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: WARNING: Oracle Test Disk API
channel ORA_SBT_TAPE_1: starting full datafile backup set
channel ORA_SBT_TAPE_1: specifying datafile(s) in backup set
input datafile file number=00004
name=+DATA/orcl/datafile/users.259.692754561
channel ORA_SBT_TAPE_1: starting piece 1 at 2009-07-
21:16:32:10
channel ORA_SBT_TAPE_1: finished piece 1 at 2009-07-
21:16:32:11
piece handle=05kkn0sa_1_1 tag=TAG20090721T163210 comment=API
Version 2.0,MMS Version 8.1.3.0
channel ORA_SBT_TAPE_1: backup set complete, elapsed time:
00:00:01
Finished backup at 2009-07-21:16:32:11

Starting Control File and SPFILE Autobackup at 2009-07-
21:16:32:11
piece handle=c-1220535480-20090721-02 comment=API Version
2.0,MMS Version 8.1.3.0
Finished Control File and SPFILE Autobackup at 2009-07-
21:16:32:16

RMAN> exit
```

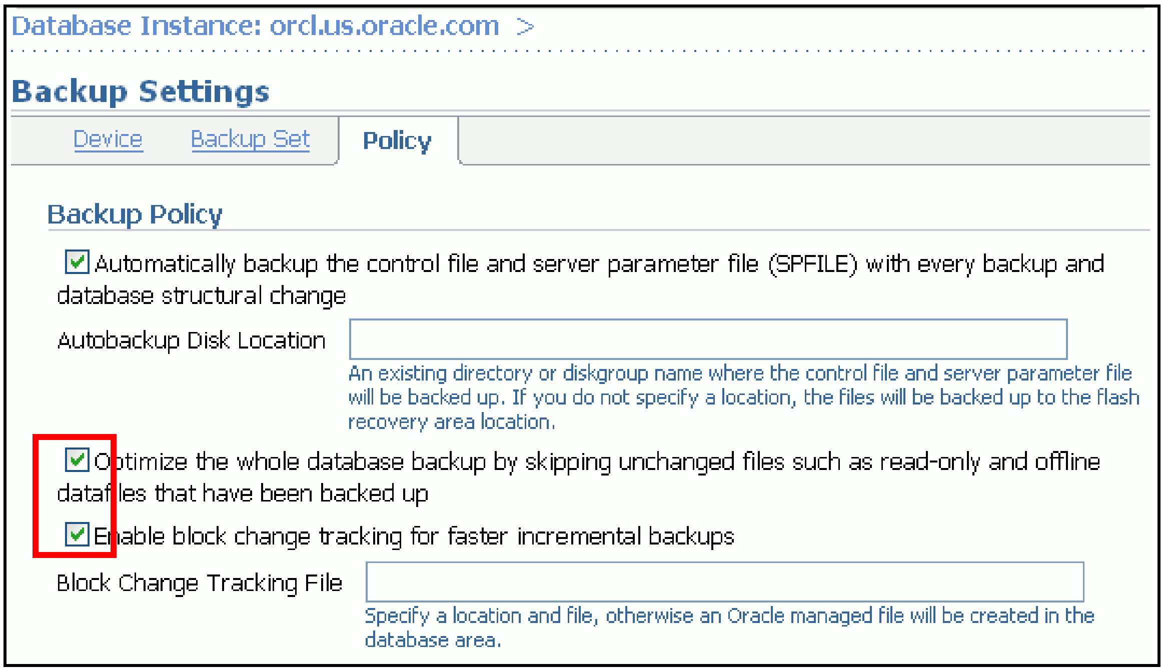
Note that it uses a channel called ORA_SBT_TAPE_1 to perform the backup.

Practice 5-1: Creating Fast Incremental Backups

In this practice, you enable block change tracking so that you can make incremental backups more quickly.

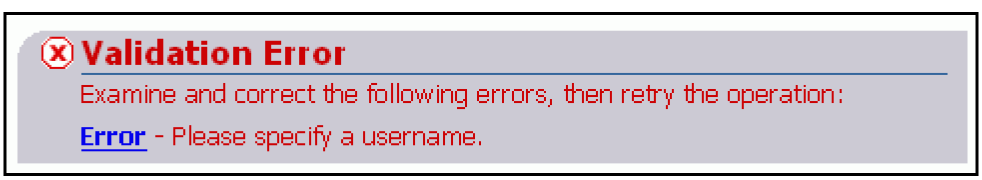
1) Use Enterprise Manager Database Control to configure backup optimization and enable block change tracking. Allow a default block change tracking file to be used.

   a) From the Database home page of EM, navigate to Availability > Backup Settings > Policy, and enable backup optimization and block change tracking.



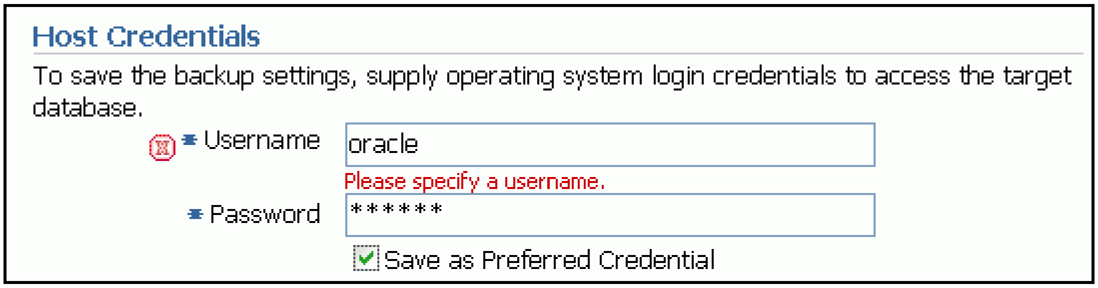
   b) Click OK.

   c) If you see the following error, you have not entered the operating system credentials. Continue with the following steps if that is the case.



   d) Scroll to the bottom of the page, and enter `oracle` as username and `oracle` as password. Also, select the option to save this as the preferred credential.

Practice 5-1: Creating Fast Incremental Backups (continued)



e) Click OK.

2) Make sure that the `ORACLE_SID` variable is set to `ORCL`.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
```

3) Log in to RMAN and create an incremental level 0 backup.

```
$ rman target / catalog rcatowner@rcat

recovery catalog database Password: oracle_4U <<<not displayed

RMAN> backup incremental level 0 database plus archivelog;

starting full resync of recovery catalog
full resync complete
.
.
.
piece
handle=+FRA/orcl/backupset/2009_07_21/annnf0_tag20090721t19055
7_0.273.692823957 tag=TAG20090721T190557 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time:
00:00:03
Finished backup at 21-JUL-09

Starting Control File and SPFILE Autobackup at 21-JUL-09
piece
handle=+FRA/orcl/autobackup/2009_07_21/s_692823962.274.6928239
65 comment=NONE
Finished Control File and SPFILE Autobackup at 21-JUL-09

RMAN>
```

4) Now that you have a level 0 incremental backup, you can take advantage of block change tracking to make fast incremental backups. Take a level 1 incremental backup.

```
RMAN> backup incremental level 1 database plus archivelog;

Starting backup at 21-JUL-09
current log archived
using channel ORA_DISK_1
.
.
```

Practice 5-1: Creating Fast Incremental Backups (continued)

```
.
Finished Control File and SPFILE Autobackup at 21-JUL-09

RMAN>
```

5) In another terminal window, make sure you are at the `~/labs` directory.

```
$ cd ~/labs
```

6) Review and execute the `query_block_count.sh` script to query `V$BACKUP_DATAFILE` to see how many of the blocks were read for creating the level 1 incremental backup.

```
$ cat query_block_count.sh
export ORACLE_SID=orcl
sqlplus / as sysdba <<-EOI
  select file#, avg(datafile_blocks),
    avg(blocks_read),
    avg(blocks_read/datafile_blocks) * 100 as
PCT_READ_FOR_BACKUP,
    avg(blocks)
  from v\$backup_datafile
  where used_change_tracking = 'YES'
    and incremental_level > 0
    group by file#;
  quit
EOI

$./query_block_count.sh

SQL>   2    3    4    5    6    7    8
     FILE# AVG(DATAFILE_BLOCKS) AVG(BLOCKS_READ)
PCT_READ_FOR_BACKUP AVG(BLOCKS)
---------- -------------------- ---------------- -------------
------ -----------
         1                87040              173
.198759191          34
         2                75520              957
1.26721398         344
         5                12800                1
.0078125           1
         4                  640                1
.15625           1
         3                12800              311
2.4296875          96
$
```

Note that the percentage of blocks read for making the backup is very low in most cases, and sometimes very close to zero.

Practice 5-2: Cross-Checking Backups

In this practice, you cross-check backups against the recovery catalog, identifying and deleting any backups that are obsolete.

1) Make sure that you are at the `~/labs` directory, and that the `ORACLE_SID` variable is set to `ORCL`.

```
$ cd ~/labs
$ . oraenv
ORACLE_SID = [orcl] ? orcl
```

2) Use the RMAN session from the previous practice or start RMAN. Make sure that you connect to both your target database and the recovery catalog database.

```
$ rman target / catalog rcatowner@rcat

connected to target database: ORCL (DBID=1220535480)
recovery catalog database Password: oracle_4U <<<not displayed
connected to recovery catalog database

RMAN>
```

3) List backups of data file 5, noting the entry corresponding to the `example` data file.

```
RMAN> list backup of datafile 5;

BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ -------------------
37      Full    68.99M     DISK        00:00:13     2009-07-21:14:54:11
        BP Key: 40   Status: AVAILABLE  Compressed: NO  Tag:
TAG20090721T145358
        Piece Name:
+FRA/orcl/backupset/2009_07_21/nnndf0_tag20090721t145358_0.260.692808839
  List of Datafiles in backup set 37
  File LV Type Ckp SCN    Ckp Time            Name
  ---- -- ---- ---------- ------------------- ----
  5       Full 908595     2009-07-21:14:53:59
+DATA/orcl/datafile/example.265.692754837

BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ -------------------
172     Incr 0  1.07G      DISK        00:02:47     2009-07-21:19:05:44
        BP Key: 176   Status: AVAILABLE  Compressed: NO  Tag:
TAG20090721T190255
        Piece Name:
+FRA/orcl/backupset/2009_07_21/nnndn0_tag20090721t190255_0.271.692823777
  List of Datafiles in backup set 172
  File LV Type Ckp SCN    Ckp Time            Name
  ---- -- ---- ---------- ------------------- ----
  5    0  Incr 924152     2009-07-21:19:02:57
+DATA/orcl/datafile/example.265.692754837

BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ -------------------
237     Incr 1  3.74M      DISK        00:00:02     2009-07-21:19:10:35
```

```
        BP Key: 241   Status: AVAILABLE  Compressed: NO  Tag:
TAG20090721T191031
        Piece Name:
+FRA/orcl/backupset/2009_07_21/nnndn1_tag20090721t191031_0.277.692824233
  List of Datafiles in backup set 237
  File LV Type Ckp SCN    Ckp Time            Name
  ---- -- ---- ---------- ------------------- ----
  5    1  Incr 924953     2009-07-21:19:10:33
+DATA/orcl/datafile/example.265.692754837

RMAN>
```

Take note of the only backup that is a full backup of data file 5. Also note that the "Piece Name" is the name of the ASM file where the backup file resides.

4) In an OS terminal window, delete the backup set file. Start a separate terminal session, so the RMAN session can stay connected. Use the +ASM environment variables, review and execute the `rm_asm_file.sh` script to do this. Supply YOUR full "Piece Name" as an argument to the script.

```
$ . oraenv
ORACLE_SID = [orcl] ? +ASM

$ cat rm_asm_file.sh
export ORACLE_SID=+ASM
asmcmd <<-EOI
        rm -f $1
        exit
EOI
Exit

 ./rm_asm_file.sh \
> +FRA/orcl/backupset/2009_07_21/nnndf0_tag20090721t145358_0.260.692808839
ASMCMD> ASMCMD> $
$
```

5) In your still connected RMAN session, list the backups again, and look for the backup that corresponds to the `example` data file.

```
RMAN> list backup of datafile 5;


List of Backup Sets
===================

BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ -------------------
37      Full    68.99M     DISK        00:00:13     2009-07-21:14:54:11
        BP Key: 40   Status: AVAILABLE  Compressed: NO  Tag:
TAG20090721T145358
        Piece Name:
+FRA/orcl/backupset/2009_07_21/nnndf0_tag20090721t145358_0.260.692808839
  List of Datafiles in backup set 37
  File LV Type Ckp SCN    Ckp Time            Name
  ---- -- ---- ---------- ------------------- ----
```

```
  5       Full 908595     2009-07-21:14:53:59
+DATA/orcl/datafile/example.265.692754837

BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ -------------------
172     Incr 0  1.07G      DISK        00:02:47     2009-07-21:19:05:44
        BP Key: 176   Status: AVAILABLE  Compressed: NO  Tag:
TAG20090721T190255
        Piece Name:
+FRA/orcl/backupset/2009_07_21/nnndn0_tag20090721t190255_0.271.692823777
  List of Datafiles in backup set 172
  File LV Type Ckp SCN    Ckp Time            Name
  ---- -- ---- ---------- ------------------- ----
  5    0  Incr 924152     2009-07-21:19:02:57
+DATA/orcl/datafile/example.265.692754837

BS Key  Type LV Size       Device Type Elapsed Time Completion Time
------- ---- -- ---------- ----------- ------------ -------------------
237     Incr 1  3.74M      DISK        00:00:02     2009-07-21:19:10:35
        BP Key: 241   Status: AVAILABLE  Compressed: NO  Tag:
TAG20090721T191031
        Piece Name:
+FRA/orcl/backupset/2009_07_21/nnndn1_tag20090721t191031_0.277.692824233
  List of Datafiles in backup set 237
  File LV Type Ckp SCN    Ckp Time            Name
  ---- -- ---- ---------- ------------------- ----
  5    1  Incr 924953     2009-07-21:19:10:33
+DATA/orcl/datafile/example.265.692754837

RMAN>
```

Note that the backups are still listed as before. That is because the repository is not aware of the fact that the file for the backup piece is missing.

6) Cross-check the backup sets to make the recovery catalog aware of the missing backup file.

```
RMAN> crosscheck backupset;

allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=48 device type=DISK
allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=57 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: WARNING: Oracle Test Disk API
crosschecked backup piece: found to be 'EXPIRED'
backup piece
handle=+FRA/orcl/backupset/2009_07_21/nnndf0_tag20090721t14535
8_0.260.692808839 RECID=1 STAMP=692808839
crosschecked backup piece: found to be 'AVAILABLE'.
.
.
.

RMAN>
```

Note that the backup in question has expired.

Practice 5-2: Cross-Checking Backups (continued)

7) Delete all the expired backup sets.

```
RMAN> delete expired backupset;

using channel ORA_DISK_1
using channel ORA_SBT_TAPE_1

List of Backup Pieces
BP Key  BS Key  Pc# Cp# Status      Device Type Piece Name
------- ------- --- --- ----------- ----------- ----------
40      37      1   1   EXPIRED     DISK
+FRA/orcl/backupset/2009_07_21/nnndf0_tag20090721t145358_0.260
.692808839

Do you really want to delete the above objects (enter YES or
NO)? YES
deleted backup piece
backup piece
handle=+FRA/orcl/backupset/2009_07_21/nnndf0_tag20090721t14535
8_0.260.692808839 RECID=1 STAMP=692808839
Deleted 1 EXPIRED objects

RMAN>
```

8) List the backup sets again. See that the backup you just deleted is no longer listed.

```
RMAN> list backup of datafile 5;


List of Backup Sets
===================


BS Key  Type LV Size       Device Type Elapsed Time Completion
Time
------- ---- -- ---------- ----------- ------------ ----------
---------
172     Incr 0  1.07G      DISK        00:02:47     2009-07-
21:19:05:44
        BP Key: 176   Status: AVAILABLE  Compressed: NO  Tag:
TAG20090721T190255
        Piece Name:
+FRA/orcl/backupset/2009_07_21/nnndn0_tag20090721t190255_0.271
.692823777
  List of Datafiles in backup set 172
  File LV Type Ckp SCN    Ckp Time            Name
  ---- -- ---- ---------- ------------------- ----
  5    0  Incr 924152     2009-07-21:19:02:57
+DATA/orcl/datafile/example.265.692754837

BS Key  Type LV Size       Device Type Elapsed Time Completion
Time
```

Practice 5-2: Cross-Checking Backups (continued)

```
------- ---- -- ---------- ----------- ------------ ----------
---------
237     Incr 1  3.74M      DISK        00:00:02     2009-07-
21:19:10:35
        BP Key: 241   Status: AVAILABLE  Compressed: NO  Tag:
TAG20090721T191031
        Piece Name:
+FRA/orcl/backupset/2009_07_21/nnndn1_tag20090721t191031_0.277
.692824233
  List of Datafiles in backup set 237
  File LV Type Ckp SCN    Ckp Time            Name
  ---- -- ---- ---------- ------------------- ----
  5    1  Incr 924953     2009-07-21:19:10:33
+DATA/orcl/datafile/example.265.692754837

RMAN>
```

Practice 5-3: Listing Backup Files and Creating Archival Backup

In this practice, you list backup files.

1) Continue to use the RMAN session from the previous practice, and generate a report of all the obsolete backup files.

```
RMAN> report obsolete;

RMAN retention policy will be applied to the command
RMAN retention policy is set to redundancy 1
Report of obsolete backups and copies
Type                 Key    Completion Time    Filename/Handle
-------------------- ------ ------------------ ---------------
-----
Backup Set           38     2009-07-21:14:54:16
  Backup Piece       41     2009-07-21:14:54:16
+FRA/orcl/autobackup/2009_07_21/s_692808854.261.692808857
Backup Set           68     2009-07-21:16:29:17
  Backup Piece       71     2009-07-21:16:29:17 03kkn0mt_1_1
Backup Set           77     2009-07-21:16:29:22
  Backup Piece       79     2009-07-21:16:29:22 c-1220535480-
20090721-01
Backup Set           93     2009-07-21:16:32:10
  Backup Piece       96     2009-07-21:16:32:10 05kkn0sa_1_1
Archive Log          151    2009-07-21:17:49:04
+FRA/orcl/archivelog/2009_07_21/thread_1_seq_9.268.692819337
Backup Set           105    2009-07-21:16:32:14
  Backup Piece       107    2009-07-21:16:32:14 c-1220535480-
20090721-02
Archive Log          163    2009-07-21:19:02:49
+FRA/orcl/archivelog/2009_07_21/thread_1_seq_10.269.692823767
Backup Set           171    2009-07-21:19:02:53
  Backup Piece       175    2009-07-21:19:02:53
+FRA/orcl/backupset/2009_07_21/annnf0_tag20090721t190251_0.270
.692823773
Backup Set           205    2009-07-21:19:06:06
  Backup Piece       207    2009-07-21:19:06:06
+FRA/orcl/autobackup/2009_07_21/s_692823962.274.692823965
RMAN>
```

Your list of files might look different. - What can you deduce about these backup files, given that the retention policy is set to `REDUNDANCY 1`?

2) Which backups would be considered obsolete if the retention policy were set to redundancy of 2, instead of 1? Answer this without changing the retention policy.

```
RMAN> report obsolete redundancy 2;

Report of obsolete backups and copies
Type                 Key    Completion Time    Filename/Handle
-------------------- ------ ------------------ ---------------
Backup Set           38     2009-07-21:14:54:16
```

Practice 5-3: Listing Backup Files and Creating Archival Backup (continued)

```
  Backup Piece       41     2009-07-21:14:54:16
+FRA/orcl/autobackup/2009_07_21/s_692808854.261.692808857
Backup Set           68     2009-07-21:16:29:17
  Backup Piece       71     2009-07-21:16:29:17 03kkn0mt_1_1
Backup Set           77     2009-07-21:16:29:22
  Backup Piece       79     2009-07-21:16:29:22 c-1220535480-
20090721-01
Backup Set           105    2009-07-21:16:32:14
  Backup Piece       107    2009-07-21:16:32:14 c-1220535480-
20090721-02

RMAN>
```

Note that far fewer backups are obsolete in this hypothetical report. That is because with an increased redundancy, more of the backups are required. This command is run without affecting the retention policy.

3) Delete all obsolete backups.

```
RMAN> delete noprompt obsolete;

RMAN retention policy will be applied to the command
RMAN retention policy is set to redundancy 1
using channel ORA_DISK_1
using channel ORA_SBT_TAPE_1
Deleting the following obsolete backups and copies:
Type                 Key    Completion Time    Filename/Handle
-------------------- ------ ------------------ ---------------
-----
Backup Set           38     2009-07-21:14:54:16
  Backup Piece       41     2009-07-21:14:54:16
+FRA/orcl/autobackup/2009_07_21/s_692808854.261.692808857
Backup Set           68     2009-07-21:16:29:17
  Backup Piece       71     2009-07-21:16:29:17 03kkn0mt_1_1
Backup Set           77     2009-07-21:16:29:22
.
.
.

deleted backup piece
backup piece handle=05kkn0sa_1_1 RECID=5 STAMP=692814730
deleted backup piece
backup piece handle=c-1220535480-20090721-02 RECID=6
STAMP=692814734
Deleted 4 objects

RMAN>
```

Practice 5-3: Listing Backup Files and Creating Archival Backup (continued)

Note: Your number of objects may be different, depending on repetitions of practice steps.

4) List the backup archivelog files that could be used to recover to a point in time one hour ago.
   Note: You may need to adjust the following SYSDATE expression to see some results that differ from the list command that does not use a SYSDATE qualifier at all. For example, try 'sysdate-05/60/24' to indicate 5 minutes ago.

```
RMAN> list backup of archivelog until time 'sysdate-1/24';


List of Backup Sets
===================


BS Key  Size       Device Type Elapsed Time Completion Time
------- ---------- ----------- ------------ ------------------
-
191     1.05M      DISK        00:00:01     2009-07-
21:19:05:58
        BP Key: 197   Status: AVAILABLE  Compressed: NO  Tag:
TAG20090721T190557
        Piece Name:
+FRA/orcl/backupset/2009_07_21/annnf0_tag20090721t190557_0.273
.692823957

  List of Archived Logs in backup set 191
  Thrd Seq     Low SCN    Low Time            Next SCN   Next
Time
  ---- ------- ---------- ------------------- ---------- -----
----
  1    11      924126     2009-07-21:19:02:46 924627     2009-
07-21:19:05:53

BS Key  Size       Device Type Elapsed Time Completion Time
------- ---------- ----------- ------------ ------------------
-
236     277.00K    DISK        00:00:01     2009-07-
21:19:10:29
        BP Key: 240   Status: AVAILABLE  Compressed: NO  Tag:
TAG20090721T191028
        Piece Name:
+FRA/orcl/backupset/2009_07_21/annnf0_tag20090721t191028_0.276
.692824229

  List of Archived Logs in backup set 236
  Thrd Seq     Low SCN    Low Time            Next SCN   Next
Time
```

Practice 5-3: Listing Backup Files and Creating Archival Backup (continued)

```
  ---- ------- ---------- ------------------- ---------- -----
----
  1    12      924627     2009-07-21:19:05:53 924928     2009-
07-21:19:10:26

BS Key  Size       Device Type Elapsed Time Completion Time
------- ---------- ----------- ------------ ------------------
-
258     7.50K      DISK        00:00:01     2009-07-
21:19:10:39
        BP Key: 264   Status: AVAILABLE  Compressed: NO  Tag:
TAG20090721T191037
        Piece Name:
+FRA/orcl/backupset/2009_07_21/annnf0_tag20090721t191037_0.279
.692824239

  List of Archived Logs in backup set 258
  Thrd Seq     Low SCN    Low Time            Next SCN   Next
Time
  ---- ------- ---------- ------------------- ---------- -----
----
  1    13      924928     2009-07-21:19:10:26 924960     2009-
07-21:19:10:36

RMAN>
```

5) List the image copies of the single data file in the USERS tablespace.

   a) Report on the schema to find out which data file number belongs to the USERS tablespace.

```
RMAN> report schema;

Report of database schema for database with db_unique_name
ORCL

List of Permanent Datafiles
===========================
File Size(MB) Tablespace           RB segs Datafile Name
---- -------- -------------------- ------- -------------------
-----
1    680      SYSTEM               YES
+DATA/orcl/datafile/system.256.692754557
2    590      SYSAUX               NO
+DATA/orcl/datafile/sysaux.257.692754559
3    100      UNDOTBS1             YES
+DATA/orcl/datafile/undotbs1.258.692754561
4    5        USERS                NO
+DATA/orcl/datafile/users.259.692754561
```

Practice 5-3: Listing Backup Files and Creating Archival Backup (continued)

```
5    100      EXAMPLE              NO
+DATA/orcl/datafile/example.265.692754837

List of Temporary Files
=======================
File Size(MB) Tablespace           Maxsize(MB) Tempfile Name
---- -------- -------------------- ----------- ---------------
-----
1    28       TEMP                 32767
+DATA/orcl/tempfile/temp.264.692754825
RMAN>
```

Note that the file in question is data file number 4.

b) List any image copies of data file number 4.

```
RMAN> list copy of datafile 4;

specification does not match any datafile copy in the repository

RMAN>
```

Note that, in this case, there is no image copy of this file.

6) Delete any data file 4 image copies that exist.
Note: You may not have any.

```
RMAN> delete copy of datafile 4;

released channel: ORA_DISK_1
released channel: ORA_SBT_TAPE_1
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=105 device type=DISK
List of Datafile Copies
=======================

Key     File S Completion Time     Ckp SCN    Ckp Time
------- ---- - ------------------- ---------- ----------------
---
115     4    A 2009-05-27:21:08:45 1917108    2009-05-
27:20:50:57
        Name: /tmp/users_copy.dat


Do you really want to delete the above objects (enter YES or
NO)? yes

RMAN> list copy of datafile 4;

specification does not match any datafile copy in the
repository
```

Practice 5-3: Listing Backup Files and Creating Archival Backup (continued)

```
RMAN>
```

7) To simplify your training environment, disable control file autobackup for the next few steps.

```
RMAN> configure controlfile autobackup off;

old RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP OFF;
new RMAN configuration parameters are successfully stored
starting full resync of recovery catalog
full resync complete

RMAN>
```

8) Make an image file backup of data file 4.

```
RMAN> backup as copy datafile 4;

Starting backup at 2009-07-21:20:27:25
released channel: ORA_SBT_TAPE_1
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile copy
input datafile file number=00004
name=+DATA/orcl/datafile/users.259.692754561
output file name=+FRA/orcl/datafile/users.274.692828845
tag=TAG20090721T202725 RECID=2 STAMP=692828849
channel ORA_DISK_1: datafile copy complete, elapsed time:
00:00:07
Finished backup at 2009-07-21:20:27:32

RMAN>
```

9) Make another image copy of the data file, but make this one an archival backup.

```
RMAN> backup as copy datafile 4 keep forever;

Starting backup at 2009-07-21:20:28:30
current log archived

using channel ORA_DISK_1
backup will never be obsolete
archived logs required to recover from this backup will be
backed up
channel ORA_DISK_1: starting datafile copy
input datafile file number=00004
name=+DATA/orcl/datafile/users.259.692754561
```

Practice 5-3: Listing Backup Files and Creating Archival Backup (continued)

```
RMAN-00571:
===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS
===============
RMAN-00571:
===========================================================
RMAN-03002: failure of backup command at 07/21/2009 20:28:34
ORA-19811: cannot have files in DB_RECOVERY_FILE_DEST with
keep attributes
RMAN>
```

Note that the backup command failed. This is because a backup with the `KEEP` attribute (an archival backup) cannot be written to the Flash Recovery Area. Allowing this has the potential of causing the Flash Recovery Area to quickly run out of space, forcing you to specify a different location.

Practice 5-3: Listing Backup Files and Creating Archival Backup (continued)

10) Create an archival backup with a `FORMAT` clause that causes the backup to be written to the `/tmp` directory.

```
RMAN> backup keep forever as copy datafile 4 format
'/tmp/bu_%d_%s_%p.dbf';

Starting backup at 2009-07-21:20:31:14
current log archived

using channel ORA_DISK_1
backup will never be obsolete
archived logs required to recover from this backup will be
backed up
channel ORA_DISK_1: starting datafile copy
input datafile file number=00004
name=+DATA/orcl/datafile/users.259.692754561
output file name=/tmp/bu_ORCL_17_1.dbf tag=TAG20090721T203114
RECID=3 STAMP=692829077
channel ORA_DISK_1: datafile copy complete, elapsed time:
00:00:03
.
.
.
piece handle=/tmp/bu_ORCL_20_1.dbf tag=TAG20090721T203114
comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time:
00:00:01
Finished backup at 2009-07-21:20:31:29

RMAN>
```

11) See if there are any obsolete backups. (If you repeat practice steps you might have obsolete ones.)

```
RMAN> report obsolete;

RMAN retention policy will be applied to the command
RMAN retention policy is set to redundancy 1
no obsolete backups found

RMAN>
```

12) Create another (nonarchive) backup of data file 4.

```
RMAN> backup as copy datafile 4;

Starting backup at 2009-07-21:20:33:52
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile copy
```

Practice 5-3: Listing Backup Files and Creating Archival Backup (continued)

```
input datafile file number=00004
name=+DATA/orcl/datafile/users.259.692754561
output file name=+FRA/orcl/datafile/users.261.692829233
tag=TAG20090721T203352 RECID=4 STAMP=692829233
channel ORA_DISK_1: datafile copy complete, elapsed time:
00:00:01
Finished backup at 2009-07-21:20:33:54

RMAN>
```

13) Report on the obsolete backups.

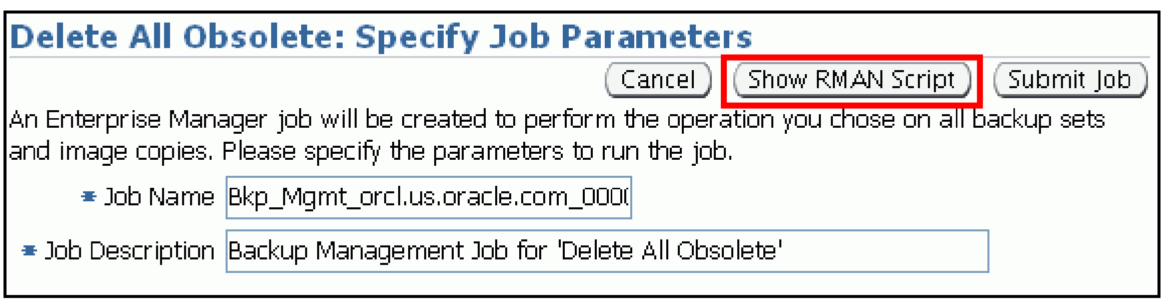
```
RMAN> report obsolete;

RMAN retention policy will be applied to the command
RMAN retention policy is set to redundancy 1
Report of obsolete backups and copies
Type                 Key    Completion Time    Filename/Handle
-------------------- ------ ------------------ ---------------
-----
Datafile Copy        349    2009-07-21:20:27:29
+FRA/orcl/datafile/users.274.692828845

RMAN>
```

Note that the first backup taken is now obsolete.

14) For an alternate view of the backups, in EM, navigate to Availability > Manage Current Backups > Image Copies.

15) Ensure that the username and password of `oracle`/`oracle` are in the host credentials section at the bottom of the page, and then at the upper-right corner of the page, click Delete All Obsolete.

16) When the Specify Job Parameters page appears, click Show RMAN Script.

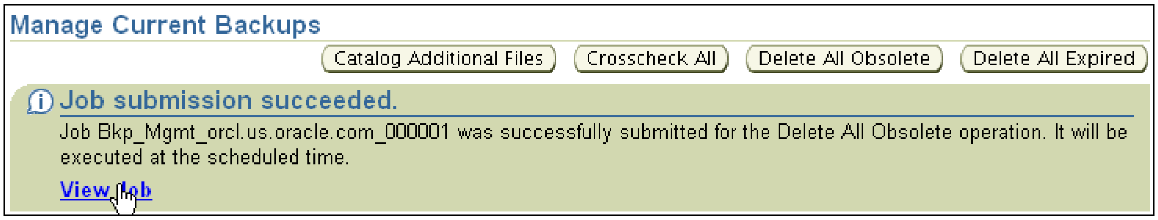


Note that the script issues the `DELETE NOPROMPT OBSOLETE` command. Click OK.

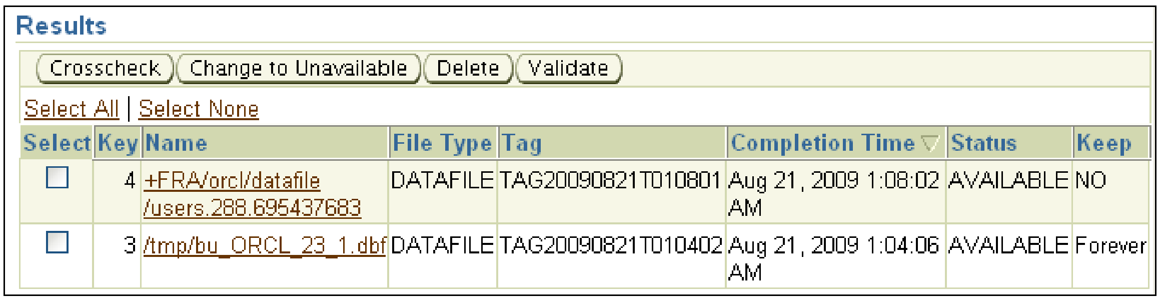
17) Click Submit Job.

18) After the job is submitted, click View Job.

Practice 5-3: Listing Backup Files and Creating Archival Backup (continued)

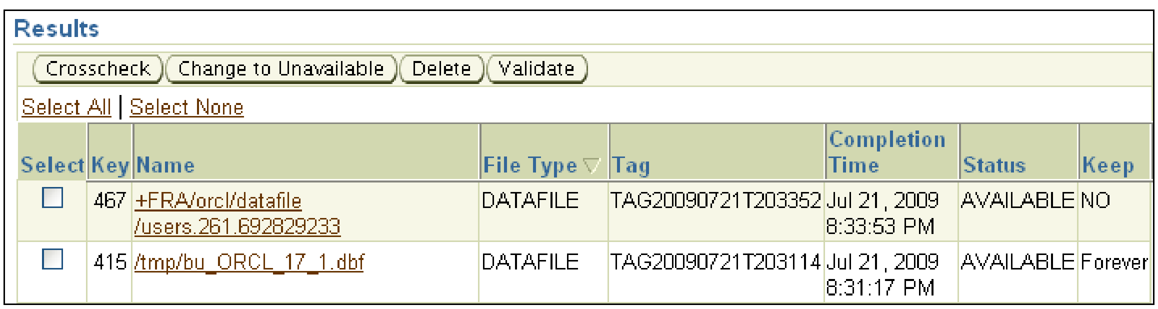


19) When the job status is Running, use your browser's Reload button until the job status appears as Succeeded. Then navigate back to the backup listing page for image copies (Database > Availability > Manage Current Backups > Image Copies).



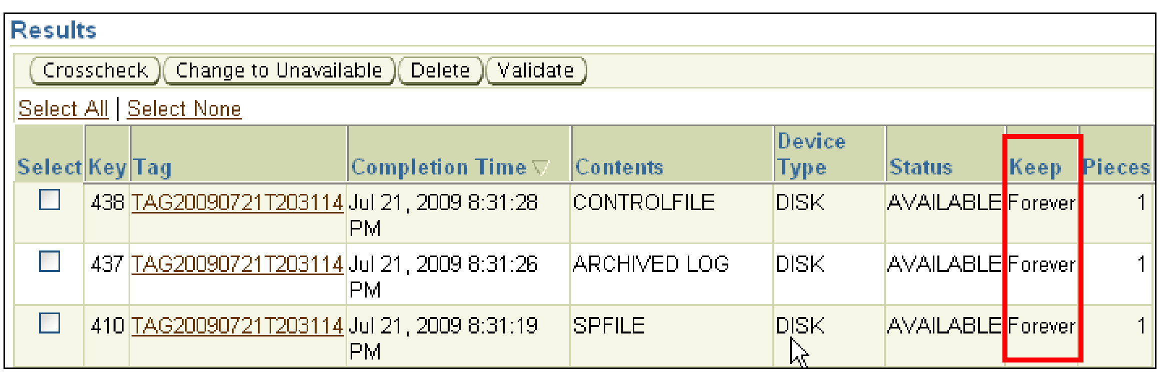
Select	Key	Name	File Type	Tag	Completion Time	Status	Keep
☐	4	+FRA/orcl/datafile /users.288.695437683	DATAFILE	TAG20090821T010801	Aug 21, 2009 1:08:02 AM	AVAILABLE	NO
☐	3	/tmp/bu_ORCL_23_1.dbf	DATAFILE	TAG20090821T010402	Aug 21, 2009 1:04:06 AM	AVAILABLE	Forever

20) Note that there are now only two backups of data file 4. The obsolete one has been deleted because it was the third of three, and the retention policy is set to 1, meaning there need only be two backups of any given file. (Remember the archival backup is not counted.)



Select	Key	Name	File Type	Tag	Completion Time	Status	Keep
☐	467	+FRA/orcl/datafile /users.261.692829233	DATAFILE	TAG20090721T203352	Jul 21, 2009 8:33:53 PM	AVAILABLE	NO
☐	415	/tmp/bu_ORCL_17_1.dbf	DATAFILE	TAG20090721T203114	Jul 21, 2009 8:31:17 PM	AVAILABLE	Forever

21) Click the Backup Sets tab on this page. What archival backup sets appear there? Why are they there?



Select	Key	Tag	Completion Time	Contents	Device Type	Status	Keep	Pieces
☐	438	TAG20090721T203114	Jul 21, 2009 8:31:28 PM	CONTROLFILE	DISK	AVAILABLE	Forever	1
☐	437	TAG20090721T203114	Jul 21, 2009 8:31:26 PM	ARCHIVED LOG	DISK	AVAILABLE	Forever	1
☐	410	TAG20090721T203114	Jul 21, 2009 8:31:19 PM	SPFILE	DISK	AVAILABLE	Forever	1

Practice 5-3: Listing Backup Files and Creating Archival Backup (continued)

Answer: These archival backup sets are here because they were taken as a byproduct of taking the data file image copy as an archival backup. When the data file archival backup was taken, control file autobackup was enabled, which causes the control file and the SPFILE to be backed up. Also, any archive redo logs that are required to recover the image copy are taken. Because the image copy was an archival backup, every other file that is backed up as a part of that backup task is also an archival backup.

22) Enable the automatic backup of the control file again and exit RMAN.

```
RMAN> configure controlfile autobackup on;
RMAN> exit
```

Practice 6-1: Backing up the Database

In this practice, you create a cold backup of the `rcat` database.

Because ASM does not expose its files to the operating system, you create backups to a regular file system destination, so that you can setup various learning situations in the following practices.

Note: The `RCAT` database is in the ARCHIVE log mode, and is capable of performing an inconsistent backup. In this exercise though, you will perform a consistent cold backup.

1) Create the `/home/oracle/BACKUP` directory to hold backup files.

```
$ cd
$ pwd
/home/oracle
$ mkdir BACKUP
```

2) Make sure that your environment is pointing at the `rcat` database.

```
$. oraenv
ORACLE_SID = [+ASM] ? rcat
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
$
```

3) Shut down the database with the `IMMEDIATE` option, then mount it.

```
$ sqlplus / as sysdba

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> startup mount
ORACLE instance started.

Total System Global Area  318046208 bytes
Fixed Size                  1336244 bytes
Variable Size             251661388 bytes
Database Buffers           58720256 bytes
Redo Buffers                6328320 bytes
```

```
Database mounted.
SQL> exit
```

4) Make a whole database image copy backup into the `$HOME/BACKUP` directory using RMAN commands.

```
$ rman target /

Recovery Manager: Release 11.2.0.1.0 - Production on Tue Jul 21
22:26:33 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates.  All rights
reserved.

connected to target database: RCAT (DBID=464959795, not open)

RMAN> backup database format '/home/oracle/BACKUP/cold_%U';

Starting backup at 2009-07-29:08:11:12
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=1 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001
name=+DATA/rcat/datafile/system.267.692935353
input datafile file number=00002
name=+DATA/rcat/datafile/sysaux.268.692935357
input datafile file number=00003
name=+DATA/rcat/datafile/undotbs1.269.692935357
input datafile file number=00005
name=+DATA/rcat/datafile/rcat.277.692937003
input datafile file number=00004
name=+DATA/rcat/datafile/users.270.692935359
channel ORA_DISK_1: starting piece 1 at 2009-07-29:08:11:13
channel ORA_DISK_1: finished piece 1 at 2009-07-29:08:13:08
piece handle=/home/oracle/BACKUP/cold_0fklb6h1_1_1
tag=TAG20090729T081113 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:01:55
Finished backup at 2009-07-29:08:13:08

Starting Control File and SPFILE Autobackup at 2009-07-29:08:13:09
piece handle=+FRA/rcat/autobackup/2009_07_29/s_693475040.330.693475991
comment=NONE
Finished Control File and SPFILE Autobackup at 2009-07-29:08:13:16
RMAN> exit
```

5) Open the `RCAT` database.

```
$ . oraenv
ORACLE_SID = [rcat] ? rcat
sqlplus / as sysdba

SQL> alter database open;

Database altered.
```

Practice 6-1: Backing up the Database (continued)

6) Use the ALTER DATABASE command to back up the control file as a trace file.

```
SQL> alter database  backup controlfile to trace as
'/home/oracle/BACKUP/trace_control.bck';

Database altered.
SQL> exit
```

7) View the content of the /home/oracle/BACKUP directory.

```
$ ls –l /home/oracle/BACKUP
total 1100568
-rw-r----- 1 oracle dba 1125867520 Jul 29 08:12 cold_0fklb6h1_1_1
-rw-r--r-- 1 oracle dba       6601 Jul 29 08:18 trace_control.bck
$
```

Practice 6-2: Recovering from the Loss of a Data File

Because ASM does not expose its files to the operating system, you create tablespaces with data files in a regular file system destination, so that you can setup various learning situations in the following practices.

In the `ORCL` database, you create the `BR_TBS` tablespace with the `CUSTOMERS` table.

1) Log into the `ORCL` instance with SQL*Plus as the `SYS` user.

```
$ . oraenv
ORACLE_SID = [rcat] ? orcl

$ sqlplus / as sysdba

SQL>
```

2) Create the `BR_TBS` tablespace with the `CUSTOMERS` table.

```
SQL> CREATE TABLESPACE br_tbs
  DATAFILE '/home/oracle/BACKUP/br01.dbf' SIZE 25M;

Tablespace created.

SQL> CREATE TABLE customers tablespace br_tbs
  AS SELECT * FROM sh.customers;

Table created.
SQL>
```

3) Find out the number of your new data file and back up the data files to the `/home/oracle/BACKUP` directory. (You might find it useful to keep the SQL*Plus window open and start the RMAN session in another terminal window, but it is not mandatory.)

```
$ . oraenv
ORACLE_SID = [+ASM] ? orcl

$ rman target / catalog rcatowner@rcat

connected to target database: ORCL (DBID=1220535480)
recovery catalog database Password: oracle_4U <<<not displayed
connected to recovery catalog database

RMAN> report schema;
starting full resync of recovery catalog
full resync complete
Report of database schema for database with db_unique_name ORCL

List of Permanent Datafiles
===========================
File Size(MB) Tablespace           RB segs Datafile Name
```

```
---- -------- -------------------- ------- ----------------------
1    690      SYSTEM               YES
+DATA/orcl/datafile/system.256.692754557
2    610      SYSAUX               NO
+DATA/orcl/datafile/sysaux.257.692754559
3    100      UNDOTBS1             YES
+DATA/orcl/datafile/undotbs1.258.692754561
4    5        USERS                NO
+DATA/orcl/datafile/users.259.692754561
5    100      EXAMPLE              NO
+DATA/orcl/datafile/example.265.692754837
6    25       BR_TBS               NO
/home/oracle/BACKUP/br01.dbf

List of Temporary Files
=======================
File Size(MB) Tablespace           Maxsize(MB) Tempfile Name
---- -------- -------------------- ----------- -------------------
--
1    28       TEMP                 32767
+DATA/orcl/tempfile/temp.264.692754825

RMAN> backup as copy datafile 6 format
'/home/oracle/BACKUP/br_%d_%s_%p.dbf';

Starting backup at 2009-07-29:08:38:47
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=29 device type=DISK
channel ORA_DISK_1: starting datafile copy
input datafile file number=00006
name=/home/oracle/BACKUP/br01.dbf
output file name=/home/oracle/BACKUP/br_ORCL_22_1.dbf
tag=TAG20090729T083848 RECID=5 STAMP=693477530
channel ORA_DISK_1: datafile copy complete, elapsed time:
00:00:03
Finished backup at 2009-07-29:08:38:51

Starting Control File and SPFILE Autobackup at 2009-07-
29:08:38:51
piece
handle=+FRA/orcl/autobackup/2009_07_29/s_693477533.332.693477535
comment=NONE
Finished Control File and SPFILE Autobackup at 2009-07-
29:08:38:54
RMAN>
```

4) To simulate the passage of time, and to make sure that this data is not cached in the buffer cache, perform the following steps and exit from SQL*Plus.

```
SQL> alter system switch logfile;

System altered;
```

Practice 6-2: Recovering from the Loss of a Data File (continued)

```
SQL> alter system checkpoint;

System altered.

SQL> alter system switch logfile;

System altered
SQL> exit
```

5) Delete the data file belonging to the BR_TBS tablespace (**not** the backup).

```
$ cd ~/BACKUP
$ ls –l br*
-rw-r----- 1 oracle dba 26222592 Jul 22 20:57 br01.dbf
-rw-r----- 1 oracle dba 26222592 Jul 22 20:52 br_ORCL_22_1.dbf
$ rm br01.dbf
$ ls –l br*
-rw-r----- 1 oracle dba 26222592 Jul 22 20:52 br_ORCL_22_1.dbf
$
```

6) Now log back in to the database, flush the buffer cache and try to access the data in the CUSTOMERS table.

```
$ sqlplus / as sysdba

SQL> alter system flush buffer_cache;
System altered.

SQL> select count(*) from sys.customers;
select count(*) from sys.customers

ERROR at line 1:
ORA-01116: error in opening database file 6
ORA-01110: data file 6: '/home/oracle/BACKUP/br01.dbf'
ORA-27041: unable to open file
Linux Error: 2: No such file or directory
Additional information: 3
```

7) Knowing you are going to have to restore this file, you now take it offline, and exit from SQL*Plus.

```
SQL> alter database datafile 6 offline;

Database altered.

SQL> exit
```

Practice 6-2: Recovering from the Loss of a Data File (continued)

8) Now navigate to your `$HOME/BACKUP` directory, to find what backups you have of the data file. Then choose the most recent, and copy *YOUR* file into where the live one should be.

```
$ cd /home/oracle/BACKUP/

$ ls -al
total 1126216
drwxr-xr-x  2 oracle oinstall       4096 Jul 29 08:42 .
drwxrwxrwx 24 oracle oinstall       4096 Jul 29 08:02 ..
-rw-r-----  1 oracle dba        26222592 Jul 29 08:38 br_ORCL_22_1.dbf
-rw-r-----  1 oracle dba      1125867520 Jul 29 08:12 cold_0fklb6h1_1_1
-rw-r--r--  1 oracle dba            6601 Jul 29 08:18 trace_control.bck

$ cp br_ORCL_22_1.dbf br01.dbf
$ ls –l
-rw-r-----  1 oracle oinstall   26222592 Jul 29 08:38 br01.dbf
-rw-r-----  1 oracle dba        26222592 Jul 29 08:38 br_ORCL_22_1.dbf
-rw-r-----  1 oracle dba      1125867520 Jul 29 08:12 cold_0fklb6h1_1_1
-rw-r--r--  1 oracle dba            6601 Jul 29 08:18 trace_control.bck
```

9) Now log in to SQL*Plus and try to bring the file online.

```
$ sqlplus / as sysdba

SQL> alter database datafile 6 online;
alter database datafile 6 online
*
ERROR at line 1:
ORA-01113: file 6 needs media recovery
ORA-01110: data file 6: '/home/oracle/BACKUP/br01.dbf'
```

10) The data file cannot be brought online as it is too old and, therefore, you need to perform media recovery to roll it forward.

```
SQL> recover datafile 6;
Media recovery complete.
SQL>
```

11) Now try to bring the data file online.

```
SQL> alter database datafile 6 online;

Database altered.
```

12) Try again to access the `sys.customers` table, and then exit SQL*Plus.

```
SQL> select count(*) from sys.customers;

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

```
     55500

SQL> exit
```

13) For the `ORCL` database, confirm that the control file is automatically backed up and that the backup rentention is set to 1 (if not, modify it), take a database backup and delete obsolete files (in preparation for the following practices).

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl

$ rman target / catalog rcatowner@rcat

connected to target database: ORCL (DBID=1220535480)
recovery catalog database Password: oracle_4U <<<not displayed
connected to recovery catalog database

RMAN> show retention policy;

RMAN configuration parameters for database with db_unique_name
ORCL are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
RMAN>
RMAN> backup database;
Starting backup
.
.
.

RMAN>

RMAN> delete noprompt obsolete;
RMAN retention policy will be applied to the command
.
.
.

RMAN> exit
$
```

Practice 6-3: Recovering from the Loss of all Control Files

This practice makes use of the `ORCL` database and the recovery catalog. You will simulate the loss of all control files, and then restore the control files and recover the database.

1) Connect to the ORCL database and make a binary backup of the control file.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
$ sqlplus / as sysdba

SQL> alter database backup controlfile to
'/home/oracle/BACKUP/ctrl.bkp';
Database altered.
```

2) Identify the current control files for the `ORCL` database. You need to know them in one of the following tasks. Then shutdown the `ORCL` database to allow the control files to be deleted and exit SQL*Plus.

```
SQL> select name from v$controlfile;

NAME
----------------------------------------------
+DATA/orcl/controlfile/current.260.692879691
+FRA/orcl/controlfile/current.256.692879691

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> exit
```

3) Set the environment to the ASM instance and run the `asmcmd` utility to delete the two control files identified above.

Note: The names of your control files may be different. Delete ***YOUR*** control files.

```
$ . oraenv
ORACLE_SID = [orcl] ? +ASM
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is
/u01/app/oracle

$ asmcmd rm +DATA/orcl/controlfile/current.260.692879691
$ asmcmd rm +FRA/orcl/controlfile/current.256.692879691
```

4) Set the environment back to the ORCL database and attempt to start up the database.

```
$ . oraenv
ORACLE_SID = [+ASM] ? orcl
```

Practice 6-3: Recovering from the Loss of all Control Files (continued)

```
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle

$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Wed Jul 29 10:03:55
2009
Copyright (c) 1982, 2009, Oracle.  All rights reserved.

Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area  418484224 bytes
Fixed Size                  1336908 bytes
Variable Size             268437940 bytes
Database Buffers          142606336 bytes
Redo Buffers                6103040 bytes
ORA-00205: error in identifying control file, check alert log for
more info

SQL> exit
```

5) Using RMAN, connect to the ORCL target database, to the RCAT catalog database, and restore the control file from the autobackup.

```
$ rman target / catalog rcatowner@rcat

Recovery Manager: Release 11.2.0.1.0 - Production on Wed Jul 29
10:06:22 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates.  All
rights reserved.

connected to target database: ORCL (not mounted)
recovery catalog database Password: oracle_4U <<<not displayed
connected to recovery catalog database

RMAN> restore controlfile from autobackup;

Starting restore at 2009-07-29:10:06:48
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=27 device type=DISK

recovery area destination: +FRA
database name (or database unique name) used for search: ORCL
```

```
channel ORA_DISK_1: AUTOBACKUP
+fra/ORCL/AUTOBACKUP/2009_07_29/s_693478885.335.693478885 found
in the recovery area
channel ORA_DISK_1: looking for AUTOBACKUP on day: 20090729
channel ORA_DISK_1: restoring control file from AUTOBACKUP
+fra/ORCL/AUTOBACKUP/2009_07_29/s_693478885.335.693478885
channel ORA_DISK_1: control file restore from AUTOBACKUP complete
output file name=+DATA/orcl/controlfile/current.260.693482811
output file name=+FRA/orcl/controlfile/current.256.693482813
Finished restore at 2009-07-29:10:06:54/
```

6) Mount the restored control file and attempt to open the database.

```
RMAN> alter database mount;
database mounted
released channel: ORA_DISK_1
RMAN> alter database open resetlogs;
RMAN-00571:
===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS
===============
RMAN-00571:
===========================================================
RMAN-03002: failure of alter db command at 07/29/2009 10:08:45
ORA-01152: file 1 was not restored from a sufficiently old backup
ORA-01110: data file 1:
'+DATA/orcl/datafile/system.256.692879503'
```

7) Because the control files have been restored, the database must be recovered.

```
RMAN> recover database;
Starting recover at 2009-07-29:10:08:58
Starting implicit crosscheck backup at 2009-07-29:10:08:58
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=27 device type=DISK
Crosschecked 12 objects
Finished implicit crosscheck backup at 2009-07-29:10:09:00

Starting implicit crosscheck copy at 2009-07-29:10:09:00
using channel ORA_DISK_1
Crosschecked 4 objects
Finished implicit crosscheck copy at 2009-07-29:10:09:01
searching for all files in the recovery area
cataloging files...
cataloging done
List of Cataloged Files
=======================
File Name:
+fra/ORCL/AUTOBACKUP/2009_07_29/s_693478885.335.693478885
using channel ORA_DISK_1
```

```
starting media recovery

archived log for thread 1 with sequence 59 is already on disk as
file +DATA/orcl/onlinelog/group_2.262.692879707
archived log file name=+DATA/orcl/onlinelog/group_2.262.692879707
thread=1 sequence=59
media recovery complete, elapsed time: 00:00:01
Finished recover at 2009-07-29:10:09:05
```

8) Attempt to open the database with the `RESETLOGS` option after recovery.

```
RMAN> alter database open resetlogs;

database opened
new incarnation of database registered in recovery catalog
starting full resync of recovery catalog
full resync complete
```

9) Perform a full database backup and delete obsolete files to be prepared for future labs.

```

RMAN> backup database;

Starting backup at 2009-07-29:10:49:21
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=46 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00002
name=+DATA/orcl/datafile/sysaux.257.692879505
channel ORA_DISK_1: starting piece 1 at 2009-07-29:10:49:22
channel ORA_DISK_1: finished piece 1 at 2009-07-29:10:50:18
piece
handle=+FRA/orcl/backupset/2009_07_29/nnndf0_tag20090729t104922_0
.340.693485363 tag=TAG20090729T104922 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:56
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001
name=+DATA/orcl/datafile/system.256.692879503
channel ORA_DISK_1: starting piece 1 at 2009-07-29:10:50:22
channel ORA_DISK_1: finished piece 1 at 2009-07-29:10:52:04
piece
handle=+FRA/orcl/backupset/2009_07_29/nnndf0_tag20090729t104922_0
.341.693485425 tag=TAG20090729T104922 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:01:42
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00006
name=/home/oracle/BACKUP/br01.dbf
```

```
input datafile file number=00003
name=+DATA/orcl/datafile/undotbs1.258.692879507
input datafile file number=00004
name=+DATA/orcl/datafile/users.259.692879509
input datafile file number=00005
name=+DATA/orcl/datafile/example.265.692879765
channel ORA_DISK_1: starting piece 1 at 2009-07-29:10:52:11
channel ORA_DISK_1: finished piece 1 at 2009-07-29:10:52:57
piece
handle=+FRA/orcl/backupset/2009_07_29/nnndf0_tag20090729t104922_0
.342.693485537 tag=TAG20090729T104922 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:46
Finished backup at 2009-07-29:10:52:57

Starting Control File and SPFILE Autobackup at 2009-07-
29:10:52:57
piece
handle=+FRA/orcl/autobackup/2009_07_29/s_693485589.343.693485603
comment=NONE
Finished Control File and SPFILE autobackup at 2009-07-29:10:53

RMAN> delete noprompt obsolete;
. . .

RMAN> exit
```

Practice 6-4: Recovering from the Loss of a Redo Log Group

This practice makes use of the ORCL database. You will delete the current redo log files, and see the effect this has on the database.

1) Log in to the ORCL database using SQL*Plus, and run the redo_view.sql query to gather information about your redo logs.

```
$ sqlplus / as sysdba

SQL> @redo_view.sql
SQL> set linesize 120
SQL> col member format a43
SQL> col status format a10
SQL> select l.group#, l.sequence#, l.archived,
  2  l.status, f.member
  3  from v$log l, v$logfile f
  4  where l.group#=f.group#;


    GROUP#  SEQUENCE# ARC STATUS     MEMBER
---------- ---------- --- ---------- ----------------------------
---------------
         3          0 YES UNUSED
+DATA/orcl/onlinelog/group_3.263.692879721
         3          0 YES UNUSED
+FRA/orcl/onlinelog/group_3.259.692879727
         2          0 YES UNUSED
+DATA/orcl/onlinelog/group_2.262.692879707
         2          0 YES UNUSED
+FRA/orcl/onlinelog/group_2.258.692879715
         1          1 NO  CURRENT
+DATA/orcl/onlinelog/group_1.261.692879693
         1          1 NO  CURRENT
+FRA/orcl/onlinelog/group_1.257.692879701

6 rows selected.
SQL>
```

2) From this it can be seen in our example that log group 1 is the current log group. Your current redo log group may be different from the one shown. Note the exact group names of your current group. You need the names during the next step.

Determine the current redo log group for your database, and then delete its member files with the asmcmd utility. You must shutdown the database in order to delete files that are in use in an ASM disk group.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
```

Practice 6-4: Recovering from the Loss of a Redo Log Group (continued)

```
SQL> exit
```

```
$ . oraenv
ORACLE_SID = [orcl] ? +ASM
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is
/u01/app/oracle
$ asmcmd rm +DATA/orcl/onlinelog/group_1.261.692879693
$ asmcmd rm +FRA/orcl/onlinelog/group_1.257.692879701
```

3) Attempt to start the `ORCL` database and observe what happens. Then exit your SQL*Plus session.

```
$ . oraenv
ORACLE_SID = [+ASM] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Tue Aug 4 03:07:48
2009
Copyright (c) 1982, 2009, Oracle.  All rights reserved.
Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area  481259520 bytes
Fixed Size                  1337352 bytes
Variable Size             411043832 bytes
Database Buffers           62914560 bytes
Redo Buffers                5963776 bytes
Database mounted.
ORA-03113: end-of-file on communication channel
Process ID: 29445
Session ID: 1 Serial number: 5
SQL> exit
```

4) Display the last section of the alert log to find out more details related to the failure in starting the `ORCL` instance.

```
$ tail /u01/app/oracle/diag/rdbms/orcl/orcl/trace/alert_orcl.log
ORA-15012: ASM file '+FRA/orcl/onlinelog/group_1.257.692879701'
does not exist
ORA-00312: online log 1 thread 1:
'+DATA/orcl/onlinelog/group_1.261.692879693'
ORA-17503: ksfdopn:2 Failed to open file
+DATA/orcl/onlinelog/group_1.261.692879693
```

Practice 6-4: Recovering from the Loss of a Redo Log Group (continued)

```
ORA-15012: ASM file '+DATA/orcl/onlinelog/group_1.261.692879693'
does not exist
Errors in file
/u01/app/oracle/diag/rdbms/orcl/orcl/trace/orcl_ora_15563.trc:
ORA-00313: open failed for members of log group 1 of thread
ORA-00312: online log 1 thread 1:
'+DATA/orcl/onlinelog/group_1.261.692879693'
ORA-00312: online log 1 thread 1:
'+FRA/orcl/onlinelog/group_1.257.692879701'
USER (ospid: 15563): terminating the instance due to error 313
Instance terminated by USER, pid = 15563
```

As you can observe, the instance terminates due to missing all the members of your log group.

5) Log into SQL*Plus as `SYSDBA`, start up and mount the `ORCL` instance. Query the `V$LOG` view to determine which log group is the current group and whether its files have been archived already.

```
$ sqlplus / as sysdba

SQL> startup mount
ORACLE instance started.

Total System Global Area  481259520 bytes
Fixed Size                  1337352 bytes
Variable Size             411043832 bytes
Database Buffers           62914560 bytes
Redo Buffers                5963776 bytes
Database mounted.

SQL> select group#,status,archived from v$log;

    GROUP# STATUS           ARC
---------- ---------------- ---
         1 CURRENT          NO
         3 UNUSED           YES
         2 UNUSED           YES
```

6) If you clear ***YOUR*** missing log file group, then this will re-create the missing log files.

```
SQL> alter database clear logfile group 1;
alter database clear logfile group 1
*
ERROR at line 1:
ORA-00350: log 1 of instance orcl (thread 1) needs to be archived
ORA-00312: online log 1 thread 1:
'+DATA/orcl/onlinelog/group_1.261.693969247'
```

Practice 6-4: Recovering from the Loss of a Redo Log Group (continued)

```
ORA-00312: online log 1 thread 1:
'+FRA/orcl/onlinelog/group_1.257.693969255'
```

This is because the logfile has been deleted, and therefore cannot be archived. Because the log file has not been archived, lgwr is not allowed to overwrite it, even if the file no longer exists.

7) Because the logfile group has not been archived, you must use the keyword “unarchived” in the command. Open the database and exit SQL*Plus.

```
SQL> alter database clear unarchived logfile group 1;
Database altered.

SQL> alter database open;
Database altered.

SQL> exit
```

8) Perform a backup of the `ORCL` database including archive logs to be ready for future labs.

```
$ rman target / catalog rcatowner@rcat

Recovery Manager: Release 11.2.0.1.0 - Production on Wed Jul 29
10:49:07 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates.  All
rights reserved.

connected to target database: ORCL (DBID=1220660426)
recovery catalog database Password: oracle_4U <<<not displayed
connected to recovery catalog database

RMAN> backup database plus archivelog;
.. .
Finished backup at 20-AUG-09

Starting Control File and SPFILE Autobackup at 20-AUG-09
piece
handle=+FRA/orcl/autobackup/2009_08_20/s_695409765.279.695409767
comment=NONE
Finished Control File and SPFILE Autobackup at 20-AUG-09

RMAN> delete noprompt obsolete;
. . .

Deleted 2 objects

RMAN> exit
```

Note: Your number of deleted objects will most likely be different.

Practice 7-1: Recovering Image Copies

In this practice, you recover an image copy of a file to the current SCN, to allow faster recovery time later. You create a new tablespace called `APPRAISAL`, which has one table in it. After creating it initially with a small amount of data, you take an incremental backup of it. Then you add many rows and take another incremental backup. At that point, you have an image copy of the `APPRAISAL` tablespace and also an incremental backup of it. Because you need to recover the tablespace later, you recover the image copy so that it is as up-to-date as the last incremental backup. This is done without going to the expense of creating a new image copy.

1) Make sure you are at the `~/labs` directory and run the `create_appraisal_ts.sh` script to create a new tablespace called `APPRAISAL`.

```
$ cd ~/labs
$ ./create_appraisal_ts.sh
Tablespace created.
Table created.
$
```

2) Make a level 1 backup of the tablespace, to be used for image copy recovery. If no level 1 exists yet, it will actually create a level 0 incremental backup. This takes several minutes.

```
$ rman target / catalog rcatowner@rcat

Recovery Manager: Release 11.2.0.1.0 - Production on Wed Jul 29
10:49:07 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates.  All rights
reserved.

connected to target database: ORCL (DBID=1220660426)
recovery catalog database Password: oracle_4U <<<not displayed
connected to recovery catalog database

RMAN> backup incremental level 1 for recover of copy with tag
'app_incr' database;

Starting backup at 2009-08-05:03:00:24
starting full resync of recovery catalog
full resync complete
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=58 device type=DISK
no parent backup or copy of datafile 1 found
no parent backup or copy of datafile 2 found
no parent backup or copy of datafile 5 found
no parent backup or copy of datafile 3 found
```

Practice 7-1: Recovering Image Copies (continued)

```
no parent backup or copy of datafile 6 found
no parent backup or copy of datafile 7 found
no parent backup or copy of datafile 4 found
channel ORA_DISK_1: starting datafile copy
input datafile file number=00001
name=+DATA/orcl/datafile/system.260.694050517
output file name=+FRA/orcl/datafile/system.301.694062029
tag=APP_INCR RECID=6 STAMP=694062094
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:01:16
channel ORA_DISK_1: starting datafile copy
input datafile file number=00002
name=+DATA/orcl/datafile/sysaux.266.694050521
output file name=+FRA/orcl/datafile/sysaux.300.694062109
tag=APP_INCR RECID=7 STAMP=694062216
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:01:56
channel ORA_DISK_1: starting datafile copy
input datafile file number=00005
name=+DATA/orcl/datafile/example.259.694050813
output file name=+FRA/orcl/datafile/example.298.694062227
tag=APP_INCR RECID=8 STAMP=694062237
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:15
channel ORA_DISK_1: starting datafile copy
input datafile file number=00003
name=+DATA/orcl/datafile/undotbs1.278.694050521
output file name=+FRA/orcl/datafile/undotbs1.297.694062243
tag=APP_INCR RECID=9 STAMP=694062251
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:15
channel ORA_DISK_1: starting datafile copy
input datafile file number=00006 name=/home/oracle/BACKUP/br01.dbf
output file name=+FRA/orcl/datafile/br_tbs.296.694062259
tag=APP_INCR RECID=10 STAMP=694062265
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:21
channel ORA_DISK_1: starting datafile copy
input datafile file number=00007
name=+DATA/orcl/datafile/appraisal.256.694061937
output file name=+FRA/orcl/datafile/appraisal.295.694062283
tag=APP_INCR RECID=11 STAMP=694062285
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:08
channel ORA_DISK_1: starting datafile copy
input datafile file number=00004
name=+DATA/orcl/datafile/users.264.694050521
output file name=+FRA/orcl/datafile/users.294.694062289 tag=APP_INCR
RECID=12 STAMP=694062289
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:00:03
Finished backup at 2009-08-05:03:04:51

Starting Control File and SPFILE Autobackup at 2009-08-05:03:04:51
piece
handle=+FRA/orcl/autobackup/2009_08_05/s_694062294.292.694062307
comment=NONE
Finished Control File and SPFILE Autobackup at 2009-08-05:03:05:09

RMAN>
```

3) In a separate terminal window, perform some DML on the table in the `APPRAISAL` tablespace. Use the `emp_inserts.sh` script.

Practice 7-1: Recovering Image Copies (continued)

```
$ cd ~/labs
$ ./emp_inserts.sh

107 rows created.

Commit complete.

214 rows created.

Commit complete.

$
```

4) List the copy of the APPRAISAL tablespace to see its SCN.

```
RMAN> list copy of tablespace appraisal;

List of Datafile Copies
=======================

Key     File S Completion Time     Ckp SCN    Ckp Time
------- ---- - ------------------- ---------- ----------------
---
1975    7    A 2009-08-05:03:04:45 853719     2009-08-
05:03:04:39
        Name: +FRA/orcl/datafile/appraisal.295.694062283
        Tag: APP_INCR
RMAN>
```

5) Run the script to perform more transactions on the table that resides in the APPRAISAL tablespace.

```
$ ./emp_inserts.sh

428 rows created.

Commit complete.

856 rows created.

Commit complete.

$
```

6) Perform another level 1 backup. This one will indeed be a level 1 because you already have a level 0.

```
RMAN> backup incremental level 1 for recover of copy with tag
'app_incr' database;


Starting backup at 2009-08-05:03:25:48
using channel ORA_DISK_1
```

Practice 7-1: Recovering Image Copies (continued)

```
channel ORA_DISK_1: starting incremental level 1 datafile
backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00001
name=+DATA/orcl/datafile/system.260.694050517
channel ORA_DISK_1: starting piece 1 at 2009-08-05:03:25:49
channel ORA_DISK_1: finished piece 1 at 2009-08-05:03:25:50
piece
handle=+FRA/orcl/backupset/2009_08_05/nnndn1_app_incr_0.285.69
4063549 tag=APP_INCR comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time:
00:00:01
channel ORA_DISK_1: starting incremental level 1 datafile
backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00002
name=+DATA/orcl/datafile/sysaux.266.694050521
channel ORA_DISK_1: starting piece 1 at 2009-08-05:03:25:51
channel ORA_DISK_1: finished piece 1 at 2009-08-05:03:25:54
piece
handle=+FRA/orcl/backupset/2009_08_05/nnndn1_app_incr_0.284.69
4063551 tag=APP_INCR comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time:
00:00:03
channel ORA_DISK_1: starting incremental level 1 datafile
backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00006
name=/home/oracle/BACKUP/br01.dbf
input datafile file number=00003
name=+DATA/orcl/datafile/undotbs1.278.694050521
input datafile file number=00004
name=+DATA/orcl/datafile/users.264.694050521
input datafile file number=00005
name=+DATA/orcl/datafile/example.259.694050813
input datafile file number=00007
name=+DATA/orcl/datafile/appraisal.256.694061937
channel ORA_DISK_1: starting piece 1 at 2009-08-05:03:25:54
channel ORA_DISK_1: finished piece 1 at 2009-08-05:03:25:55
piece
handle=+FRA/orcl/backupset/2009_08_05/nnndn1_app_incr_0.277.69
4063555 tag=APP_INCR comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time:
00:00:01
Finished backup at 2009-08-05:03:25:55

Starting Control File and SPFILE Autobackup at 2009-08-
05:03:25:55
piece
handle=+FRA/orcl/autobackup/2009_08_05/s_694063556.261.6940635
57 comment=NONE
```

Practice 7-1: Recovering Image Copies (continued)

```
Finished Control File and SPFILE Autobackup at 2009-08-
05:03:25:59
RMAN>
```

7) List **and note** the SCN of the APPRAISAL tablespace incremental backup (not the image copy).

```
RMAN> list backup of tablespace appraisal;

List of Backup Sets
===================


BS Key  Type LV Size       Device Type Elapsed Time Completion
Time
------- ---- -- ---------- ----------- ------------ ----------
---------
2036    Incr 1  2.24M      DISK        00:00:00     2009-08-
05:03:25:54
        BP Key: 2041   Status: AVAILABLE  Compressed: NO  Tag:
APP_INCR
        Piece Name:
+FRA/orcl/backupset/2009_08_05/nnndn1_app_incr_0.277.694063555
  List of Datafiles in backup set 2036
  File LV Type Ckp SCN    Ckp Time            Name
  ---- -- ---- ---------- ------------------- ----
  7    1  Incr 855661     2009-08-05:03:25:54
+DATA/orcl/datafile/appraisal.256.694061937
RMAN>
```

8) Use the incremental backup to recover the APPRAISAL tablespace image copy.

```
RMAN> recover copy of tablespace appraisal with tag
'app_incr';

Starting recover at 2009-08-05:03:29:20
using channel ORA_DISK_1
channel ORA_DISK_1: starting incremental datafile backup set
restore
channel ORA_DISK_1: specifying datafile copies to recover
recovering datafile copy file number=00007
name=+FRA/orcl/datafile/appraisal.288.694063519
channel ORA_DISK_1: reading from backup piece
+FRA/orcl/backupset/2009_08_05/nnndn1_app_incr_0.277.694063555
channel ORA_DISK_1: piece
handle=+FRA/orcl/backupset/2009_08_05/nnndn1_app_incr_0.277.69
4063555 tag=APP_INCR
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished recover at 2009-08-05:03:29:21

Starting Control File and SPFILE Autobackup at 2009-08-
05:03:29:21
```

```
piece
handle=+FRA/orcl/autobackup/2009_08_05/s_694063763.260.6940637
63 comment=NONE
Finished Control File and SPFILE Autobackup at 2009-08-
05:03:29:26
RMAN>
```

9) List the SCN for the APPRAISAL tablespace image copy. What is it now?

```
RMAN> list copy of tablespace appraisal;

List of Datafile Copies
=======================

Key     File S Completion Time     Ckp SCN    Ckp Time
------- ---- - ------------------- ---------- ----------------
---
2090    7    A 2009-08-05:03:29:20 855661     2009-08-
05:03:25:54
        Name: +FRA/orcl/datafile/appraisal.288.694063519
        Tag: APP_INCR
RMAN>
```

Note that it is now equal to the SCN of the last incremental backup.

10) Recover all the data file image copies in the database based on the most recent incremental backup.

```
RMAN> recover copy of database with tag 'app_incr';

Starting recover at 2009-08-05:03:33:29
using channel ORA_DISK_1
no copy of datafile 7 found to recover
channel ORA_DISK_1: starting incremental datafile backup set
restore
channel ORA_DISK_1: specifying datafile copies to recover
recovering datafile copy file number=00001
name=+FRA/orcl/datafile/system.291.694063333
channel ORA_DISK_1: reading from backup piece
+FRA/orcl/backupset/2009_08_05/nnndn1_app_incr_0.285.694063549
channel ORA_DISK_1: piece
handle=+FRA/orcl/backupset/2009_08_05/nnndn1_app_incr_0.285.69
4063549 tag=APP_INCR
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting incremental datafile backup set
restore
channel ORA_DISK_1: specifying datafile copies to recover
recovering datafile copy file number=00002
name=+FRA/orcl/datafile/sysaux.316.694063393
channel ORA_DISK_1: reading from backup piece
+FRA/orcl/backupset/2009_08_05/nnndn1_app_incr_0.284.694063551
```

Practice 7-1: Recovering Image Copies (continued)

```
channel ORA_DISK_1: piece
handle=+FRA/orcl/backupset/2009_08_05/nnndn1_app_incr_0.284.69
4063551 tag=APP_INCR
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
channel ORA_DISK_1: starting incremental datafile backup set
restore
channel ORA_DISK_1: specifying datafile copies to recover
recovering datafile copy file number=00003
name=+FRA/orcl/datafile/undotbs1.290.694063491
recovering datafile copy file number=00004
name=+FRA/orcl/datafile/users.283.694063527
recovering datafile copy file number=00005
name=+FRA/orcl/datafile/example.307.694063477
recovering datafile copy file number=00006
name=+FRA/orcl/datafile/br_tbs.289.694063517
channel ORA_DISK_1: reading from backup piece
+FRA/orcl/backupset/2009_08_05/nnndn1_app_incr_0.277.694063555
channel ORA_DISK_1: piece
handle=+FRA/orcl/backupset/2009_08_05/nnndn1_app_incr_0.277.69
4063555 tag=APP_INCR
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished recover at 2009-08-05:03:33:33

Starting Control File and SPFILE Autobackup at 2009-08-
05:03:33:33
piece
handle=+FRA/orcl/autobackup/2009_08_05/s_694064014.262.6940640
15 comment=NONE
Finished Control File and SPFILE Autobackup at 2009-08-
05:03:33:37
RMAN>
```

11) View the SCNs of all the image copies now. They should all be, at the most, equal to the latest incremental backup.

```
RMAN> list copy;

specification does not match any control file copy in the
repository
List of Datafile Copies
=======================

Key     File S Completion Time     Ckp SCN    Ckp Time
------- ---- - ------------------- ---------- ----------------
---
2144    1    A 2009-08-05:03:33:30 855654     2009-08-
05:03:25:49
        Name: +FRA/orcl/datafile/system.291.694063333
        Tag: APP_INCR
```

Practice 7-1: Recovering Image Copies (continued)

```
2145    2    A 2009-08-05:03:33:31 855657     2009-08-
05:03:25:51
        Name: +FRA/orcl/datafile/sysaux.316.694063393
        Tag: APP_INCR

2149    3    A 2009-08-05:03:33:32 855661     2009-08-
05:03:25:54
        Name: +FRA/orcl/datafile/undotbs1.290.694063491
        Tag: APP_INCR

2148    4    A 2009-08-05:03:33:32 855661     2009-08-
05:03:25:54
        Name: +FRA/orcl/datafile/users.283.694063527
        Tag: APP_INCR

1491    4    A 2009-08-05:02:19:40 847825     2009-08-
05:02:19:40
        Name: +FRA/orcl/datafile/users.299.694059581
        Tag: TAG20090805T021940

1436    4    A 2009-08-05:02:18:41 847727     2009-08-
05:02:18:41
        Keep: BACKUP_LOGS        Until: FOREVER
        Name: /tmp/bu_ORCL_17_1.dbf
        Tag: TAG20090805T021838

2146    5    A 2009-08-05:03:33:32 855661     2009-08-
05:03:25:54
        Name: +FRA/orcl/datafile/example.307.694063477
        Tag: APP_INCR

2147    6    A 2009-08-05:03:33:32 855661     2009-08-
05:03:25:54
        Name: +FRA/orcl/datafile/br_tbs.289.694063517
        Tag: APP_INCR

1641    6    A 2009-08-05:02:24:21 848546     2009-08-
05:02:24:20
        Name: /home/oracle/BACKUP/br_ORCL_23_1.dbf
        Tag: TAG20090805T022420

2090    7    A 2009-08-05:03:29:20 855661     2009-08-
05:03:25:54
        Name: +FRA/orcl/datafile/appraisal.288.694063519
        Tag: APP_INCR

RMAN>
```

Practice 7-2: Performing Fast Recovery

In this practice, you take advantage of the Flash Recovery Area to perform a fast recovery of a data file.
Note: The data file numbers in your database may differ from what is shown here.

1) Use the RMAN session from the previous practice and take the `APPRAISAL` data file offline.

```
RMAN> sql "alter tablespace appraisal offline";

sql statement: alter tablespace appraisal offline
starting full resync of recovery catalog
full resync complete

RMAN>
```

2) Use the `SWITCH` command to replace the data file from the flash recovery area.

   a) Determine the name of the data file that is currently associated with the `APPRAISAL` tablespace.

```
RMAN> report schema;

Report of database schema for database with db_unique_name
ORCL

List of Permanent Datafiles
===========================
File Size(MB) Tablespace           RB segs Datafile Name
---- -------- -------------------- ------- -------------------
-----
1    680      SYSTEM               YES
+DATA/orcl/datafile/system.260.694050517
2    570      SYSAUX               NO
+DATA/orcl/datafile/sysaux.266.694050521
3    85       UNDOTBS1             YES
+DATA/orcl/datafile/undotbs1.278.694050521
4    5        USERS                NO
+DATA/orcl/datafile/users.264.694050521
5    100      EXAMPLE              NO
+DATA/orcl/datafile/example.259.694050813
6    25       BR_TBS               NO
/home/oracle/BACKUP/br01.dbf
7    25       APPRAISAL            NO
+DATA/orcl/datafile/appraisal.256.694061937

List of Temporary Files
=======================
File Size(MB) Tablespace           Maxsize(MB) Tempfile Name
---- -------- -------------------- ----------- ------------
1    28       TEMP                 32767
+DATA/orcl/tempfile/temp.265.694050795
```

Practice 7-2: Performing Fast Recovery (continued)

Note that there is only one data file for the tablespace. In this case, it is data file number 7. You can use that data file number, instead of the tablespace name, in the upcoming set of commands.

b) Confirm that you have an image copy for data file 7 that you can switch to.

```
RMAN> list copy of datafile 7;

List of Datafile Copies
=======================

Key     File S Completion Time     Ckp SCN    Ckp Time
------- ---- - ------------------- ---------- ----------------
---
2090    7    A 2009-08-05:03:29:20 855661     2009-08-
05:03:25:54
        Name: +FRA/orcl/datafile/appraisal.288.694063519
        Tag: APP_INCR
RMAN>
```

c) Switch to that copy of the image file.

```
RMAN> switch datafile 7 to copy;

datafile 7 switched to datafile copy
"+FRA/orcl/datafile/appraisal.288.694063519"
starting full resync of recovery catalog
full resync complete

RMAN>
```

d) Recover data file 7.

```
RMAN> recover datafile 7;

Starting recover at 2009-08-05:03:43:34
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:01

Finished recover at 2009-08-05:03:43:35
RMAN>
```

3) Bring the APPRAISAL tablespace back online.

```
RMAN> sql "alter tablespace appraisal online";

sql statement: alter tablespace appraisal online
starting full resync of recovery catalog
full resync complete

RMAN>
```

Practice 7-2: Performing Fast Recovery (continued)

4) Report the schema to note the file name for the APPRAISAL tablespace.

```
RMAN> report schema;

Report of database schema for database with db_unique_name
ORCL

List of Permanent Datafiles
===========================
File Size(MB) Tablespace           RB segs Datafile Name
---- -------- -------------------- ------- -------------------
-----
1    680      SYSTEM               YES
+DATA/orcl/datafile/system.260.694050517
2    570      SYSAUX               NO
+DATA/orcl/datafile/sysaux.266.694050521
3    85       UNDOTBS1             YES
+DATA/orcl/datafile/undotbs1.278.694050521
4    5        USERS                NO
+DATA/orcl/datafile/users.264.694050521
5    100      EXAMPLE              NO
+DATA/orcl/datafile/example.259.694050813
6    25       BR_TBS               NO
/home/oracle/BACKUP/br01.dbf
7    25       APPRAISAL            NO
+FRA/orcl/datafile/appraisal.288.694063519

List of Temporary Files
=======================
File Size(MB) Tablespace           Maxsize(MB) Tempfile Name
---- -------- -------------------- ----------- --------------
-----
1    28       TEMP                 32767
+DATA/orcl/tempfile/temp.265.694050795

RMAN>
```

Note that you are now using the Flash Recovery Area data file as the open data file for the online tablespace.

5) Determine what happened to the original data file.

List the image copies for data file 7, to see if it is listed.

```
RMAN> list copy of datafile 7;

List of Datafile Copies
=======================

Key     File S Completion Time     Ckp SCN    Ckp Time
------- ---- - ------------------- ---------- ----------------
---
```

Practice 7-2: Performing Fast Recovery (continued)

```
2248    7    A 2009-08-05:03:42:49 856719     2009-08-
05:03:39:13
        Name: +DATA/orcl/datafile/appraisal.256.694061937
```

Note that it is still there and is listed now as an image copy. But, the reason for switching was because this data file was considered to be on a damaged disk, or the data file itself was corrupted.

6) In a separate terminal window, make sure you are at the `~/labs` directory.

```
$ cd ~/labs
```

7) Remember that the reason for switching was because this data file was considered to be on a damaged disk, or the data file itself was corrupted in some way. So, run the `rm_asm_file.sh` script to delete the file. But first, you must ensure that you are pointing to the +ASM instance.

```
$ . oraenv
ORACLE_SID = [orcl] ? +ASM
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is
/u01/app/oracle
$ asmcmd rm +DATA/orcl/datafile/appraisal.256.694061937
```

8) Eventually, you want to refrain from using the Flash Recovery Area as storage for active data files. By now, the hardware or corruption problem has been remedied, so switch the data file back to the original location.

   a) Back up the data file, specifying the `DATA` ASM disk group, using the `DATAFILE` template.

```
RMAN> backup as copy to destination '+DATA(datafile)' datafile
7;

Starting backup at 2009-08-05:03:49:30
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile copy
input datafile file number=00007
name=+FRA/orcl/datafile/appraisal.288.694063519
output file name=+DATA/orcl/datafile/appraisal.256.694064971
tag=TAG20090805T034930 RECID=28 STAMP=694064971
channel ORA_DISK_1: datafile copy complete, elapsed time:
00:00:03
Finished backup at 2009-08-05:03:49:33

Starting Control File and SPFILE Autobackup at 2009-08-
05:03:49:33
piece
handle=+FRA/orcl/autobackup/2009_08_05/s_694064975.295.6940649
77 comment=NONE
Finished Control File and SPFILE Autobackup at 2009-08-
05:03:49:42
RMAN>
```

Practice 7-2: Performing Fast Recovery (continued)

b) Take the data file offline.

```
RMAN> sql "alter database datafile 7 offline";

sql statement: alter database datafile 7 offline

RMAN>
```

c) Switch the data file to the newly made copy.

```
RMAN> switch datafile 7 to copy;

datafile 7 switched to datafile copy
"+DATA/orcl/datafile/appraisal.256.694064971"
starting full resync of recovery catalog
full resync complete

RMAN>
```

d) Report the schema to confirm that the data file location has changed.

```
RMAN> report schema;

Report of database schema for database with db_unique_name
ORCL

List of Permanent Datafiles
===========================
File Size(MB) Tablespace           RB segs Datafile Name
---- -------- -------------------- ------- -------------------
-----
1    680      SYSTEM               YES
+DATA/orcl/datafile/system.260.694050517
2    570      SYSAUX               NO
+DATA/orcl/datafile/sysaux.266.694050521
3    85       UNDOTBS1             YES
+DATA/orcl/datafile/undotbs1.278.694050521
4    5        USERS                NO
+DATA/orcl/datafile/users.264.694050521
5    100      EXAMPLE              NO
+DATA/orcl/datafile/example.259.694050813
6    25       BR_TBS               NO
/home/oracle/BACKUP/br01.dbf
7    25       APPRAISAL            NO
+DATA/orcl/datafile/appraisal.256.694064971

List of Temporary Files
=======================
File Size(MB) Tablespace           Maxsize(MB) Tempfile Name
---- -------- -------------------- ----------- ---------------
-----
1    28       TEMP                 32767
+DATA/orcl/tempfile/temp.265.694050795
```

Practice 7-2: Performing Fast Recovery (continued)

9) Recover the data file.

```
RMAN> recover datafile 7;

Starting recover at 2009-08-05:03:53:27
using channel ORA_DISK_1

starting media recovery
media recovery complete, elapsed time: 00:00:00

Finished recover at 2009-08-05:03:53:27
RMAN>
```

10) Bring the data file online.

```
RMAN> sql "alter database datafile 7 online";

sql statement: alter database datafile 7 online

RMAN>
```

11) Cross-check the image copy backups and remove the obsolete data file 7 image copy. Exit RMAN when finished.

```
RMAN> crosscheck copy;

released channel: ORA_DISK_1
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=58 device type=DISK
specification does not match any control file copy in the
repository
validation succeeded for datafile copy
datafile copy file
name=+FRA/orcl/datafile/system.291.694063333 RECID=21
STAMP=694064010
..
validation failed for datafile copy
datafile copy file
name=+DATA/orcl/datafile/appraisal.256.694061937 RECID=27
STAMP=694064569
..
name=+FRA/orcl/archivelog/2009_08_05/thread_1_seq_2.326.694061
513 RECID=21 STAMP=694061513
validation succeeded for archived log
archived log file
name=+FRA/orcl/archivelog/2009_08_05/thread_1_seq_3.311.694061
747 RECID=22 STAMP=694061750
Crosschecked 29 objects

RMAN> list expired copy;

specification does not match any archived log in the
repository
```

Practice 7-2: Performing Fast Recovery (continued)

```
List of Datafile Copies
=======================

Key     File S Completion Time     Ckp SCN    Ckp Time
------- ---- - ------------------- ---------- ----------------
---
2248    7    X 2009-08-05:03:42:49 856719     2009-08-
05:03:39:13
        Name: +DATA/orcl/datafile/appraisal.256.694061937

RMAN> delete expired copy;

released channel: ORA_DISK_1
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=58 device type=DISK
specification does not match any control file copy in the
repository
specification does not match any archived log in the
repository
List of Datafile Copies
=======================

Key     File S Completion Time     Ckp SCN    Ckp Time
------- ---- - ------------------- ---------- ----------------
---
2248    7    X 2009-08-05:03:42:49 856719     2009-08-
05:03:39:13
        Name: +DATA/orcl/datafile/appraisal.256.694061937
Do you really want to delete the above objects (enter YES or
NO)? YES
deleted datafile copy
datafile copy file
name=+DATA/orcl/datafile/appraisal.256.694061937 RECID=27
STAMP=694064569
Deleted 1 EXPIRED objects

RMAN> exit
```

Practice 8-1: Monitoring RMAN Jobs

In this practice, you execute a long-running RMAN job and monitor its progress to determine how much longer it is expected to run.

1) Point to the `ORCL` instance, invoke RMAN and delete all obsolete backups. You need to perform a cross-check first, so that the repository is up-to-date.

```
$ . oraenv
orcl
$ rman target / catalog rcatowner@rcat
recovery catalog database Password: oracle_4U <<<not displayed

RMAN> crosscheck backup;
. . .
RMAN> delete noprompt obsolete;
. . .
```

2) Open a second terminal window. Change to the `labs` directory and point to the `20cl` instance. Invoke SQL*Plus and connect as `SYSDBA`. You use this second session to monitor a database backup.

```
$ . oraenv
orcl
$ sqlplus / as sysdba
```

3) Return to your first terminal window. In your RMAN session, begin a database backup to the tape device.

```
RMAN> backup device type sbt database;

Starting backup at 2009-08-05:04:23:41
released channel: ORA_DISK_1
allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=47 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: WARNING: Oracle Test Disk API
channel ORA_SBT_TAPE_1: starting full datafile backup set
channel ORA_SBT_TAPE_1: specifying datafile(s) in backup set
input datafile file number=00001
name=+DATA/orcl/datafile/system.260.694050517
channel ORA_SBT_TAPE_1: starting piece 1 at 2009-08-
05:04:23:42
channel ORA_SBT_TAPE_1: finished piece 1 at 2009-08-
05:04:24:37
piece handle=20klt7qe_1_1 tag=TAG20090805T042342 comment=API
Version 2.0,MMS Ver
sion 8.1.3.0...
```

Practice 8-1: Monitoring RMAN Jobs (continued)

4) Use your SQL*Plus session to monitor the progress of the database backup by querying the `V$SESSION_LONGOPS` view. By using this view, you can determine whether the backup is progressing normally or hanging. If the backup is progressing normally, the `TIME_REMAINING` column should be decreasing. Execute the `query_longops2.sql` script a few times to query `V$SESSION_LONGOPS`. Since the script contains multiple SELECT statements, you cannot enter slash to rerun the script. You must type or paste the script name for each run. Each time you should see the `TIME_REMAINING` value decreasing for the `SID` associated with the backup task. The detail data section pertains to individual channel openings and closings. The aggregate data section describes the entire RMAN job progress.

```
SQL> @query_longops2.sql

Detail Data
--------------------------------------
Detail Progress Information (per file)


       SID START_TIME          ELAPSED_SECONDS TIME_REMAINING
---------- ------------------- --------------- --------------
        40 2009-08-05:05:12:19              20             24


Aggregate Data
------------------------------
Aggregate Progress Information


  SID     SERIAL#     CONTEXT       SOFAR   TOTALWORK %_COMPLETE
------- ---------- ---------- ---------- ---------- -------
   21         934          4       43999      236000      18.64

SQL> @query_longops2.sql
Detail Data
--------------------------------------
Detail Progress Information (per file)


       SID START_TIME          ELAPSED_SECONDS TIME_REMAINING
---------- ------------------- --------------- --------------
        40 2009-08-05:05:13:26              12             35


Aggregate Data
------------------------------
Aggregate Progress Information


    SID     SERIAL#     CONTEXT       SOFAR   TOTALWORK %_COMPLETE
-------- ---------- ---------- ---------- ---------- -------
     21         934          4      131039      236000      55.53
```

Practice 8-1: Monitoring RMAN Jobs (continued)

5) Exit your RMAN and SQL sessions.
6) Run the `cleanup_archivelogs.sh` script to back up and remove archivelog files. This takes several minutes to run.

```
$ cd ~/labs
$ ./cleanup_archivelogs.sh
```

7) Run the `cleanup_tape_dir.sh` script to remove files from the `/tape` directory.

```
$ ./cleanup_tape_dir.sh
```

Practices for Lesson 9

In this lesson you learn about diagnosing and repairing block corruption.
In these practices, you employ various diagnostic capabilities to solve errors.

Practice 9-1: Diagnostic Scenario

In this optional practice, you create a data dictionary corruption that you analyze using Support Workbench and ADRCI. ***If you begin this practice, you must complete it.***

1) Execute the `hm_setup.sh script` from the `labs` subdirectory. This script corrupts the data dictionary.

```
$ . oraenv
orcl
$ cd ~/labs
$ cat hm_setup.sh
#!/bin/bash

cd /home/oracle/labs

sqlplus / as sysdba <<EOF

set echo on
create table scott.tabjmw(c number) tablespace users;
variable obj number;

begin
select object_id into :obj from dba_objects where
owner='SCOTT' and object_name='TABJMW';
end;
/
print obj;
update tab$ set cols=1001 where obj#=:obj;
commit;
EOF

$ ./hm_setup.sh

SQL> SQL> SQL>
Table created.

SQL> SQL> SQL>   2    3    4
PL/SQL procedure successfully completed.

SQL>
       OBJ
----------
     74657

SQL>
1 row updated.

SQL>
Commit complete.
$
```

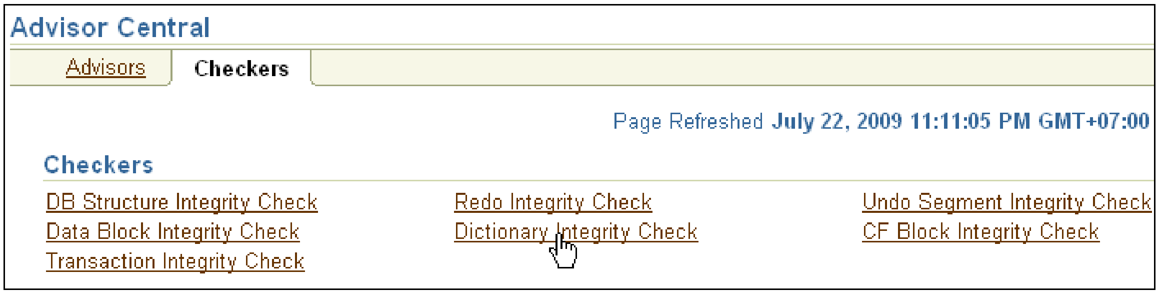
Practice 9-1: Diagnostic Scenario (continued)

2) Use EM to trigger a health check on the data dictionary.

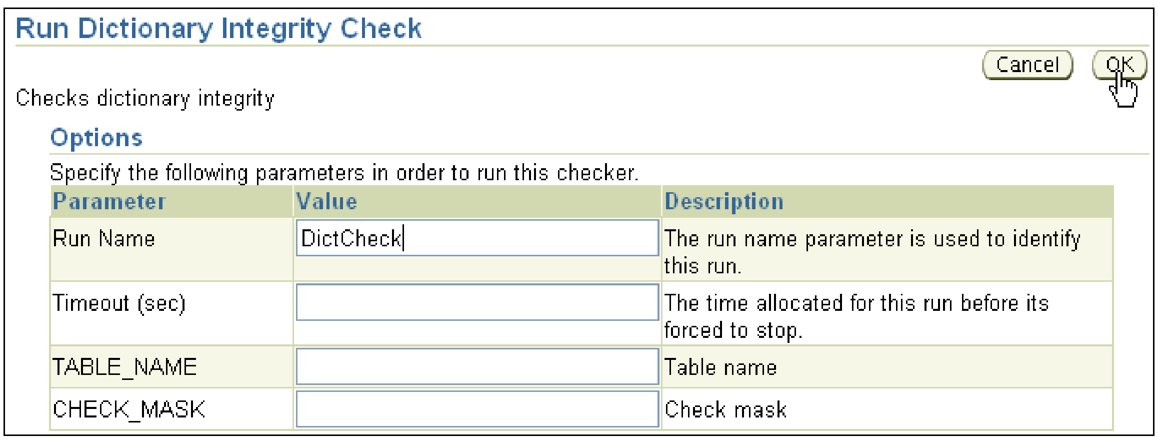
   a) From the Database home page (connected as the SYS user), click Advisor Central at the bottom of the page in the Related Links section.

   b) On the Advisor Central page, click the Checkers tab.

   c) On the Checker tabbed page, click Dictionary Integrity Check.

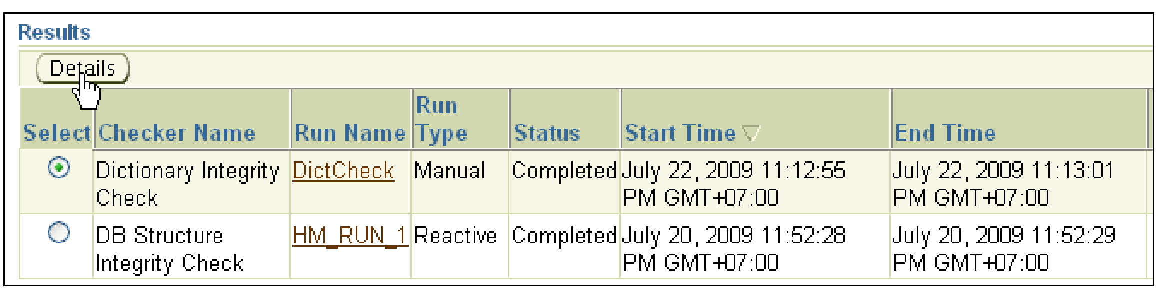


   d) On the Run Dictionary Integrity Check page, specify a Run Name: `DictCheck` and click OK.



   e) Back to the Advisor Central Checkers page, you should see a Confirmation message indicating that your check ran successfully.

   f) Select your DictCheck run and click Details.



   g) This takes you to the Run Details Findings tabbed page, where you can see some Data Corruption. In particular, you should see the following finding: "SQL dictionary health check: invalid column number 8 on object TAB$ failed."

Practice 9-1: Diagnostic Scenario (continued)

Select	Description	Priority	Damage Translation	Incident ID	Status	Time Detected
☐	All Findings					
☐	SQL dictionary health check: invalid column number 8 on object TAB$ failed	Critical	Damaged rowid is AAAAACAABAAAUfPAAC - description: Object SCOTT.TABJMW is referenced		Open	July 22, 2009 11:12:56 PM GMT+07:00

The corruption should correspond to a row in `TAB$` that references `SCOTT.TABJMW`.

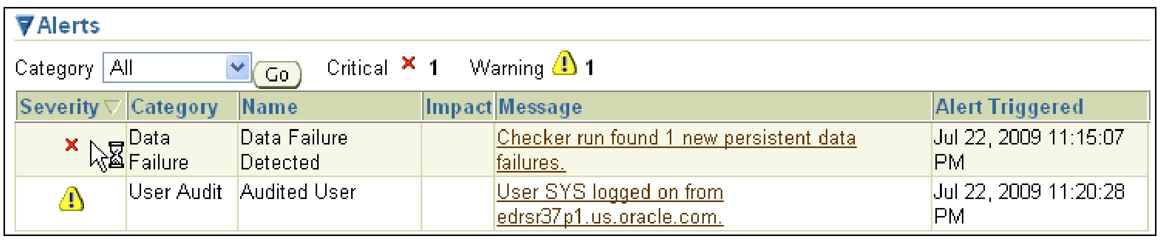
3) How would you get a report about the previous finding using SQL*Plus?

```
$ sqlplus / as sysdba

SQL> set long 100000
SQL> set pages 999

SQL> select dbms_hm.get_run_report('DictCheck') from dual;

DBMS_HM.GET_RUN_REPORT('DICTCHECK')
--------------------------------------------------------------------
------------
Basic Run Information
 Run Name                     : DictCheck

 Run Id                       : 481

 Check Name                   : Dictionary I
ntegrity Check
 Mode                         : MANUAL

 Status                       : COMPLETED

 Start Time                   : 2009-07-22
23:12:55.920739 +07:00
 End Time                     : 2009-07-22 23:13:01.5127
11 +07:00
 Error Encountered            : 0

 Source Incident Id           : 0

 Number of Incidents Created  : 0


Input Paramters for the Run
 TABLE_NAME=ALL_CORE_TABLES
 CHECK_MASK=ALL

Run Findings And Recommendations
 Finding

 Finding Name  : Dictionary Inconsistency

 Finding ID    : 482
```

```
 Type           : FAILURE

 Status         : OPEN

 Priority       : CRITICAL

 Message        : SQL dictionary health check: i
nvalid column number 8 on
               object TAB$ failed

 Message        : Damaged rowid is AAAAACAABAAAU
fPAAC - description: Object
               SCOTT.TABJMW is referenced


SQL>
```

4) Navigate to your EM Home page. What do you observe in the Alerts section?

   You should see a new critical alert (Data Failure) for the previously detected corruption in the Alerts section of the Home page. (*It may take a while to appear*.)



Alerts

Category All Go Critical 1 Warning 1

Severity	Category	Name	Impact	Message	Alert Triggered
Critical	Data Failure	Data Failure Detected		Checker run found 1 new persistent data failures.	Jul 22, 2009 11:15:07 PM
Warning	User Audit	Audited User		User SYS logged on from edrsr37p1.us.oracle.com.	Jul 22, 2009 11:20:28 PM

5) Flush your shared pool and buffer cache first, and then exit and reconnect as the SYS user.

```
$ sqlplus / as sysdba
SQL> alter system flush shared_pool;
System altered.

SQL>
SQL> alter system flush buffer_cache;
System altered.

SQL> exit
```

```
$ sqlplus / as sysdba
```

6) From a SQL*Plus session connected as the SYS user, execute the following statement: `select * from scott.tabjmw;`
   Question: What do you observe?

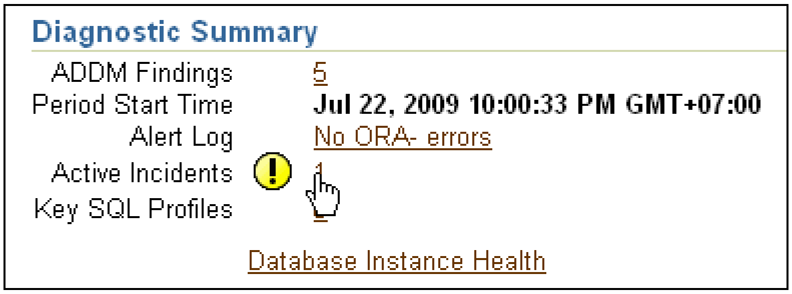
```
SQL> select * from scott.tabjmw;

select * from scott.tabjmw
*
ERROR at line 1:
```

```
ORA-03113: end-of-file on communication channel
Process ID: 14872
Session ID: 66 Serial number: 6259
SQL>
```

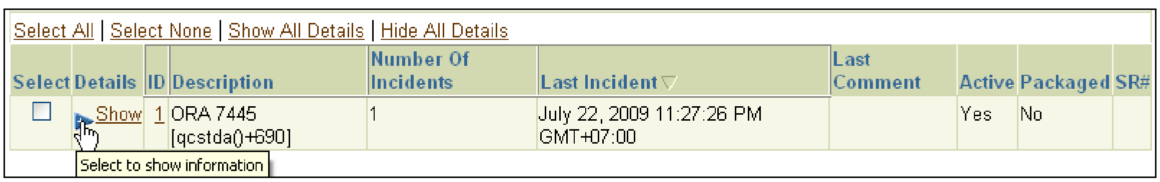
Answer: At first the session seems to hang, then the `ORA-03113` error is displayed.

7) Use EM to investigate the previous problem.

   a) From the EM Home page, you should see one active incident. Click the Active Incident link. If not, click Software and Support. On the Software and Support tabbed page, click Support Workbench in the Support section.



   b) On the Support Workbench Problems tabbed page, you should see a new Active Incident whose description is similar to `ORA 7445 [qcstda()+690]`.

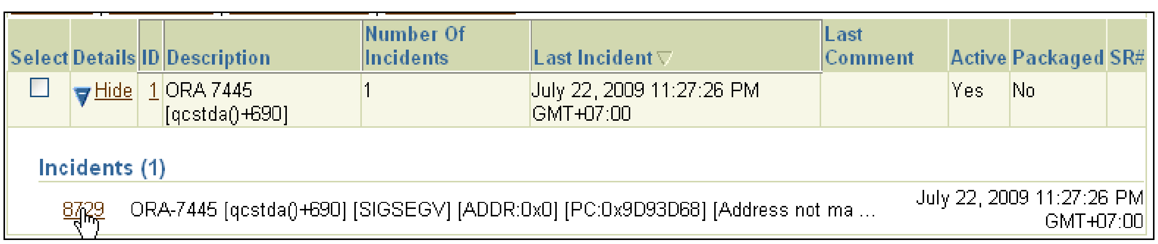


   c) Click the "+" icon in the Details column for this problem.

   d) This shows you the corresponding first-time incident.

   e) Make a note of the incident number (In this screenshot it is 8729).

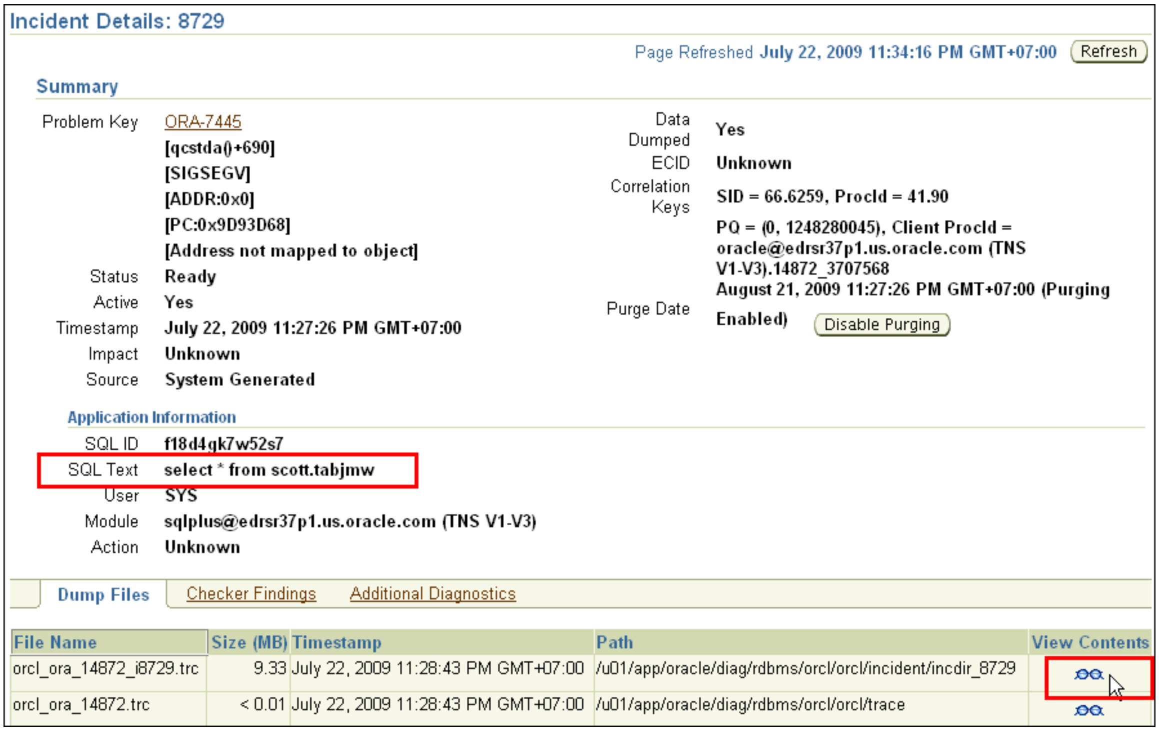
   f) Click that incident number.



   g) This takes you to the corresponding Incident Details page.

   h) On the Incident Details page, in the Application Information section, you can see the cause the incident. It is: "`select * from scott.tabjmw`".

Practice 9-1: Diagnostic Scenario (continued)



You can also see two dump files generated from the Dump Files tabbed page. The first one corresponds to the incident dump file and is located in the ADR directory `/u01/app/oracle/diag/rdbms/orcl/orcl/incident/incdir_`*`nnnnn`*. The second one is the classical corresponding trace file located in `/u01/app/oracle/diag/rdbms/orcl/orcl/trace`.

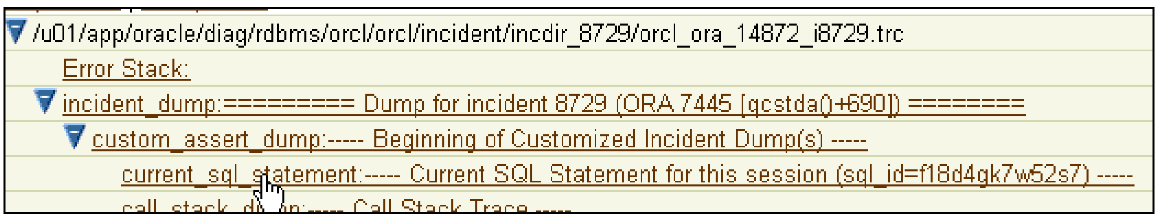
h) Click the eyeglasses icon in the View Contents column for the incident trace (first row).

i) If you did not save your Host credentials previously, then the Host Credentials page is displayed. Enter `oracle` as username and enter `oracle` as password, then click Continue.

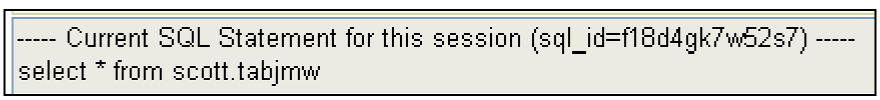
8) On the Contents page, you see the structured trace information. Expand the `incident_dump` and `custom_assert_dump` nodes.

Practice 9-1: Diagnostic Scenario (continued)

9) Then click the "current sql statement" link.

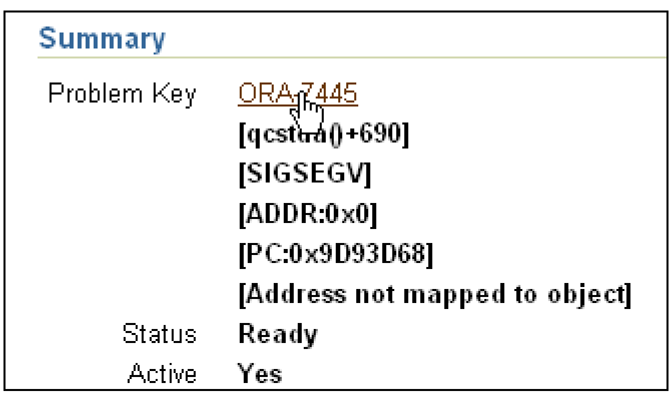


10) This prints the culprit SQL statement in the window below the Trace Map.



11) Click OK.

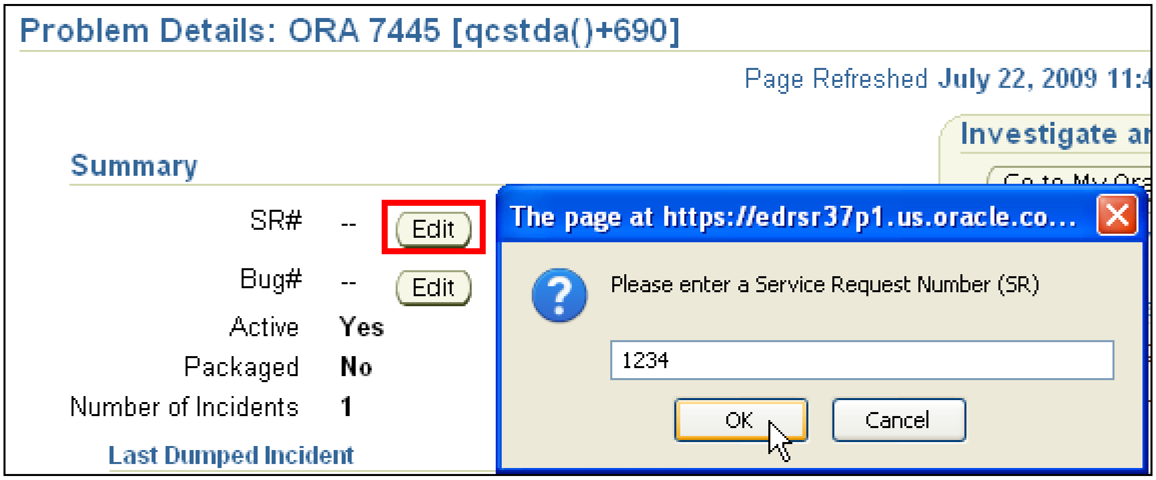
12) On the Incident Details page again, click the Problem Key link.



13) On the Problem Details page, click Edit to the right of the SR field.

14) Enter `1234` in the SR Number pop-up window that appears and click OK. It is assumed that you already opened an SR (1234) with MetaLink.

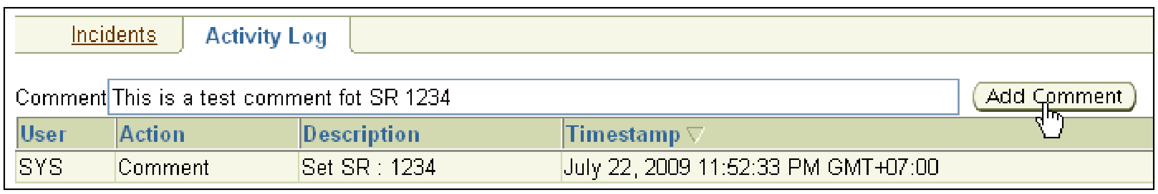
Practice 9-1: Diagnostic Scenario (continued)



15) You should now see the SR number on the Problem Details page.

16) Click the Activity Log tab.

17) This takes you to the Activity Log tabbed page, on which you can now see your last action on the problem. You can optionally enter a comment here. (If you do, click Add Comment to save your comment.)



18) Back to your terminal window, locate your incident and your health check report in ADR. (Your incident number may be different.)

```
$ cd $ORACLE_BASE/diag/rdbms/orcl/orcl

$ ls
alert  cdump  hm  incident  incpkg  ir  lck  metadata  stage
sweep  trace
$ cd incident
$ ls -la
total 12
drwxr-x---  3 oracle dba 4096 Jul 22 23:27 .
drwxr-x--- 13 oracle dba 4096 Jul 20 23:49 ..
drwxr-xr-x  2 oracle dba 4096 Jul 22 23:27 incdir_8729

$ cd incdir_8729

$ ls
orcl_ora_14872_i8729.trc  orcl_ora_14872_i8729.trm

$ cd ../..
$ ls
alert  cdump  hm  incident  incpkg  ir  lck  metadata  stage
sweep  trace
```

Practice 9-1: Diagnostic Scenario (continued)

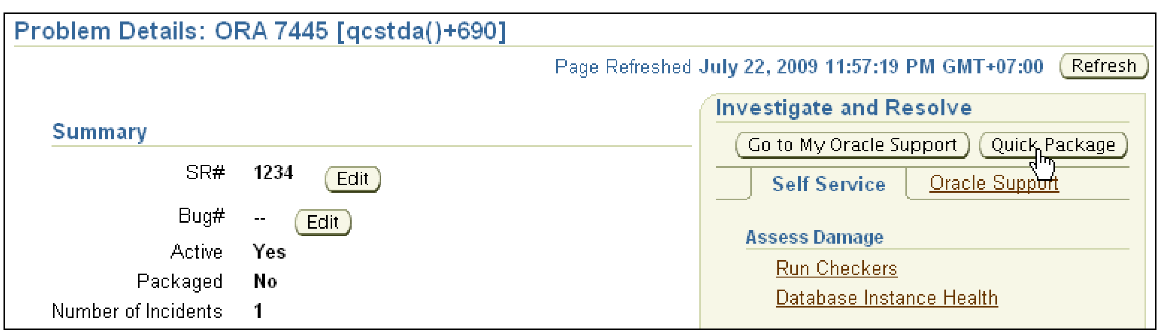
```
$ cd hm
$ ls
HMREPORT_DictCheck.hm
$ cd ..
$
```

19) Use ADRCI to locate your incident and problem information.

```
$ adrci

ADRCI: Release 11.2.0.0.2 - Beta on Thu Jun 4 20:14:32 2009
ADRCI: Release 11.2.0.1.0 - Production on Thu Jul 23 01:04:39
2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates.  All
rights reserved.

ADR base = "/u01/app/oracle"
adrci> set homepath diag/rdbms/orcl/orcl

adrci> show homes
ADR Homes:
diag/rdbms/orcl/orcl
adrci> show incidents
ADR Home = /u01/app/oracle/diag/rdbms/orcl/orcl:
**************************************************************
***********
INCIDENT_ID          PROBLEM_KEY
CREATE_TIME
-------------------- -----------------------------------------
------------------- ----------------------------------------
8729                 ORA 7445 [qcstda()+690]
2009-07-22 23:27:26.129000 +07:00
1 rows fetched
adrci>
```

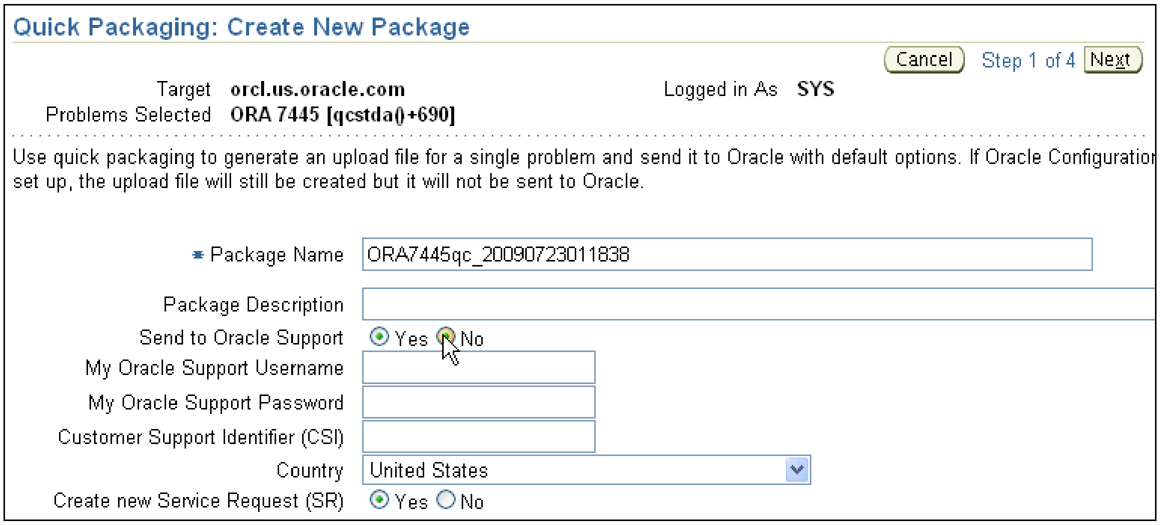
20) Use Enterprise Manager to package your problem quickly.

21) From the Problem Details page, click Quick Package in the Investigate and Resolve section.

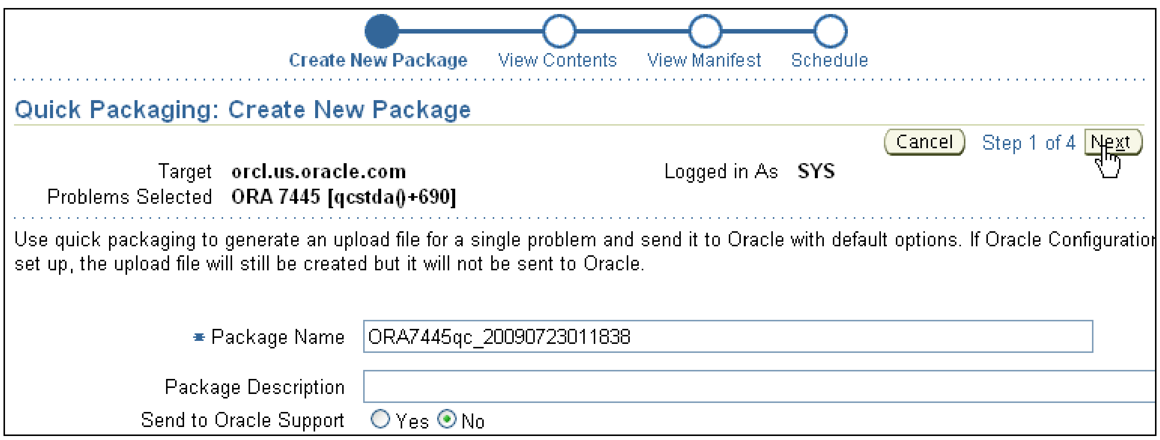


Practice 9-1: Diagnostic Scenario (continued)

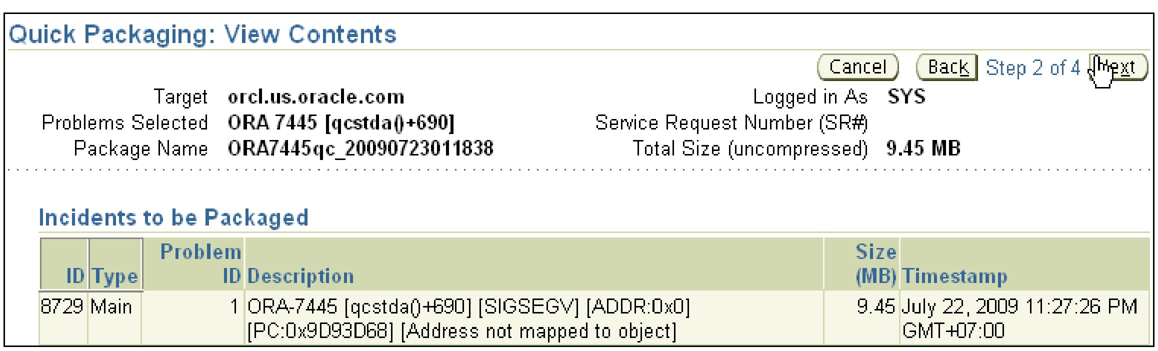
22) On the Quick Packaging: Create New Package page, leave the default package name (something similar to `ORA7445qc_20090723011838`) and select No for "Send to Oracle Support" entry.



23) When the page is refreshed, click Next.



24) On the Quick Packaging: View Contents page, you can see the list of incidents that are part of this package. There should be only one incident in your case. Click Next.

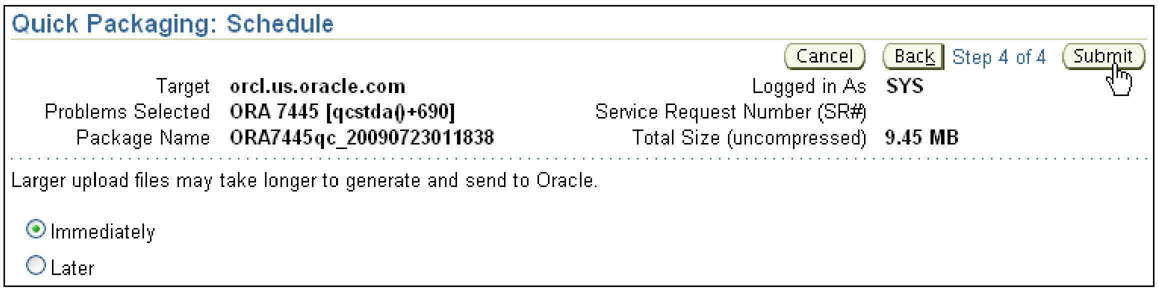


25) On the Quick Packaging: View Manifest page, look at the package name and the path.

Practice 9-1: Diagnostic Scenario (continued)

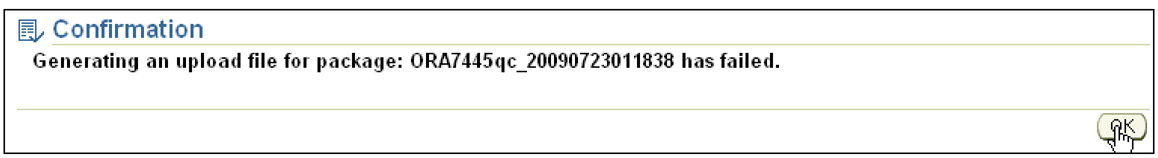
26) Click Next.

27) On the Quick Packaging: Schedule page, ensure Immediately is selected and click Submit.



28) The Processing: Generating Upload File for Package page appears. Wait until it is finished.

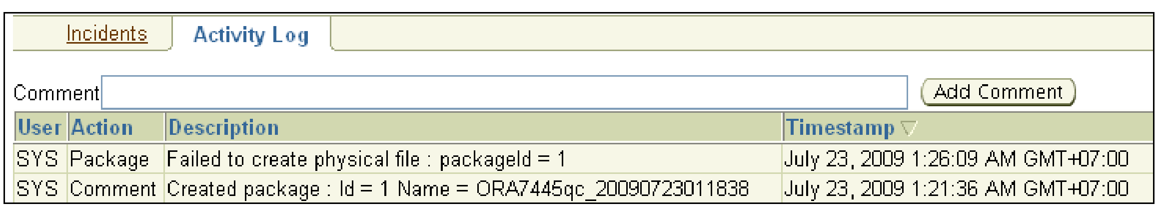
29) On the Confirmation page, you should see something similar to: "Generating an upload file for package: ORA7445qc_20090723011838 has failed."



30) Click OK.

31) On the Problem Details page, click the Activity Log tab.

32) Back to the Problem Details Activity Log tabbed page, you should now see two new entries in the log reporting about the package creation for this problem.



33) In the Summary section, click Yes to the right of the Packaged entry.

34) This takes you to the Packages page, from where you can see your package.

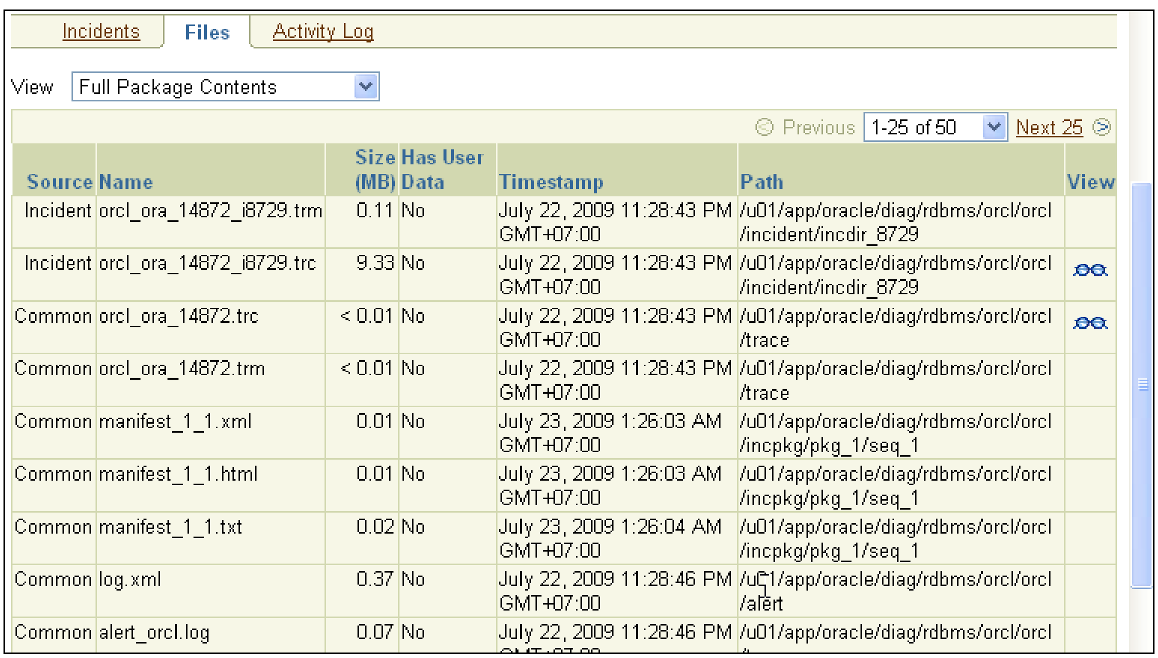
Practice 9-1: Diagnostic Scenario (continued)

35) Select your package, and click View.

ID	Type	Problem ID	Description	Size (MB)	Timestamp
8729	Main	1	ORA-7445 [qcstda()+690] [SIGSEGV] [ADDR:0x0] [PC:0x9D93D68] [Address not mapped to object]	0	July 22, 2009 11:27:26 PM GMT+07:00

This takes you to the Package Details page, where you can see the package details.

36) Click the Files tab.



Source	Name	Size (MB)	Has User Data	Timestamp	Path	View
Incident	orcl_ora_14872_i8729.trm	0.11	No	July 22, 2009 11:28:43 PM GMT+07:00	/u01/app/oracle/diag/rdbms/orcl/orcl /incident/incdir_8729	
Incident	orcl_ora_14872_i8729.trc	9.33	No	July 22, 2009 11:28:43 PM GMT+07:00	/u01/app/oracle/diag/rdbms/orcl/orcl /incident/incdir_8729	
Common	orcl_ora_14872.trc	< 0.01	No	July 22, 2009 11:28:43 PM GMT+07:00	/u01/app/oracle/diag/rdbms/orcl/orcl /trace	
Common	orcl_ora_14872.trm	< 0.01	No	July 22, 2009 11:28:43 PM GMT+07:00	/u01/app/oracle/diag/rdbms/orcl/orcl /trace	
Common	manifest_1_1.xml	0.01	No	July 23, 2009 1:26:03 AM GMT+07:00	/u01/app/oracle/diag/rdbms/orcl/orcl /incpkg/pkg_1/seq_1	
Common	manifest_1_1.html	0.01	No	July 23, 2009 1:26:03 AM GMT+07:00	/u01/app/oracle/diag/rdbms/orcl/orcl /incpkg/pkg_1/seq_1	
Common	manifest_1_1.txt	0.02	No	July 23, 2009 1:26:04 AM GMT+07:00	/u01/app/oracle/diag/rdbms/orcl/orcl /incpkg/pkg_1/seq_1	
Common	log.xml	0.37	No	July 22, 2009 11:28:46 PM GMT+07:00	/u01/app/oracle/diag/rdbms/orcl/orcl /alert	
Common	alert_orcl.log	0.07	No	July 22, 2009 11:28:46 PM	/u01/app/oracle/diag/rdbms/orcl/orcl	

On the Files tabbed page, you can now see the Full Package Content view.

37) Use ADRCI to determine the list of existing packages.

```
$ adrci

ADRCI: Release 11.1.0.5.0 - Beta on Thu May 24 07:57:13 2007
```

```
Copyright (c) 1982, 2007, Oracle.  All rights reserved.

ADR base = "/u01/app/oracle"
adrci> set homepath diag/rdbms/orcl/orcl
adrci> query ips_package

ADR Home = /u01/app/oracle/diag/rdbms/orcl/orcl:
**************************************************************
***********
PACKAGE_ID           FLAGS                STATUS
CREATION_TIME                            LAST_SEQUENCE
LAST_COMPLETE_SEQUEN PROBLEM_ID           NAME
DESCRIPTION
CORRELATION_LEVEL    DRIVING_INCIDENT     BEGIN_TIME
END_TIME                                 UPDATE_TIME
LAST_BASE_SEQUENCE
-------------------- -------------------- --------------------
---------------------------------------- --------------------
-------------------- -------------------- --------------------
--------------------------------------------------------------
------------------- ------------------------------------------
--------------------------------------------------------------
--------------------------------------------------------------
--------------------------------------------------------------
--------------------------------------------------------------
----------- -------------------- -------------------- --------
-------------------------------- -----------------------------
----------- ---------------------------------------- ---------
-----------
1                    0                    4
2009-07-23 01:21:35.798774 +07:00        1
1                    1
ORA7445qc_20090723011838
2                    8729
2009-07-23 01:21:35.919160 +07:00        0
1 rows fetched

adrci> exit
```

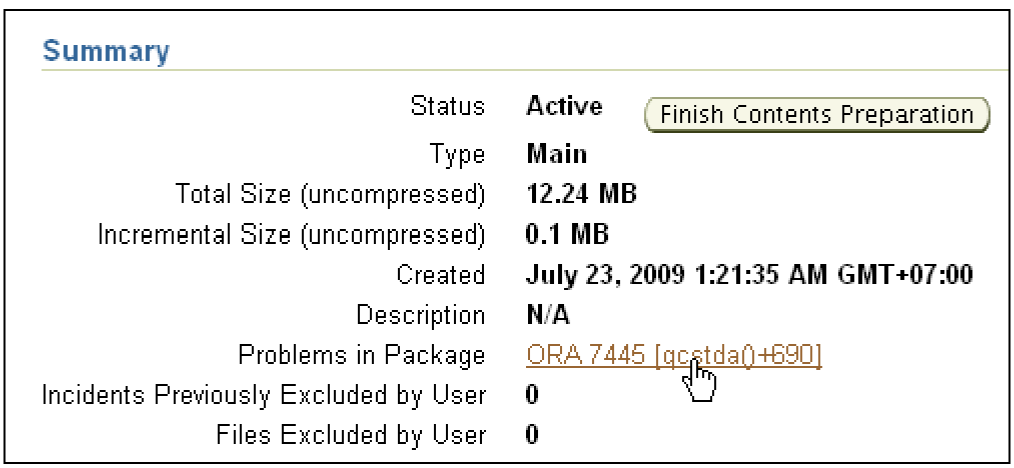
38) After analyzing the problem, you know that the number of columns for `TABJMW` is wrong in `TAB$`. Fix the issue by updating `TAB$` and close your problem. You can execute `hm_fix.sh` from the `labs` subdirectory for that purpose.

```
$ cd ~/labs
$ cat hm_fix.sh
#!/bin/bash
# For training purpose only

sqlplus / as sysdba <<EOF
set echo on
variable obj number;
```

```
begin
select object_id into :obj from dba_objects where
owner='SCOTT' and object_name='TABJMW';
end;
/
print obj;
update tab$ set cols=1 where obj#=:obj;
commit;
EOF
$ ./hm_fix.sh
SQL> SQL> SQL>   2    3    4
PL/SQL procedure successfully completed.

SQL>
       OBJ
----------
     74657

SQL>
1 row updated.

SQL>
Commit complete.
$
```

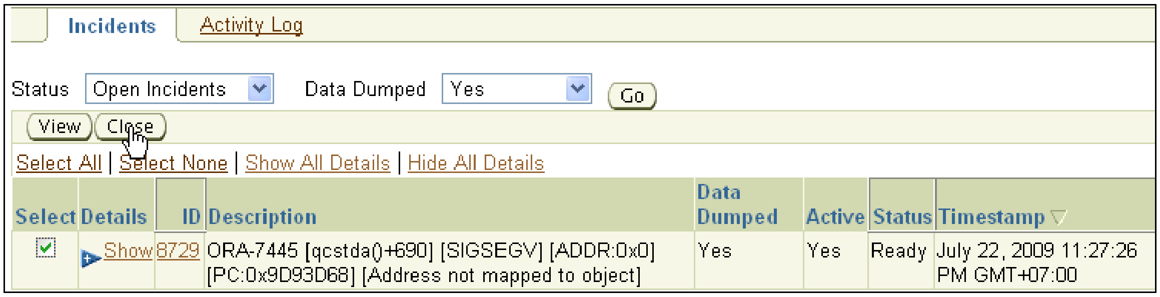
39) Use Enterprise Manager to close your incident.

40) On the Package Details page, click the “problem” link to the right of the “Problems in Package” field in the Summary section.



41) On the Problem Details Incidents tabbed page, select your incident and click Close.

Practice 9-1: Diagnostic Scenario (continued)



42) On the Confirmation page, click Yes.

Practice 9-2: Repairing Block Corruption

1) To set up this block corruptions practice, use a terminal window, navigate to the `$HOME/labs` directory, and execute the `bc_setup.sh` script. This script creates a tablespace called `BCTBS` and a user called `BC`. The tablespace is then populated. A backup of the new tablespace is performed.

   Take a few moments to inspect each script actions before executing it. As usual you point to the `orcl` instance.

```
$ cd ~/labs
$ . oraenv
ORACLE_SID = [orcl] ? orcl

$ cat bc_setup.sh
#!/bin/bash
# For training purpose only
# Run as oracle OS user

sqlplus -S /nolog > /tmp/setup.log 2>&1 <<EOF
connect / as sysdba

-- CLEANUP from previous run
DROP USER bc CASCADE;

DROP TABLESPACE bctbs INCLUDING CONTENTS AND DATAFILES;

-- Create tablespace
CREATE TABLESPACE bctbs
DATAFILE '/home/oracle/BACKUP/bctbs01.dbf' SIZE 10M
SEGMENT SPACE MANAGEMENT MANUAL;

-- Create user
CREATE USER bc IDENTIFIED BY oracle_4U
DEFAULT TABLESPACE bctbs
QUOTA UNLIMITED ON bctbs;

GRANT CREATE SESSION TO bc;

-- create table and populate
-- be sure table is at least 2 blocks long
CREATE TABLE bc.bccopy
TABLESPACE bctbs
AS SELECT * FROM HR.EMPLOYEES;

INSERT INTO bc.bccopy
SELECT * FROM bc.bccopy;

INSERT INTO bc.bccopy
SELECT * FROM bc.bccopy;
EOF
```

```
#-- Create backup of the bctbs tablespace

rman target / > /tmp/rman.log 2>&1 <<EOF

BACKUP AS COPY TABLESPACE bctbs;
EOF

#-- update the table
sqlplus -S /nolog >> /tmp/setup.log 2>&1 <<EOF
connect / as sysdba

UPDATE bc.bccopy SET salary = salary+1;

COMMIT;

EOF
```

```
$  ./bc_setup.sh
```

2) Log into SQL*Plus as the SYS user and execute the `bc_critical` script, which corrupts the data file. Enter your displayed block number.

```
$ sqlplus / as sydba

SQL> @bc_critical
Connected.

   FILE_NO   BLOCK_NO
---------- ----------
         7        129

System altered.

'Enter Block number when prompted'
Enter value for block_no: 129
0+1 records in
0+1 records out
80 bytes (80 B) copied, 6.1724e-05 seconds, 1.3 MB/s
0+1 records in
0+1 records out
79 bytes (79 B) copied, 5.7166e-05 seconds, 1.4 MB/s

SELECT * from bc.bccopy
                 *
ERROR at line 1:
ORA-01578: ORACLE data block corrupted (file # 7, block # 129)
ORA-01110: data file 7: '/home/oracle/BACKUP/bctbs01.dbf'

SQL>
```

Practice 9-2: Repairing Block Corruption (continued)

3) Use RMAN to back up the TBSBC tablespace. What happens?

```
$ rman target / nocatalog
RMAN> backup tablespace bctbs;

Starting backup at 2009-07-23:16:21:16
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=1 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00007
name=/home/oracle/BACKUP/bctbs01.dbf
channel ORA_DISK_1: starting piece 1 at 2009-07-23:16:21:16
RMAN-00571:
===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS
===============
RMAN-00571:
===========================================================
RMAN-03009: failure of backup command on ORA_DISK_1 channel at
07/23/2009 16:21:23
ORA-19566: exceeded limit of 0 corrupt blocks for file
/home/oracle/BACKUP/bctbs01.dbf

RMAN> exit
```

The backup fails upon encountering the first corrupt block.

4) Use the RMAN `list failure` command to provide details about the errors.

```
$ rman target / nocatalog

Recovery Manager: Release 11.1.0.6.0 - Production on Mon Aug
27 09:49:50 2007

connected to target database: ORCL (DBID=1158576740)

RMAN> list failure detail;

RMAN> List of Database Failures
=========================

Failure ID Priority Status    Time Detected       Summary
---------- -------- --------- ------------------- -------
562        HIGH     OPEN      2009-07-23:16:11:04 Datafile 7:
'/home/oracle/BACKUP/bctbs01.dbf' contains one or more corrupt
blocks
  Impact: Some objects in tablespace BCTBS might be
unavailable
  List of child failures for parent failure ID 562
  Failure ID Priority Status    Time Detected       Summary
```

Practice 9-2: Repairing Block Corruption (continued)

```
  ---------- -------- --------- ------------------- -------
  571        HIGH     OPEN      2009-07-23:16:11:05 Block 130
in datafile 7: '/home/oracle/BACKUP/bctbs01.dbf' is media
corrupt
    Impact: Object BCCOPY owned by BC might be unavailable
  565        HIGH     OPEN      2009-07-23:16:11:04 Block 129
in datafile 7: '/home/oracle/BACKUP/bctbs01.dbf' is media
corrupt
    Impact: Object BCCOPY owned by BC might be unavailable
```

Note: **Do not close this RMAN session.**

5) Alternatively, in another terminal window, you can query the `V$IR_FAILURE` view to view details about the failure:

```
$ sqlplus / as sysdba
SQL> set pages 999
SQL> select failure_id, time_detected, description, impacts
from V$IR_FAILURE where impacts like '%BC%';

FAILURE_ID TIME_DETECTED
---------- -------------------
DESCRIPTION
--------------------------------------------------------------
------------------
IMPACTS
--------------------------------------------------------------
------------------
       562 2009-07-23:16:11:04
Datafile 7: '/home/oracle/BACKUP/bctbs01.dbf' contains one or
more corrupt block
s
Some objects in tablespace BCTBS might be unavailable

       565 2009-07-23:16:11:04
Block 129 in datafile 7: '/home/oracle/BACKUP/bctbs01.dbf' is
media corrupt
Object BCCOPY owned by BC might be unavailable

       571 2009-07-23:16:11:05
Block 130 in datafile 7: '/home/oracle/BACKUP/bctbs01.dbf' is
media corrupt
Object BCCOPY owned by BC might be unavailable


SQL>
SQL> exit
```

Practice 9-2: Repairing Block Corruption (continued)

6) Continue in your RMAN session. Execute the RMAN `advise failure all` command to view recovery options.

```
RMAN> advise failure all;
List of Database Failures
=========================

Failure ID Priority Status    Time Detected       Summary
---------- -------- --------- ------------------- -------
562        HIGH     OPEN      2009-07-23:16:11:04 Datafile 7:
'/home/oracle/BACKUP/bctbs01.dbf' contains one or more corrupt
blocks

analyzing automatic repair options; this may take some time
using channel ORA_DISK_1
allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=24 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: WARNING: Oracle Test Disk API
analyzing automatic repair options complete

Mandatory Manual Actions
========================
no manual actions available

Optional Manual Actions
=======================
no manual actions available

Automated Repair Options
========================
Option Repair Description
------ ------------------
1      Recover multiple corrupt blocks in datafile 7
  Strategy: The repair includes complete media recovery with
no data loss
  Repair script:
/u01/app/oracle/diag/rdbms/orcl/orcl/hm/reco_3952673413.hm

RMAN>
```

Note: An `advise failure` command **must** precede a `repair` command. So, do not execute other commands, but continue with the next step.

Practice 9-2: Repairing Block Corruption (continued)

7) Continue in your RMAN session and execute the RMAN `repair failure` command . Enter **YES**, when asked if you want to execute the repair.
   Note: Your actual script name is most likely different than the example.

```
RMAN> repair failure;

Strategy: The repair includes complete media recovery with no
data loss
Repair script:
/u01/app/oracle/diag/rdbms/orcl/orcl/hm/reco_3952673413.hm

contents of repair script:
   # block media recovery for multiple blocks
   recover datafile 7 block 129 to 130;

Do you really want to execute the above repair (enter YES or
NO)? YES
executing repair script

Starting recover at 2009-07-23:16:35:34
using channel ORA_DISK_1
using channel ORA_SBT_TAPE_1

ORA-01013: user requested cancel of current operatio
channel ORA_SBT_TAPE_1: restoring block(s) from datafile copy
+FRA/orcl/datafile/bctbs.296.692986085

starting media recovery
media recovery complete, elapsed time: 00:00:03

Finished recover at 2009-07-23:16:35:38
repair failure complete

RMAN>
```

8) List all failures in RMAN. There should by none.

```
RMAN> list failure all;

no failures found that match specification

RMAN> exit
```

9) To clean up the practice environment, run the `bc_cleanup.sh` script from the `$HOME/labs` directory.

```
$ cat bc_cleanup.sh
#!/bin/bash
# For training purpose only
#-- Cleanup the tablespace, and user from the corrupt blocks

```

Practice 9-2: Repairing Block Corruption (continued)

```
sqlplus -S /nolog > /tmp/cleanup.lo 2>&1 <<EOF
connect / as sysdba

-- CLEANUP from previous run
DROP USER bc CASCADE;

DROP TABLESPACE bctbs INCLUDING CONTENTS AND DATAFILES;

EXIT;
EOF
$ ./bc_cleanup.sh
```

Practices for Lesson 10

Flashback Transaction Backout is a logical recovery option to roll back a specific transaction and dependent transactions while the database remains online. A dependent transaction is related by a write-after-write (WAW) relationship, in which a transaction modifies the same data that was changed by the target transaction, or a primary-key constraint relationship, in which a transaction reinserts the primary-key value that was deleted by the target transaction. Flashback Transaction utilizes undo and the redo generated for undo blocks to create and execute a compensating transaction for reverting the affected data to its original state.

Practice 10-1: Flashback Transaction Backout

In this practice, you view a demonstration of backing out erroneous transactions.

1) Click the `oracle's Home` icon on your desktop.
2) Navigate to the `/home/oracle/demos/fbt_backout` directory.
3) Double-click the `fbt_backout_viewlet_swf.html` file.
4) In the Run or Display window, click Display and view the presentation.
5) Use the controls at the bottom of the viewlet window to start, pause and stop the presentation, as suits your personal learning style.
6) Uninterrupted viewing of the demos takes about ten minutes. When you have finished viewing the presentation, close your Web browser window.

Practices for Lesson 11

These practices cover the use of Flashback Data Archive.

Practice 11-1: Using Flashback Data Archive

In this practice, you use Oracle Total Recall.

1) Log into SQL*Plus as the `SYS` user and execute the `flada_setup` script from the `$HOME/labs` directory. The setup script creates a second undo tablespace, a small `FLA_TBS1` tablespace, and the `ARCHIVE_ADMIN` user with the `oracle_4U` password. The password is case-sensitive by default. Take a few moments to inspect the script before executing it.

```
$ sqlplus / as sysdba

SQL> @flada_setup
Connected.
SQL> set serveroutput on
SQL> -- set verify on
SQL> set term on
SQL> set lines 200
SQL> set pages 44
SQL> set pause on
SQL>
SQL> /*== Create a tablespace for your flashback data archive
==*/
SQL> DROP TABLESPACE fla_tbs1 INCLUDING CONTENTS
  2  /
DROP TABLESPACE fla_tbs1 INCLUDING CONTENTS
*
ERROR at line 1:
ORA-00959: tablespace 'FLA_TBS1' does not exist

SQL> CREATE SMALLFILE TABLESPACE fla_tbs1
  2  DATAFILE '$HOME/BACKUP/fla_tbs01.dbf'
  3  SIZE 10M REUSE AUTOEXTEND ON NEXT 640K MAXSIZE 32767M
  4  NOLOGGING EXTENT MANAGEMENT LOCAL SEGMENT SPACE
MANAGEMENT AUTO
  5  /
Tablespace created.
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...

SQL>
SQL> /*== Create a second undo tablespace for testing ==*/
SQL> DROP TABLESPACE undotbs2 INCLUDING CONTENTS
  2  /
DROP TABLESPACE undotbs2 INCLUDING CONTENTS
*
ERROR at line 1:
ORA-00959: tablespace 'UNDOTBS2' does not exist

SQL> CREATE SMALLFILE UNDO TABLESPACE undotbs2
  2  DATAFILE '$HOME/BACKUP/undotbs02.dbf'
  3  SIZE 105M REUSE AUTOEXTEND ON NEXT 5120K MAXSIZE 32767M
```

```
  4  /

Tablespace created.

SQL> pause Press [Enter] to continue...
Press [Enter] to continue...

SQL> /*== Create an ARCHIVE_ADMIN user like the HR user ==*/
SQL> /*==   with FLA_TBS1 default tablespace ==*/
SQL> CREATE USER ARCHIVE_ADMIN PROFILE DEFAULT IDENTIFIED BY
"oracle_4U"
  2  DEFAULT TABLESPACE FLA_TBS1 TEMPORARY TABLESPACE TEMP
  3  ACCOUNT UNLOCK;

User created.

SQL> pause Press [Enter] to continue...
Press [Enter] to continue...

SQL> GRANT ALTER SESSION TO ARCHIVE_ADMIN;
Grant succeeded.

SQL> GRANT CREATE DATABASE LINK TO ARCHIVE_ADMIN;
Grant succeeded.

SQL> GRANT CREATE SEQUENCE TO ARCHIVE_ADMIN;
Grant succeeded.

SQL> GRANT CREATE SESSION TO ARCHIVE_ADMIN;
Grant succeeded.

SQL> GRANT CREATE SYNONYM TO ARCHIVE_ADMIN;
Grant succeeded.

SQL> GRANT CREATE VIEW TO ARCHIVE_ADMIN;
Grant succeeded.

SQL> GRANT UNLIMITED TABLESPACE TO ARCHIVE_ADMIN;
Grant succeeded.

SQL> GRANT EXECUTE ON SYS.DBMS_STATS TO ARCHIVE_ADMIN;
Grant succeeded.

SQL> GRANT CONNECT, RESOURCE TO ARCHIVE_ADMIN;
Grant succeeded.
SQL>
SQL> /*== Setup for Flashback Data Archive completed ==*/
SQL> /*== The ARCHIVE_ADMIN user has the password: oracle_4U
==*/
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
SQL>
```

Practice 11-1: Using Flashback Data Archive (continued)

2) To give the ARCHIVE_ADMIN user administrative privileges for creating, maintaining, and dropping flashback data archives, execute the following command as the SYS user:
   GRANT FLASHBACK ARCHIVE ADMINISTER TO archive_admin;

```
$ sqlplus / as sysdba
SQL> GRANT FLASHBACK ARCHIVE ADMINISTER TO archive_admin;

Grant succeeded.

SQL> exit
```

3) In SQL*Plus, connect as the ARCHIVE_ADMIN user with the ARCHIVE_ADMIN password.
   Note: The password is case-sensitive; the username is not case-sensitive.

```
$ sqlplus archive_admin
Enter password: oracle_4U  <<< not displayed

SQL>
```

4) To create a flashback data archive, execute the following command:

   CREATE FLASHBACK ARCHIVE fla1
   TABLESPACE fla_tbs1
   QUOTA 10M
   RETENTION 1 YEAR
   /

   Entering the command is recommended for better retention, but if typing is a difficulty, you can also execute the flada_create.sql script.

```
SQL> CREATE FLASHBACK ARCHIVE fla1
TABLESPACE fla_tbs1
QUOTA 10M
RETENTION 1 YEAR
/  2    3    4    5

Flashback archive created.

SQL>
```

5) Give the privilege to use the FLA1 archive to the HR user, by executing the following command:

   GRANT FLASHBACK ARCHIVE on FLA1 to HR;

```
SQL> GRANT FLASHBACK ARCHIVE on FLA1 to HR;

Grant succeeded.

SQL>
```

Practice 11-1: Using Flashback Data Archive (continued)

6) You now switch to the role of a flashback archive user. Connect as the `HR` user with the `oracle_4U` password. To enable this flashback data archive for the `EMPLOYEES` table, execute the following command:
`ALTER TABLE hr.employees FLASHBACK ARCHIVE fla1;`

```
SQL> connect HR
Enter password: oracle_4U <<< not displayed
Connected.
SQL> ALTER TABLE hr.employees FLASHBACK ARCHIVE fla1;

Table altered.

SQL>
```

7) To view and increase the salary of Mr. Fox three times by 1000, execute the `flada_dml` script as the `HR` user. This produces activity in the flashback data archive.

```
SQL> @flada_dml
SQL> REM "*********************************************** "
SQL> REM "For demo purposes ONLY: Flashback Data Archive "
SQL>
SQL> set echo on
SQL> set serveroutput on
SQL> -- set verify on
SQL> set term on
SQL> set lines 200
SQL> set pages 44
SQL> set pause on pause "Press [Enter] to continue..."
SQL>
SQL> /*== Query the current salary for Mr. Fox ==*/
SQL>
SQL> SELECT employee_id, last_name, salary
  2  FROM   hr.employees
  3  WHERE  last_name = 'Fox'
  4  /
Press [Enter] to continue...

EMPLOYEE_ID LAST_NAME                     SALARY
----------- ------------------------- ----------
        170 Fox                             9600

1 row selected.

SQL>
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...

SQL>
SQL> /*== Increase the  salary three times by 1000 ==*/
SQL>
SQL> UPDATE hr.employees
```

Practice 11-1: Using Flashback Data Archive (continued)

```
  2  SET salary = salary + 1000
  3  WHERE last_name = 'Fox'
  4  /

1 row updated.

SQL> COMMIT
  2  /

Commit complete.

SQL> UPDATE hr.employees
  2  SET salary = salary + 1000
  3  WHERE last_name = 'Fox'
  4  /

1 row updated.

SQL> COMMIT
  2  /

Commit complete.

SQL> UPDATE hr.employees
  2  SET    salary = salary + 1000
  3  WHERE  last_name = 'Fox'
  4  /

1 row updated.

SQL> COMMIT
  2  /

Commit complete.

SQL> pause Press [Enter] to continue...
Press [Enter] to continue...

SQL>
SQL> /*== Query the up-to-date value for Mr. Fox ==*/
SQL>
SQL> SELECT employee_id, last_name, salary
  2  FROM   hr.employees
  3  WHERE  last_name = 'Fox'
  4  /
Press [Enter] to continue...

EMPLOYEE_ID LAST_NAME                     SALARY
----------- ------------------------- ----------
        170 Fox                            12600

1 row selected.
```

Practice 11-1: Using Flashback Data Archive (continued)

```
SQL>
SQL> pause  Press [Enter] to continue...
Press [Enter] to continue...

SQL>
SQL> set pause off
SQL>
```

8) To query the internal name of the archive table, execute the following command:

`SELECT * FROM USER_FLASHBACK_ARCHIVE_TABLES;`

```
SQL> SELECT * FROM USER_FLASHBACK_ARCHIVE_TABLES;

TABLE_NAME                      OWNER_NAME
------------------------------ ------------------------------
FLASHBACK_ARCHIVE_NAME
--------------------------------------------------------------
ARCHIVE_TABLE_NAME
-----------------------------------------------------
EMPLOYEES                       HR
FLA1

SYS_FBA_HIST_73137

1 row selected.

SQL>
```

9) To ensure that the following queries use the archive tables, and not the undo, switch the undo tablespaces and drop the old one. As a `SYS` user, assign **`UNDOTBS2`** as the current undo tablespace, by using Enterprise Manager > Server> Automatic Undo Management > "Change Tablespace" button, selecting the `UNDOTBS2` tablespace, and clicking OK, or by executing the following commands:

`connect / as sysdba`

`ALTER SYSTEM SET UNDO_TABLESPACE=UNDOTBS2;`

`DROP TABLESPACE UNDOTBS1;`

```
SQL> connect / as sysdba
Connected.

SQL> ALTER SYSTEM SET UNDO_TABLESPACE=UNDOTBS2;
System altered.

SQL> DROP TABLESPACE UNDOTBS1;
Tablespace dropped.
```

Note: If you execute the `DROP` command too quickly, while internal processing still occurs, you might receive an ORA-30013 error. Wait several minutes and try again.

Practice 11-1: Using Flashback Data Archive (continued)

10) As the HR user, choose a time after the creation of the flashback data archive and before you executed the erroneous DML. To view Mr. Fox's employee record as of that time, execute the following query (replace *'15' MINUTE* with your chosen historic date, format examples: '50' SECOND, '10' DAY, '5' MONTH):
Note: You receive an ORA-1466 error, if the time you specify a time before the flashback data archive was started. Reduce the time to a smaller interval and try again. If you still see the salary of 12600, increase your time interval.

```
SELECT employee_id, last_name, salary
FROM   hr.employees AS OF TIMESTAMP
       (SYSTIMESTAMP - INTERVAL '15' MINUTE)
WHERE  last_name = 'Fox';
```

Entering the command is recommended for better retention and selecting the right time interval, but if typing is a difficulty, you can also execute the `flada_sel.sql` script.

```
SQL> connect hr
Ener password: oracle_4U  <<< not displayed
Connected.
SQL> SELECT employee_id, last_name, salary
  2  FROM hr.employees AS OF TIMESTAMP
  3     (SYSTIMESTAMP - INTERVAL '15' MINUTE)
  4     WHERE last_name = 'Fox';

EMPLOYEE_ID LAST_NAME                     SALARY
----------- ------------------------- ----------
        170 Fox                             9600
1 row selected.
SQL>
```

11) As the HR user, you realize that the recent updates were mistakes. To revert to the original values for your chosen historic date (for example, ten minutes ago), execute the following command (replace *'15' MINUTE* with your chosen historic date):

```
UPDATE hr.employees
SET salary = (SELECT salary FROM hr.employees
    AS OF TIMESTAMP (SYSTIMESTAMP - INTERVAL '15' MINUTE)
    WHERE last_name = 'Fox')
WHERE last_name = 'Fox';
```

Entering the command is recommended for better retention and selecting the right time interval, but if typing is a difficulty, you can also execute the `flada_upd.sql` script.

```
SQL> UPDATE hr.employees
  2  SET salary = (SELECT salary FROM hr.employees
  3     AS OF TIMESTAMP (SYSTIMESTAMP - INTERVAL '15' MINUTE)
  4     WHERE last_name = 'Fox')
```

```
  5  WHERE last_name = 'Fox';

1 row updated.

SQL>
```

12) From your SQL*Plus session, connect to the database as the `SYS` user and list the data dictionary views available to you. Execute the `flada_list1.sql` file:

```
SQL> connect / as sysdba
Connected.
SQL> @flada_list1
SQL> REM "******************************************** "
SQL> REM "For demo purposes ONLY:"
SQL>
SQL> connect / as sysdba
Connected.
SQL>
SQL> set echo on
SQL> set serveroutput on
SQL> -- set verify on
SQL> set term on
SQL> set lines 200
SQL> set pages 44
SQL> set pause on pause "Press [Enter] to continue ..."
SQL>
SQL> /*== To list the available data dictioary views ==*/
SQL>
SQL> SELECT table_name
  2  FROM   dict
  3  WHERE  table_name LIKE '%FLASHBACK_ARCHIVE%'
  4  /
Press [Enter] to continue ...

TABLE_NAME
------------------------------
DBA_FLASHBACK_ARCHIVE
DBA_FLASHBACK_ARCHIVE_TABLES
DBA_FLASHBACK_ARCHIVE_TS
USER_FLASHBACK_ARCHIVE
USER_FLASHBACK_ARCHIVE_TABLES

SQL> pause Press [Enter] to continue ...
Press [Enter] to continue ...

SQL>
SQL> col FLASHBACK_ARCHIVE_NAME format A25
SQL> col ARCHIVE_TABLE_NAME format A20
SQL> col TABLE_NAME format A12
SQL> col OWNER_NAME format A10
SQL>
SQL> DESC dba_flashback_archive
```

```
 Name
Null?    Type
 -------------------------------------------------------------
---------------------------------------------------- --------
-------------------------------------------------------------
--------------
 OWNER_NAME
VARCHAR2(30)
 FLASHBACK_ARCHIVE_NAME
NOT NULL VARCHAR2(255)
 FLASHBACK_ARCHIVE#
NOT NULL NUMBER
 RETENTION_IN_DAYS
NOT NULL NUMBER
 CREATE_TIME
TIMESTAMP(9)
 LAST_PURGE_TIME
TIMESTAMP(9)
 STATUS
VARCHAR2(7)

SQL> pause Press [Enter] to continue ...
Press [Enter] to continue ...

SQL>
SQL> /*== To query the time when the flashback data archive(s)
have been created ==*/
SQL>
SQL> SELECT flashback_archive_name, create_time, status
  2  FROM   dba_flashback_archive
  3  /
Press [Enter] to continue ...

FLASHBACK_ARCHIVE_NAME    CREATE_TIME
STATUS
------------------------- -------------------------------------
-------------------------------------- -------
FLA1                      05-JUN-09 11.06.09.000000000 PM

SQL> pause Press [Enter] to continue ...
Press [Enter] to continue ...

SQL>
SQL> DESC dba_flashback_archive_ts
 Name
Null?    Type
 -------------------------------------------------------------
---------------------------------------------------- --------
-------------------------------------------------------------
--------------
 FLASHBACK_ARCHIVE_NAME
NOT NULL VARCHAR2(255)
```

Practice 11-1: Using Flashback Data Archive (continued)

```
 FLASHBACK_ARCHIVE#
NOT NULL NUMBER
 TABLESPACE_NAME
NOT NULL VARCHAR2(30)
 QUOTA_IN_MB
VARCHAR2(40)
```

```
SQL> pause Press [Enter] to continue ...
Press [Enter] to continue ...

SQL>
SQL> /*== To list the tablespace(s), which are used for
flashback data archives ==*/
SQL>
SQL> SELECT *
  2  FROM   dba_flashback_archive_ts
  3  /
Press [Enter] to continue ...

FLASHBACK_ARCHIVE_NAME    FLASHBACK_ARCHIVE# TABLESPACE_NAME
QUOTA_IN_MB
------------------------ ----------------- -----------------
------------- ----------------------------------------
FLA1                                     1 FLA_TBS1
10

SQL> pause Press [Enter] to continue ...
Press [Enter] to continue ...

SQL>
SQL> DESC dba_flashback_archive_tables
 Name
Null?    Type
 -------------------------------------------------------------
---------------------------------------------------- --------
-------------------------------------------------------------
--------------
 TABLE_NAME
NOT NULL VARCHAR2(30)
 OWNER_NAME
NOT NULL VARCHAR2(30)
 FLASHBACK_ARCHIVE_NAME
NOT NULL VARCHAR2(255)
 ARCHIVE_TABLE_NAME
VARCHAR2(53)
 STATUS
VARCHAR2(8)

SQL> pause Press [Enter] to continue ...
Press [Enter] to continue ...
```

```
SQL>
SQL> /*== Query the table name(s), the owner name(s), and ==*/
SQL> /*== the internal "history" table name(s) of the
flashback data archive ==*/
SQL>
SQL> SELECT *
  2  FROM   dba_flashback_archive_tables
  3  /
Press [Enter] to continue ...
```

```
TABLE_NAME  OWNER_NAME FLASHBACK_ARCHIVE_NAME ARCHIVE_TABLE_NAME
----------- ---------- ---------------------- ------------------
EMPLOYEES   HR         FLA1                   SYS_FBA_HIST_73137

SQL>
SQL> pause Press [Enter] to continue ...
Press [Enter] to continue ...

SQL> clear columns
SQL>
```

13) As the HR user (with the `oracle_4U` password), list the `FLASHBACK_ARCHIVE` data dictionary tables, which are accessible to you. You can execute the `flada_list2` script, if you prefer to not enter the commands directly.

```
SQL> connect hr
Enter password: oracle_4U <<< not dislplayed
Connected.
SQL> @flada_list2

SQL> SELECT table_name
  2  FROM   dict
  3  WHERE  table_name LIKE '%FLASHBACK_ARCHIVE%'
  4  /
Press [Enter] to continue ...

TABLE_NAME
------------------------------
USER_FLASHBACK_ARCHIVE
USER_FLASHBACK_ARCHIVE_TABLES

SQL> pause Press [Enter] to continue ...
Press [Enter] to continue ...

SQL> exit
$
```

14) To practice additional flashback data archive maintenance tasks, perform the following steps: Log into SQL*Plus as the `ARCHIVE_ADMIN` user.

```
$ sqlplus ARCHIVE_ADMIN
Enter password: oracle_4U <<< not displayed
```

Practice 11-1: Using Flashback Data Archive (continued)

```
SQL>
```

15) Data in the flashback data archive is automatically purged when the retention time expires. However, you can also purge data explicitly with the following command to purge data older than two minutes:

```
ALTER FLASHBACK ARCHIVE fla1 PURGE BEFORE TIMESTAMP
(SYSTIMESTAMP - INTERVAL '2' MINUTE);
```

```
SQL> ALTER FLASHBACK ARCHIVE fla1 PURGE BEFORE
TIMESTAMP(SYSTIMESTAMP - INTERVAL '2' MINUTE);

Flashback archive altered.

SQL>
```

16) Execute the `flada_tbs2.sql` script to create an additional 10 MB tablespace as the SYS user.

```
SQL> @flada_tbs2
Connected.
SQL> set serveroutput on
SQL> -- set verify on
SQL> set term on
SQL> set lines 200
SQL> set pages 44
SQL> set pause on
SQL>
SQL> /*== Create another tablespace ==*/
SQL>
SQL> DROP TABLESPACE fla_tbs2 INCLUDING CONTENTS
  2  /
DROP TABLESPACE fla_tbs2 INCLUDING CONTENTS
*
ERROR at line 1:
ORA-00959: tablespace 'FLA_TBS2' does not exist


SQL>
SQL> CREATE SMALLFILE TABLESPACE fla_tbs2
  2  DATAFILE '$HOME/BACKUP/fla_tbs02.dbf'
  3  SIZE 10M REUSE AUTOEXTEND ON NEXT 640K MAXSIZE 32767M
  4  NOLOGGING EXTENT MANAGEMENT LOCAL SEGMENT SPACE
MANAGEMENT AUTO
  5  /

Tablespace created.

SQL> pause Press [Enter] to continue...
Press [Enter] to continue...

SQL>
```

Practice 11-1: Using Flashback Data Archive (continued)

17) As the ARCHIVE_ADMIN user, add 5 MB of the FLA_TBS2 tablespace to the FLA1 flashback data archive.

```
SQL> connect ARCHIVE_ADMIN
Enter password: oracle_4U <<< not displayed
Connected.
SQL> ALTER FLASHBACK ARCHIVE fla1 ADD TABLESPACE fla_tbs2
QUOTA 5M;

Flashback archive altered.

SQL>
```

18) All tables in one specific flashback data archive have the same retention time. Change the retention time of the FLA1 flashback data archive to two years.

```
SQL> ALTER FLASHBACK ARCHIVE fla1 MODIFY RETENTION 2 YEAR;

Flashback archive altered.

SQL>
```

19) As the ARCHIVE_ADMIN user, drop the FLA1 flashback data archive.
Note: Dropping a flashback data archive includes dropping the internal tamper-proofed history table. You cannot drop this table directly due to auditing and security requirements. Dropping a flashback data archive does not drop the tablespaces in which they are stored, because the tablespaces might contain other data.

```
SQL> DROP FLASHBACK ARCHIVE fla1;

Flashback archive dropped.
SQL>
```

20) Wait a minute or two and connect as SYS user; clean up your environment by executing the flada_cleanup.sql script.

```
SQL> @flada_cleanup
SQL> REM Execute this script script for training purposes only
SQL> REM Undo Cleanup for Flashback Data Archive
SQL> REM Execute script as SYSDBA
SQL>
SQL> set echo on
SQL> set serveroutput on
SQL> set term on
SQL> set lines 200
SQL> set pause on
SQL>
SQL> connect / as sysdba
Connected.
SQL> set echo off

Tablespace created.
```

Practice 11-1: Using Flashback Data Archive (continued)

```
SQL>
SQL> ALTER SYSTEM SET UNDO_TABLESPACE=undotbs1
  2  /

System altered.

SQL> DROP TABLESPACE fla_tbs1 INCLUDING CONTENTS
  2  /

Tablespace dropped.

SQL> DROP TABLESPACE fla_tbs2 INCLUDING CONTENTS
  2  /

Tablespace dropped.

SQL>
SQL> DROP TABLESPACE undotbs2 INCLUDING CONTENTS
  2  /

Tablespace dropped.

SQL> host rm -f $HOME/BACKUP/fla_tbs01.dbf

SQL> host rm -f $HOME/BACKUP/fla_tbs02.dbf

SQL> host rm -f $HOME/BACKUP/undotbs02.dbf

SQL> prompt Flashback Data Archive cleanup complete.
Flashback Data Archive cleanup complete.
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...

SQL> exit
```

Note: If you execute the `DROP` command too quickly while internal processing is still occurring, you might receive an ORA-30013 error. Wait and try again.

Practice 11-2: Using the Recycle Bin

In this practice you use the recycle bin to restore dropped objects. You can do this with SQL commands or with Enterprise Manager. *For the former, perform steps 1 through 4. For the latter, perform step 1, and then steps 5 through 11.*

1) This workshop scenario simulates loss of data. Point to the `orcl` instance. To introduce the problem, change directory to `$HOME/labs` and run the `recyclebin_lab.sh` as shown:

```
$ cd $HOME/labs
$. oraenv
ORACLE_SID = [orcl] ? orcl

$ ./recyclebin_lab.sh
SQL> SQL>
Table dropped.

SQL> SQL> select * from hr.departments
                  *
ERROR at line 1:
ORA-00942: table or view does not exist


SQL> select * from hr.departments
                  *
ERROR at line 1:
ORA-00942: table or view does not exist


SQL> select * from hr.departments
                  *
ERROR at line 1:
ORA-00942: table or view does not exist
```

2) The lab script drops the `HR.DEPARTMENTS` table including indexes, restraints, and so on. Query the `DBA_RECYCLE_BIN` view to confirm this.

```
$ sqlplus / as sysdba
SQL> COL OWNER FORMAT A5
SQL> SELECT owner, original_name, droptime
FROM dba_recyclebin WHERE owner = 'HR';

OWNER ORIGINAL_NAME                    DROPTIME
----- -------------------------------- -------------------
HR    DEPT_ID_PK                       2009-06-06:15:31:05
HR    DEPARTMENTS                      2009-06-06:15:31:05
HR    DEPT_LOCATION_IX                 2009-06-06:15:31:04
```

3) Use Flashback Table to restore the dropped objects. You can use Flashback Table through Enterprise Manager or through SQL*Plus. To flash back the table using Enterprise Manager, go to step 5.

Practice 11-2: Using the Recycle Bin (continued)

```
SQL> FLASHBACK TABLE hr.departments TO BEFORE DROP;

Flashback complete.
SQL>
```

4) Perform a select operation against the `HR.DEPARTMENTS` table to confirm the success of the Flashback Table operation, then exit

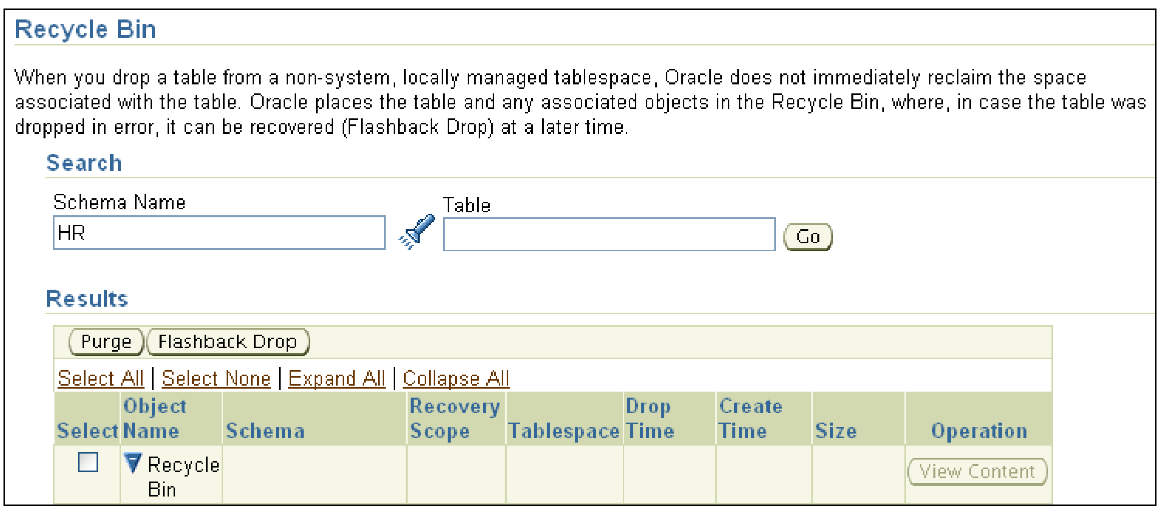
```
SQL> SELECT * FROM hr.departments;

DEPARTMENT_ID DEPARTMENT_NAME                MANAGER_ID LOCATION_ID
------------- ------------------------------ ---------- -----------
           10 Administration                        200        1700
           20 Marketing                             201        1800
           30 Purchasing                            114        1700
 . . .
SQL> exit
```

5) The `HR.DEPARTMENTS` table can also be flashed back using Enterprise Manager.

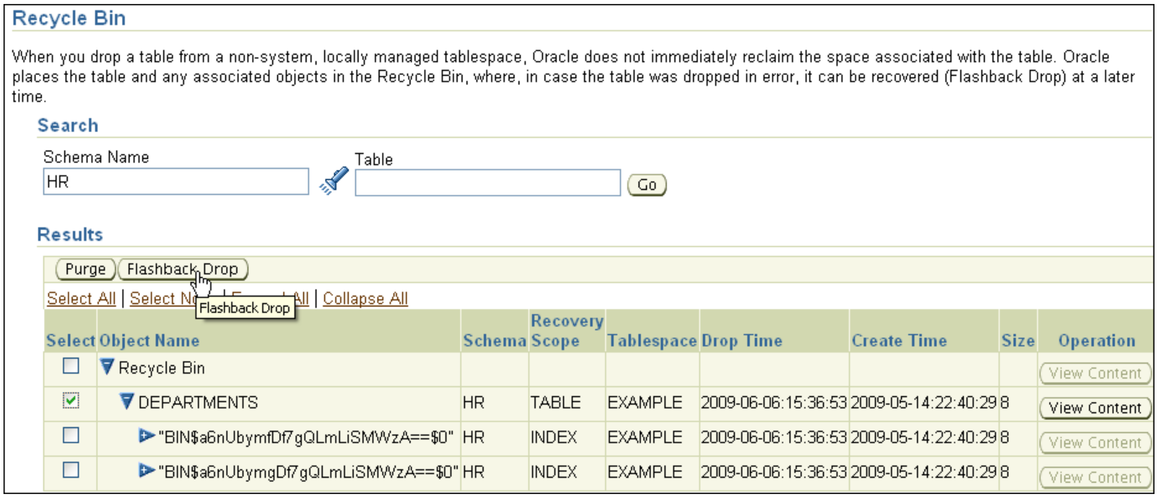
6) Log in as the `SYS` user with the `oracle_4U` password and connect as `SYSDBA`.

7) In EM, navigate to Schema > Tables.

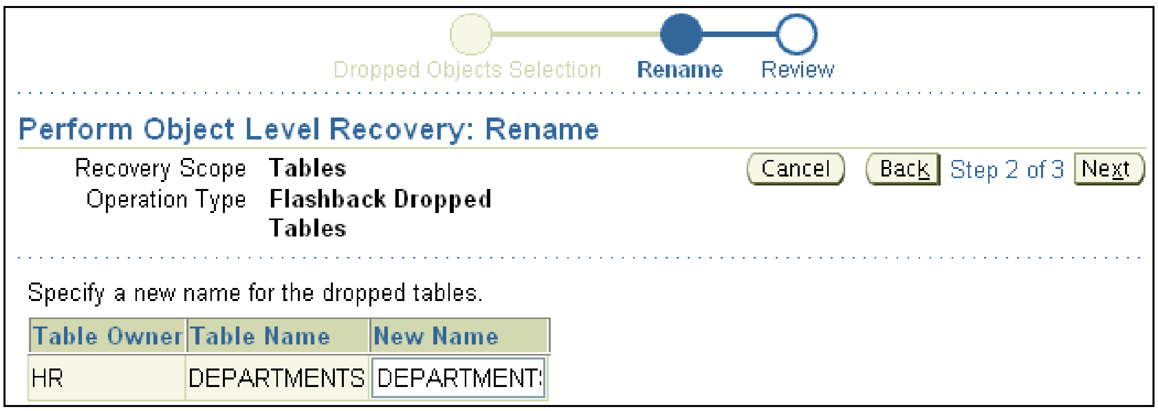
8) Click the Recycle Bin button, enter `HR` in the Schema Name field and click Go.



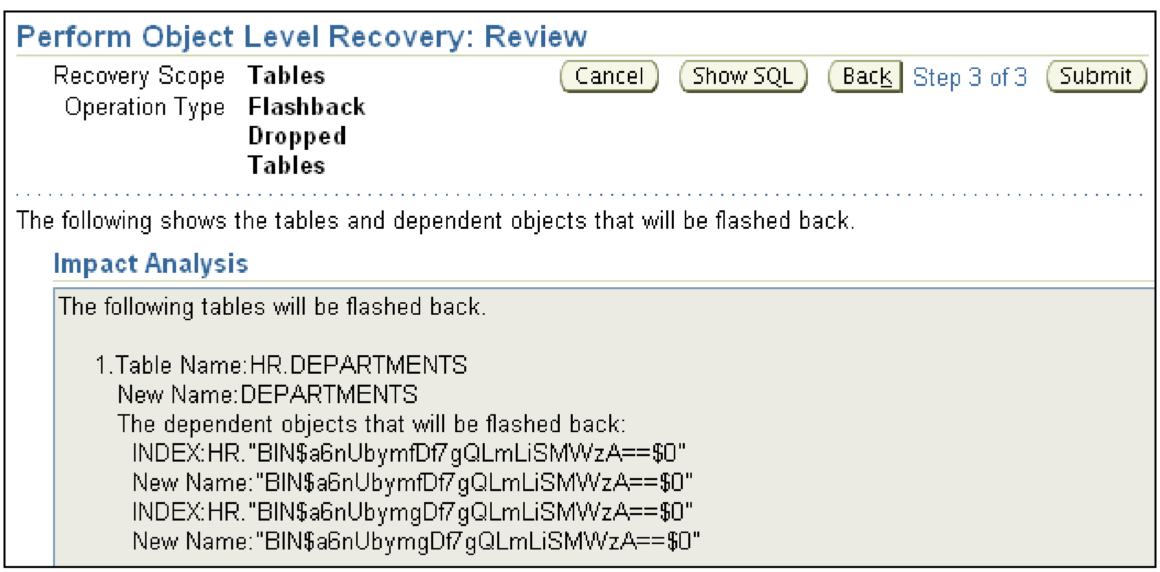
9) Click the + icon before `DEPARTMENTS`, to see the dependant objects.



10) Select DEPARTMENTS from the list and click the Flashback Drop button.

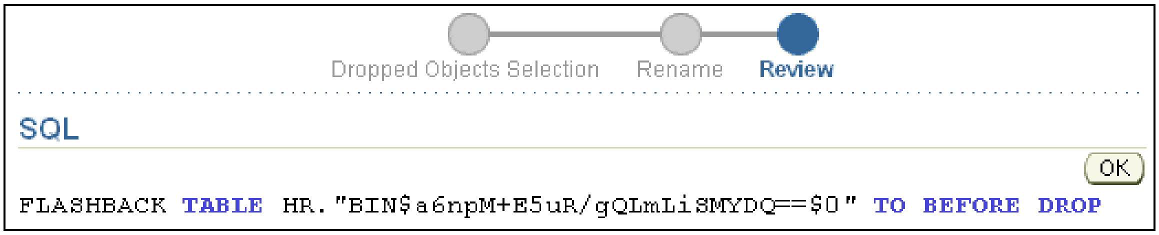


11) The Perform Object Level Recovery: Rename page allows you to rename the restored object. Accept the default or original name and click Next.

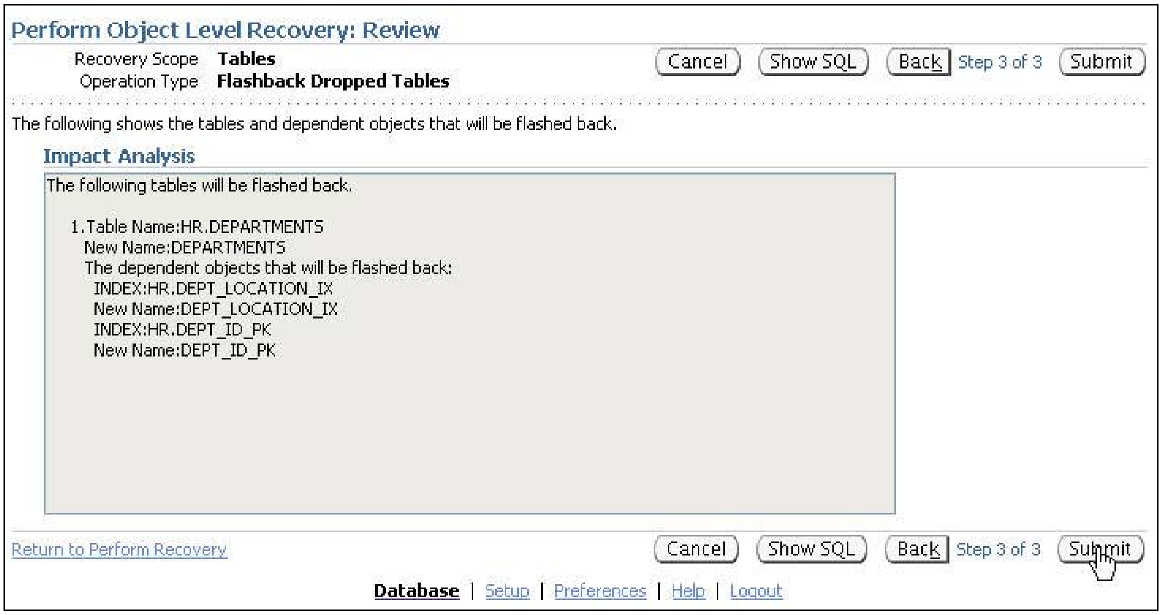


12) On the Review page, inspect the Impact Analysis. Note that the table and dependent objects are to be restored and note the names they will be restored to.

Practice 11-2: Using the Recycle Bin (continued)



13) Optionally, click the Show SQL button, review the SQL and click OK.



14) Click Submit.

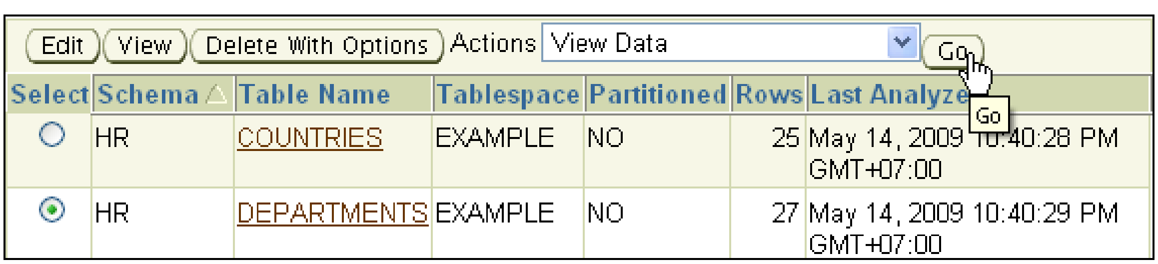
15) Click OK on the Confirmation page.

16) To view table data in EM:

a) Click the Tables breadcrumb.

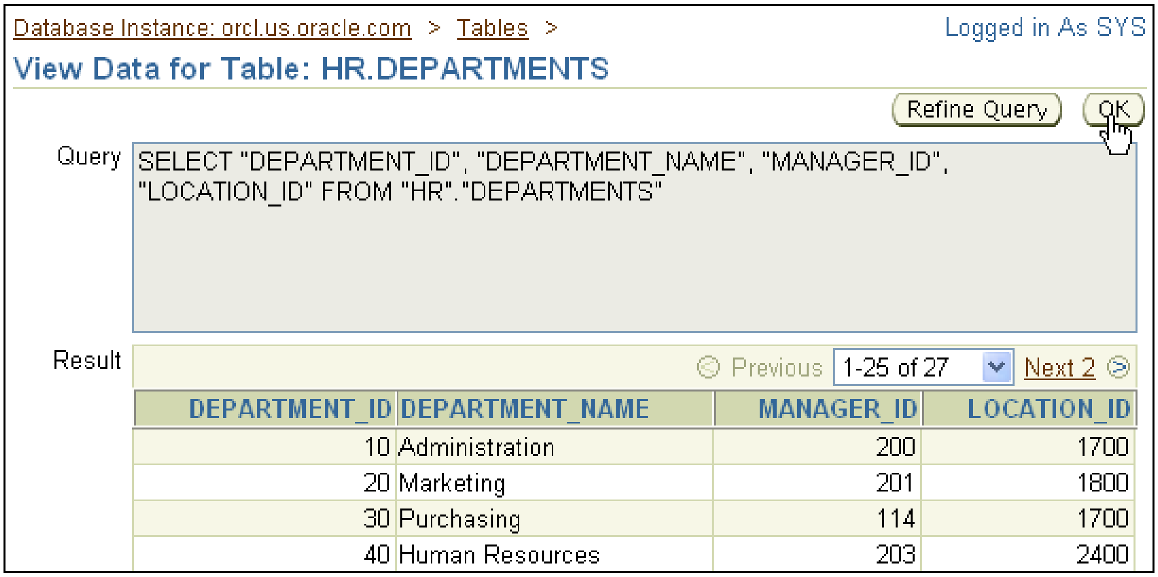
b) Enter `HR` as Schema and click Go.

c) Select the `DEPARTMENTS` table, then the View Data Actions, and click Go to execute the action.



Practice 11-2: Using the Recycle Bin (continued)

17) After confirming that the data are restored, click OK. Then exit EM.



DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400

Practices for Lesson 12

In this lesson you learn to flash back an entire database.

Background: In this scenario, the HR reorganization job runs prematurely, and you must undo its changes. The changes are such that you are not sure what tables are involved. So, you decide (now that flashback logging has been enabled) to use Flashback Database instead of performing a recovery.

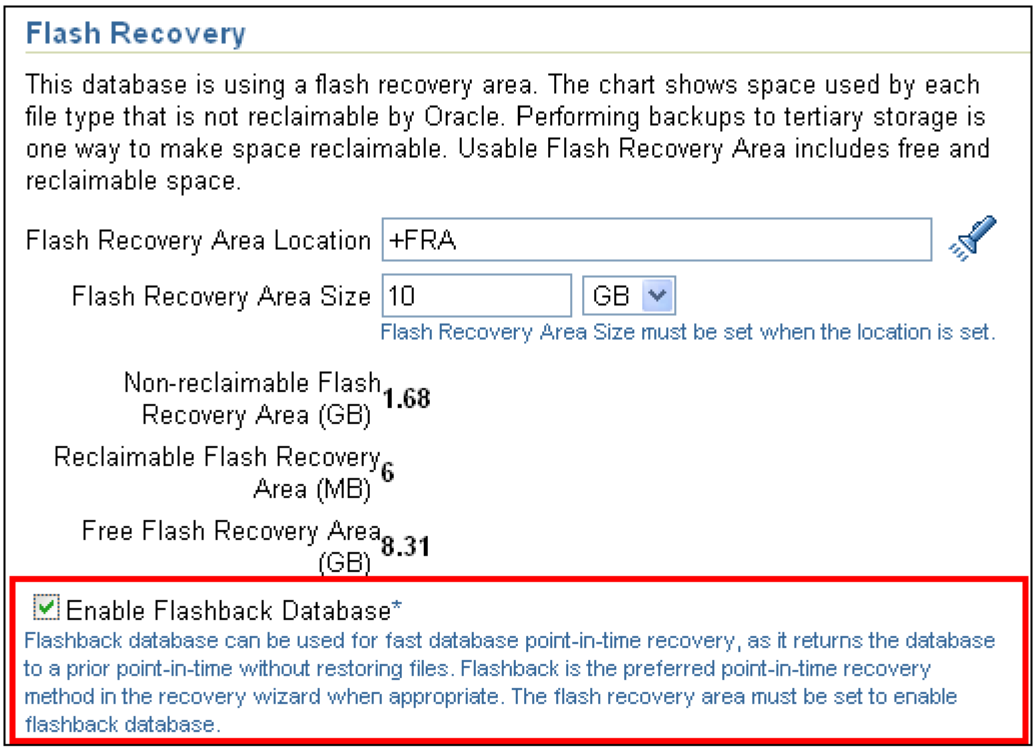
Practice 12-1: Flashback Database

Flash back an entire database. Unless specified otherwise, you should log in as the SYS user as SYSDBA through SQL*Plus or Database Control.

1) Using Enterprise Manager, turn on flashback logging for your database (enable Flashback Database).

   a) In Database Control, click the Availability tab, and then the Recovery Settings link in the Backup/Recovery Setup region.

   b) Scroll down and select the Enable Flashback Database check box. Click Apply. On the Confirmation page, click Yes to proceed with the database shutdown.

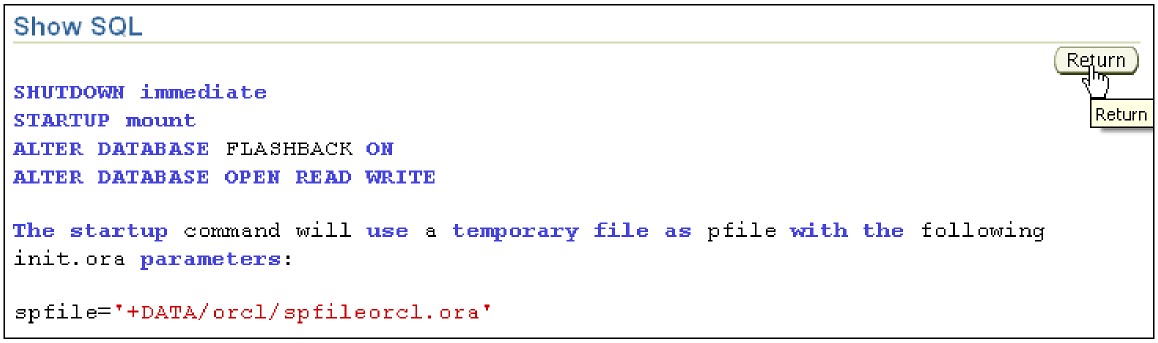


   c) On the Confirmation page, click Yes to restart your database.

   d) On the Restart Database: Specify Host and Target Database Credentials, ensure that `oracle` is entered as username and as password. Then click OK.

   e) Optionally, on the Restart Database: Confirmation page, click Show SQL, review the SQL and click Return.

Practice 12-1: Flashback Database (continued)



f) Click Yes on the Restart Database: Confirmation page.

g) Click Refresh and wait for the database to restart. (If you receive an internal error, just cllick OK and than again Refresh.) - You may need to click Refresh several times.

2) After the database has been restarted with flashback logging enabled, note the lowest SCN of the database

a) On the Database home page, click the Flashback Time link in the High Availability section. [45]

b) On the Recovery Settings page, at the bottom of the Flash Recovery section, note the Lowest SCN [[46r]] (1136492 in this example).

3) Alternatively, you can view and note the current SCN of the database by querying the `V$DATABASE` view.

Note: *You will need the SCN later.*

SCN: ________________

a) Enter the following:

```
$ cd $HOME/labs
$ sqlplus / as sysdba
SQL> SELECT current_scn FROM v$database;

CURRENT_SCN
-----------
    1137367
```

4) Note the sum of the salaries in the `HR.EMPLOYEES` table and the count of the rows in the `JOB_HISTORY` table.

```
SQL> SELECT SUM(salary) FROM hr.employees;

SUM(SALARY)
-----------
```

Practice 12-1: Flashback Database (continued)

```
     691416


SQL> SELECT COUNT(*) FROM hr.job_history;

  COUNT(*)
----------
        10
```

5) To perform some HR reorganization updates, .

   a) Connect as `HR` user with the `oracle_4U` password.

   b) Execute the `flb_db_txn.sql` script.

```
SQL> connect hr
Enter password: oracle_4U <<< not displayed
Connected.
SQL> @flb_db_txn.sql
SQL>
SQL> update employees set department_id = 90 where job_id =
'IT_PROG';

5 rows updated.

SQL>
SQL> update employees e set salary = least(e.salary,(select
(min_salary + max_salary)/2 * 1.10 from jobs j where j.job_id
= e.job_id)) where job_id not like 'AD_%';

103 rows updated.

SQL>
SQL> commit;

Commit complete.

SQL>
```

6) As the `SYS` user, note the current SCN in the database again, and also the salary sum and the `JOB_HISTORY` row count.

   Note that these values are different from what was queried in steps 2 and 3.

```
SQL> connect / as sysdba
Connected.
SQL> SELECT current_scn FROM v$database;

CURRENT_SCN
-----------
```

```
    1137586

SQL> SELECT SUM(salary) FROM hr.employees;

SUM(SALARY)
-----------
   679092.4

SQL> SELECT COUNT(*) FROM hr.job_history;

  COUNT(*)
----------
        15

SQL>
```

7) Using RMAN, flash back the database to the first SCN value noted in step 2 (1137367 in this example).

   a) Shut down and then mount the database by entering the following:

```
SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> STARTUP MOUNT
ORACLE instance started.

Total System Global Area  481259520 bytes
Fixed Size                  1337324 bytes
Variable Size             385878036 bytes
Database Buffers           88080384 bytes
Redo Buffers                5963776 bytes
Database mounted.
SQL> exit
```

   b) If you have any RMAN open, close them. Then log in to RMAN again.

```
$ rman target / nocatalog
connected to target database: ORCL (DBID=1220535480, not open)
using target database control file instead of recovery catalog

RMAN>
```

   c) At the RMAN prompt, enter the FLASHBACK DATABASE command, and supply the SCN number recorded in step 2:

```
RMAN> FLASHBACK DATABASE TO SCN=1137367;

Starting flashback at 2009-07-23:21:05:29
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=27 device type=DISK
```

```
allocated channel: ORA_SBT_TAPE_1
channel ORA_SBT_TAPE_1: SID=28 device type=SBT_TAPE
channel ORA_SBT_TAPE_1: WARNING: Oracle Test Disk API


starting media recovery
media recovery complete, elapsed time: 00:00:03

Finished flashback at 2009-07-23:21:05:34

RMAN>
```

8) Before opening the database for read and write operations, verify that the database was flashed back to the correct SCN by looking at the contents of the tables and seeing that they are back to what was noted in step 3.

   a) Logged into SQL*Plus as SYSDBA, enter the following coomand to open the database in read-only mode:

```
SQL> alter database open read only;
Database altered.
```

   b) Enter the following at the SQL prompt to determine the salary sum for the `EMPLOYEES` table.

```
SQL> SELECT SUM(salary) FROM hr.employees;

SUM(SALARY)
-----------
     691416
```

9) Open the database for read and write opeations. You have to use the `RESETLOGS` keyword.

   a) Shut down the database:

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
```

   b) Start up the database in the `MOUNT` state:

```
SQL> startup mount
ORACLE instance started.

Total System Global Area  481259520 bytes
Fixed Size                  1337324 bytes
Variable Size             385878036 bytes
Database Buffers           88080384 bytes
```

Practice 12-1: Flashback Database (continued)

```
Redo Buffers                  5963776 bytes
Database mounted.
SQL>
```

c) Open the database in read/write mode with resetlogs:

```
SQL> alter database open resetlogs;

Database altered.
```

(*The command might take a while to execute.*)

10) At this point you can familiarize yourself with the flashback-related V$ views. To see time values, alter your session to display hours, minutes, and seconds with any date values.

```
SQL> ALTER SESSION SET NLS_DATE_FORMAT="yyyy-mm-dd
hh24:mi:ss";
```

11) Query the `V$FLASHBACK_DATABASE_LOG` view and determine the lowest SCN that the database can be flashed back to. Record your answer here: _______________

```
SQL> set wrap off
SQL> select * from v$FLASHBACK_DATABASE_LOG;

OLDEST_FLASHBACK_SCN OLDEST_FLASHBACK_TI RETENTION_TARGET
FLASHBACK_SIZE
-------------------- ------------------- ---------------- ----
----------
             1136492 2009-07-23 20:47:17             1440
16384000

SQL>
```

12) View the overhead associated with flashback logging and related operations by querying `V$FLASHBACK_DATABASE_STAT`. What is the average number of bytes of flashback data written per minute during this time interval?

```
SQL> select * from V$FLASHBACK_DATABASE_STAT;

truncating (as requested) before column
ESTIMATED_FLASHBACK_SIZE

OLDEST_FLASHBACK_SCN OLDEST_FLASHBACK_TI RETENTION_TARGET
FLASHBACK_SIZE
-------------------- ------------------- ---------------- ----
----------
             1136492 2009-07-23 20:47:17             1440
16384000

SQL> select * from V$FLASHBACK_DATABASE_STAT;
```

Practice 12-1: Flashback Database (continued)

```
truncating (as requested) before column
ESTIMATED_FLASHBACK_SIZE


BEGIN_TIME          END_TIME            FLASHBACK_DATA
DB_DATA  REDO_DATA
------------------- ------------------- -------------- -------
--- ----------
2009-07-23 21:07:56 2009-07-23 21:12:13        4538368
0    1372160
SQL>
```

In the example above, the answer is `4538368 / (21:07 – 21:12)` which is `907673` bytes per minute. Your numbers may vary slightly.

13) Determine the current size of stored flashback data by querying `V$FLASHBACK_DATABASE_LOG`. Record your answer here: ____________.

```
SQL> SELECT flashback_size FROM V$FLASHBACK_DATABASE_LOG;

FLASHBACK_SIZE
--------------
      24576000
```

Note: Your results will probably vary slightly from those shown here.

14) To turn off the flashback database functionality, execute the following command.

```
SQL> ALTER DATABASE FLASHBACK OFF;
Database altered.
SQL> exit
```

Note: The command might take a couple of minutes to execute.

Practices for Lesson 13

In this practice, you use Automatic Memory Management to show that you no longer need to manually modify `SGA_TARGET` and `PGA_AGGREGATE_TARGET`. You observe the memory distribution when you run an expensive parallel query that consumes a lot of SGA memory.

Practice 13-1: Using Automatic Memory Management

In this practice, you use the various new memory management capabilities.

1) For your `orcl` database, navigate into the `labs` directory, turn off archiving, and stop Enterprise Manager Database Control.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
$ cd ~/labs
$ ./orcl_to_noarchivelog.sh
Database closed.
Database dismounted.
ORACLE instance shut down.
ORACLE instance started.

Total System Global Area  481259520 bytes
Fixed Size                  1337352 bytes
Variable Size             373295096 bytes
Database Buffers          100663296 bytes
Redo Buffers                5963776 bytes
Database mounted.

Database altered.

Database altered.

$ emctl stop dbconsole
Oracle Enterprise Manager 11g Database Control Release
11.2.0.0.2
Copyright (c) 1996, 2009 Oracle Corporation.  All rights
reserved.
https://edrsr37p1.us.oracle.com:1158/em/console/aboutApplicati
on
Stopping Oracle Enterprise Manager 11g Database Control ...
 ...  Stopped.
$
```

2) **Make a copy of your SPFILE**, logged in to SQL*Plus for the `orcl` instance as the SYS user with the `oracle_4U` password.

```
$ sqlplus / as sysdba
SQL> CREATE PFILE='/tmp/initorcl.ora.bak' FROM SPFILE;

File created.
SQL>
```

Practice 13-1: Using Automatic Memory Management (continued)

3) Still connected as the SYS user in SQL*Plus, set the following parameters to the given value in your SPFILE only! Use the `amm_parameters.sql` file located in your `/home/oracle/labs` directory.

```
_PX_use_large_pool = TRUE
_memory_broker_stat_interval = 5
_memory_management_tracing = 31
parallel_execution_message_size = 36864
parallel_max_servers = 200
parallel_adaptive_multi_user = FALSE
processes = 200
sga_target = 0
pga_aggregate_target = 0
memory_target = 300M
```

```
SQL> @amm_parameters
SQL> alter system set "_PX_use_large_pool" = TRUE
SCOPE=SPFILE;

System altered.

SQL> alter system set "_memory_broker_stat_interval" = 5
SCOPE=SPFILE;

System altered.

SQL> alter system set "_memory_management_tracing" = 31
SCOPE=SPFILE;

System altered.

SQL> alter system set "parallel_execution_message_size" =
36864 SCOPE=SPFILE;

System altered.

SQL> alter system set "parallel_max_servers" = 200
SCOPE=SPFILE;

System altered.

SQL> alter system set "parallel_adaptive_multi_user" = FALSE
SCOPE=SPFILE;

System altered.

SQL> alter system set "processes" = 200 SCOPE=SPFILE;

System altered.

SQL> alter system set "pga_aggregate_target" = 0 SCOPE=SPFILE;
```

Practice 13-1: Using Automatic Memory Management (continued)

```
System altered.

SQL> alter system set "sga_target" = 0 SCOPE=SPFILE;

System altered.

SQL> alter system set "memory_target" = 300M SCOPE=SPFILE;

System altered.

SQL>
```

4) Execute the `amm_setup.sql` script.

   a) Drop and recreate the `TBSSGA` and `MYTEMP` tablespaces and the `AMM` DBA user for whom they are defaults. Then press Enter to continue.

```
SQL> @amm_setup.sql
SQL> REM "************************** "
SQL> REM "For training purposes ONLY, execute as the oracle OS
user
SQL>
SQL> set echo on
SQL> set serveroutput on
SQL> set term on
SQL> set lines 200
SQL> set pages 44
SQL> set pause on pause "Press [Enter] to continue..."
SQL>

SQL> drop tablespace tbssga including contents and datafiles;
drop tablespace tbssga including contents and datafiles
*
ERROR at line 1:
ORA-00959: tablespace 'TBSSGA' does not exist


SQL>
SQL> create tablespace tbssga datafile '+DATA' size 20m;

Tablespace created.

SQL>
SQL> drop tablespace mytemp including contents and datafiles;
drop tablespace mytemp including contents and datafiles
*
ERROR at line 1:
ORA-00959: tablespace 'MYTEMP' does not exist
```

Practice 13-1: Using Automatic Memory Management (continued)

```
SQL>
SQL> create temporary tablespace mytemp tempfile '+DATA' size
40m reuse;

Tablespace created.
SQL>
SQL> drop user amm cascade;
drop user amm cascade
          *
ERROR at line 1:
ORA-01918: user 'AMM' does not exist


SQL>
SQL> create user amm identified by “oracle_4U’
  2  default tablespace tbssga
  3  temporary tablespace mytemp;

User created.

SQL>
SQL> grant connect,resource,dba to amm;

Grant succeeded.
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```

b) To view the current memory components, query the `V$MEMORY_DYNAMIC_COMPONENTS` view by pressing Enter to continue the script.

```
SQL>
SQL>
SQL> SELECT substr(COMPONENT, 0, 10) COMP, CURRENT_SIZE CS,
USER_SPECIFIED_SIZE US
  2    FROM v$memory_dynamic_components
  3   WHERE CURRENT_SIZE!=0;
Press [Enter] to continue...
```

c) View the query result by pressing Enter to continue the script.

```
SQL> column COMP format a10
SQL>
SQL> SELECT substr(COMPONENT, 0, 10) COMP, CURRENT_SIZE CS,
USER_SPECIFIED_SIZE US
  2    FROM v$memory_dynamic_components
  3   WHERE CURRENT_SIZE!=0;
Press [Enter] to continue...

COMP                 CS         US
---------- ---------- ----------
```

Practice 13-1: Using Automatic Memory Management (continued)

```
shared poo  167772160         0
large pool    4194304         0
java pool     4194304         0
SGA Target  289406976         0
DEFAULT bu  104857600         0
PGA Target  192937984         0

6 rows selected.

SQL> pause Press [Enter] to continue...
Press [Enter] to continue...

SQL>
```

5) Log in as the AMM user with the oracle_4U password, recreate the TABSGA table and insert rows by pressing Enter to continue the script.

```
SQL> connect amm
Enter password: oracle_4U <<< not displayed
Connected.
SQL> @amm_setup2.sql
SQL> REM "************************** "
SQL> REM "For training purposes ONLY
SQL> REM  Connected as the AMM user with the oracle_4U
password
SQL> set serveroutput on
SQL> set term on
SQL> set lines 200
SQL> set pages 44
SQL> set pause on pause "Press [Enter] to continue..."
SQL> drop table tabsga purge;
drop table tabsga purge
           *
ERROR at line 1:
ORA-00942: table or view does not exist


SQL>
SQL> create table tabsga(a number, b number) tablespace
tbssga;

Table created.

SQL>
SQL> begin
  2    for i in 1..100000 loop
  3      insert into tabsga values (i, i);
  4    end loop;
  5  end;
  6  /
```

Practice 13-1: Using Automatic Memory Management (continued)

```
PL/SQL procedure successfully completed.

SQL> commit;

Commit complete.

SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```

a) Modify the `TABSGA` table to "`parallel 64`", create a `TESTPGA` procedure (which creates a workload) by pressing Enter to continue the script.

```
SQL>
SQL> alter table tabsga parallel 64;

Table altered.

SQL>
SQL> create or replace procedure testpga( psize number ) as
  2  begin
  3  declare
  4  TYPE nAllotment_tabtyp    IS TABLE OF char(2048) INDEX BY
BINARY_INTEGER;
  5  myarray nAllotment_tabtyp;
  6  begin
  7    for i in 1..psize loop
  8       myarray(i) := to_char(i);
  9    end loop;
 10  end;
 11  end;
 12  /

Procedure created.
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```

b) Confirm that there are no errors and query the dynamic memory components again by pressing Enter to continue the script.

```
SQL> show errors
No errors.
SQL>
SQL> SELECT substr(COMPONENT, 0, 10) COMP, CURRENT_SIZE CS,
USER_SPECIFIED_SIZE US
  2    FROM v$memory_dynamic_components
  3   WHERE CURRENT_SIZE!=0;
Press [Enter] to continue...
```

c) To view the query results, press Enter to continue the script.

```
COMP               CS         US
---------- ---------- ----------
```

Practice 13-1: Using Automatic Memory Management (continued)

```
shared poo  167772160          0
large pool    4194304          0
java pool     4194304          0
SGA Target  289406976          0
DEFAULT bu  104857600          0
PGA Target  192937984          0

6 rows selected.

SQL>
SQL> pause Press [Enter] to exit the script...
Press [Enter] to exit the script...
```

d) Exit the script, but remain in the SQL*Plus session.

```
SQL> set pause off
SQL>
```

6) Connect as SYSDBA in your SQL*Plus session, shut down and start your database instance and then connect again as the AMM user with the oracle_4U password:

```
SQL> connect / as sysdba
Connected.
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
SQL> STARTUP
ORACLE instance started.

Total System Global Area  313860096 bytes
Fixed Size                  1336204 bytes
Variable Size             209718388 bytes
Database Buffers           96468992 bytes
Redo Buffers                6336512 bytes
Database mounted.
Database opened.
SQL>
SQL> connect amm
Enter password: oracle_4U <<< not displayed
Connected.
SQL>
```

7) As the AMM user, determine the current settings for the various memory buffers as well as the list of resized operations that were done since you started your instance.

a) You can use the amm_components.sql script for that purpose.

```
SQL> @amm_components.sql
SQL> set serveroutput on
SQL> set term on
```

```
SQL> set lines 200
SQL> set pages 100
SQL> set heading on
SQL> column comp format a18
SQL> column final_size format 999999999
SQL> column oper_type format a9
SQL> set pause on pause "Press [Enter] to continue..."
SQL>
SQL> SELECT substr(COMPONENT, 0, 18) COMP, CURRENT_SIZE CS,
USER_SPECIFIED_SIZE US
  2   FROM v$memory_dynamic_components
  3  WHERE CURRENT_SIZE!=0;
Press [Enter] to continue...
```

b) To view the query result, press Enter to continue the script.

```
COMP                       CS         US
------------------ ---------- ----------
shared pool          83886080          0
large pool            4194304          0
java pool             4194304          0
SGA Target          188743680          0
DEFAULT buffer cac   88080384          0
PGA Target          125829120          0
6 rows selected.

SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```

c) View the memory components (ordered by descending `START_TIME`) by pressing Enter to continue the script.

```
 SQL> SELECT substr(COMPONENT,0,20) comp, FINAL_SIZE,
OPER_TYPE, OPER_MODE, status
  2   FROM v$memory_resize_ops
  3  ORDER BY  START_TIME desc;
Press [Enter] to continue...
```

d) To view the query result, press Enter to continue the script.

```
COMP                 FINAL_SIZE OPER_TYPE OPER_MODE STATUS
------------------ ---------- --------- --------- ---------
shared pool          83886080 GROW      IMMEDIATE COMPLETE
DEFAULT buffer cac   88080384 SHRINK    IMMEDIATE COMPLETE
DEFAULT buffer cac   92274688 SHRINK    IMMEDIATE COMPLETE
shared pool          79691776 GROW      IMMEDIATE COMPLETE
large pool            4194304 GROW      IMMEDIATE COMPLETE
java pool             4194304 STATIC              COMPLETE
streams pool                0 STATIC              COMPLETE
SGA Target          188743680 STATIC              COMPLETE
PGA Target          125829120 STATIC              COMPLETE
DEFAULT buffer cac   96468992 STATIC    IMMEDIATE COMPLETE
```

```
DEFAULT buffer cac  100663296 STATIC    IMMEDIATE COMPLETE
ASM Buffer Cache            0 STATIC              COMPLETE
DEFAULT buffer cac  104857600 STATIC              COMPLETE
DEFAULT buffer cac  104857600 STATIC    IMMEDIATE COMPLETE
DEFAULT 2K buffer           0 STATIC              COMPLETE
DEFAULT 4K buffer           0 STATIC              COMPLETE
DEFAULT 8K buffer           0 STATIC              COMPLETE
DEFAULT 16K buffer          0 STATIC              COMPLETE
DEFAULT 32K buffer          0 STATIC              COMPLETE
KEEP buffer cache           0 STATIC              COMPLETE
RECYCLE buffer cac          0 STATIC              COMPLETE
large pool            4194304 STATIC              COMPLETE
shared pool          67108864 STATIC              COMPLETE
shared pool          71303168 GROW      IMMEDIATE COMPLETE
shared pool          75497472 GROW      IMMEDIATE COMPLETE
DEFAULT buffer cac   96468992 INITIALIZ IMMEDIATE COMPLETE
                              ING
26 rows selected.
SQL> pause Press [Enter] to exit the script...
Press [Enter] to exit the script...
```

```
SQL> set pause off
SQL>
SQL>
```

8) Remain connected as the AMM user in your SQL*Plus session and execute the following query. Immediately after that, determine the component sizes and resized operations. You can use query1.sql script for that purpose. What do you observe?

```
select /*+ PARALLEL(s 24) */ count(*) from (select /*+
parallel(s 24) */ * from tabsga s group by a);
```

   a) Execute the amm_query1.sql script. You can see that the large pool has a much bigger size while the buffer cache is smaller. This memory transfer was automatically done by the system.

```
SQL> @amm_query1.sql
SQL> select /*+ PARALLEL(s 24) */ count(*) from (select /*+
parallel(s 24) */ * from tabsga s group by a);

  COUNT(*)
----------
    100000

SQL>
SQL> column COMP format a12
SQL>
SQL> select substr(COMPONENT, 0, 10) COMP, CURRENT_SIZE CS,
USER_SPECIFIED_SIZE US from v$memory_dynamic_components where
CURRENT_SIZE!=0;
```

Practice 13-1: Using Automatic Memory Management (continued)

```
COMP                 CS         US
------------ ---------- ----------
shared poo     88080384          0
large pool     83886080          0
java pool       4194304          0
SGA Target    188743680          0
DEFAULT bu      4194304          0
PGA Target    125829120          0

6 rows selected.

SQL>
SQL> select substr(COMPONENT, 0, 10) COMP, FINAL_SIZE,
OPER_TYPE, OPER_MODE, status from v$memory_resize_ops order by
START_TIME;

COMP         FINAL_SIZE OPER_TYPE OPER_MODE STATUS
------------ ---------- --------- --------- ---------
shared poo     67108864 STATIC              COMPLETE
shared poo     71303168 GROW      IMMEDIATE COMPLETE
shared poo     75497472 GROW      IMMEDIATE COMPLETE
large pool      4194304 GROW      IMMEDIATE COMPLETE
large pool      4194304 STATIC              COMPLETE
java pool       4194304 STATIC              COMPLETE
streams po            0 STATIC              COMPLETE
SGA Target    188743680 STATIC              COMPLETE
ASM Buffer            0 STATIC              COMPLETE
DEFAULT bu    104857600 STATIC    IMMEDIATE COMPLETE
DEFAULT bu    104857600 STATIC              COMPLETE
DEFAULT bu    100663296 STATIC    IMMEDIATE COMPLETE
DEFAULT bu     96468992 STATIC    IMMEDIATE COMPLETE
DEFAULT bu     96468992 INITIALIZ IMMEDIATE COMPLETE
                        ING

DEFAULT 2K            0 STATIC              COMPLETE
DEFAULT 4K            0 STATIC              COMPLETE
DEFAULT 8K            0 STATIC              COMPLETE
DEFAULT 16            0 STATIC              COMPLETE
DEFAULT 32            0 STATIC              COMPLETE
KEEP buffe            0 STATIC              COMPLETE
RECYCLE bu            0 STATIC              COMPLETE
PGA Target    125829120 STATIC              COMPLETE
DEFAULT bu     92274688 SHRINK    IMMEDIATE COMPLETE
shared poo     79691776 GROW      IMMEDIATE COMPLETE
DEFAULT bu     88080384 SHRINK    IMMEDIATE COMPLETE
shared poo     83886080 GROW      IMMEDIATE COMPLETE
DEFAULT bu     83886080 SHRINK    DEFERRED  COMPLETE
shared poo     88080384 GROW      DEFERRED  COMPLETE
DEFAULT bu     79691776 SHRINK    IMMEDIATE COMPLETE
large pool      8388608 GROW      IMMEDIATE COMPLETE
```

```
DEFAULT bu     67108864 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu     71303168 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu     75497472 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu     62914560 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu     58720256 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu     54525952 SHRINK    IMMEDIATE COMPLETE
large pool     12582912 GROW      IMMEDIATE COMPLETE
large pool     16777216 GROW      IMMEDIATE COMPLETE
large pool     20971520 GROW      IMMEDIATE COMPLETE
large pool     29360128 GROW      IMMEDIATE COMPLETE
large pool     33554432 GROW      IMMEDIATE COMPLETE
large pool     25165824 GROW      IMMEDIATE COMPLETE
DEFAULT bu     29360128 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu     25165824 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu     20971520 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu     16777216 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu     16777216 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu     12582912 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu      8388608 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu     33554432 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu     37748736 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu     41943040 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu     46137344 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu     50331648 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu     54525952 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu      4194304 SHRINK    IMMEDIATE COMPLETE
large pool     79691776 GROW      IMMEDIATE COMPLETE
large pool     75497472 GROW      IMMEDIATE COMPLETE
large pool     71303168 GROW      IMMEDIATE COMPLETE
large pool     71303168 GROW      IMMEDIATE COMPLETE
large pool     67108864 GROW      IMMEDIATE COMPLETE
large pool     62914560 GROW      IMMEDIATE COMPLETE
large pool     58720256 GROW      IMMEDIATE COMPLETE
large pool     54525952 GROW      IMMEDIATE COMPLETE
large pool     50331648 GROW      IMMEDIATE COMPLETE
large pool     46137344 GROW      IMMEDIATE COMPLETE
large pool     41943040 GROW      IMMEDIATE COMPLETE
large pool     37748736 GROW      IMMEDIATE COMPLETE
large pool     33554432 GROW      IMMEDIATE COMPLETE
large pool     83886080 GROW      IMMEDIATE COMPLETE

70 rows selected.
SQL>
SQL>
```

9) Redo the same thing as in the previous step, but this time use the following query. You can use `amm_query2.sql` script for that purpose. What do you observe?

   Possible Answer: The same trend continues.

```
SQL> @amm_query2.sql
```

Practice 13-1: Using Automatic Memory Management (continued)

```
SQL> select /*+ PARALLEL(s 25) */ count(*) from (select /*+
parallel(s 25) */ * from tabsga s group by a);

  COUNT(*)
----------
    100000

SQL>
SQL> column COMP format a12
SQL>
SQL> select substr(COMPONENT, 0, 10) COMP, CURRENT_SIZE CS,
USER_SPECIFIED_SIZE US from v$memory_dynamic_components where
CURRENT_SIZE!=0;


COMP                 CS         US
------------ ---------- ----------
shared poo     88080384          0
large pool     83886080          0
java pool       4194304          0
SGA Target    188743680          0
DEFAULT bu      4194304          0
PGA Target    125829120          0

6 rows selected.

SQL>
SQL> select substr(COMPONENT, 0, 10) COMP, FINAL_SIZE,
OPER_TYPE, OPER_MODE, status from v$memory_resize_ops order by
START_TIME;

COMP         FINAL_SIZE OPER_TYPE OPER_MODE STATUS
------------ ---------- --------- --------- ---------
shared poo     67108864 STATIC              COMPLETE
shared poo     71303168 GROW      IMMEDIATE COMPLETE
shared poo     75497472 GROW      IMMEDIATE COMPLETE
large pool      4194304 GROW      IMMEDIATE COMPLETE
large pool      4194304 STATIC              COMPLETE
java pool       4194304 STATIC              COMPLETE
streams po            0 STATIC              COMPLETE
SGA Target    188743680 STATIC              COMPLETE
ASM Buffer            0 STATIC              COMPLETE
DEFAULT bu    104857600 STATIC    IMMEDIATE COMPLETE
DEFAULT bu    104857600 STATIC              COMPLETE
DEFAULT bu    100663296 STATIC    IMMEDIATE COMPLETE
DEFAULT bu     96468992 STATIC    IMMEDIATE COMPLETE
DEFAULT bu     96468992 INITIALIZ IMMEDIATE COMPLETE
                        ING

DEFAULT 2K            0 STATIC              COMPLETE
```

Practice 13-1: Using Automatic Memory Management (continued)

```
DEFAULT 4K          0 STATIC              COMPLETE
DEFAULT 8K          0 STATIC              COMPLETE
DEFAULT 16          0 STATIC              COMPLETE
DEFAULT 32          0 STATIC              COMPLETE
KEEP buffe          0 STATIC              COMPLETE
RECYCLE bu          0 STATIC              COMPLETE
PGA Target  125829120 STATIC              COMPLETE
DEFAULT bu   92274688 SHRINK    IMMEDIATE COMPLETE
shared poo   79691776 GROW      IMMEDIATE COMPLETE
DEFAULT bu   88080384 SHRINK    IMMEDIATE COMPLETE
shared poo   83886080 GROW      IMMEDIATE COMPLETE
DEFAULT bu   83886080 SHRINK    DEFERRED  COMPLETE
shared poo   88080384 GROW      DEFERRED  COMPLETE
DEFAULT bu   79691776 SHRINK    IMMEDIATE COMPLETE
large pool    8388608 GROW      IMMEDIATE COMPLETE
DEFAULT bu   67108864 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu   71303168 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu   75497472 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu   62914560 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu   58720256 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu   54525952 SHRINK    IMMEDIATE COMPLETE
large pool   12582912 GROW      IMMEDIATE COMPLETE
large pool   16777216 GROW      IMMEDIATE COMPLETE
large pool   20971520 GROW      IMMEDIATE COMPLETE
large pool   29360128 GROW      IMMEDIATE COMPLETE
large pool   33554432 GROW      IMMEDIATE COMPLETE
large pool   25165824 GROW      IMMEDIATE COMPLETE
DEFAULT bu   29360128 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu   25165824 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu   20971520 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu   16777216 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu   16777216 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu   12582912 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu    8388608 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu   33554432 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu   37748736 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu   41943040 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu   46137344 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu   50331648 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu   54525952 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu    4194304 SHRINK    IMMEDIATE COMPLETE
large pool   79691776 GROW      IMMEDIATE COMPLETE
large pool   75497472 GROW      IMMEDIATE COMPLETE
large pool   71303168 GROW      IMMEDIATE COMPLETE
large pool   71303168 GROW      IMMEDIATE COMPLETE
large pool   67108864 GROW      IMMEDIATE COMPLETE
large pool   62914560 GROW      IMMEDIATE COMPLETE
large pool   58720256 GROW      IMMEDIATE COMPLETE
large pool   54525952 GROW      IMMEDIATE COMPLETE
large pool   50331648 GROW      IMMEDIATE COMPLETE
```

```
large pool     46137344 GROW      IMMEDIATE COMPLETE
large pool     41943040 GROW      IMMEDIATE COMPLETE
large pool     37748736 GROW      IMMEDIATE COMPLETE
large pool     33554432 GROW      IMMEDIATE COMPLETE
large pool     83886080 GROW      IMMEDIATE COMPLETE

70 rows selected.
SQL>
```

10) Still connected as the AMM user from your SQL*Plus session, execute the following command and, immediately afterward, determine the memory component sizes and the list of resize operations. You can use amm_query3.sql for that purpose. What do you observe?

Possible Answer: The same style of growing and shrinking of the memory components.

```
SQL> @amm_query3.sql
SQL> exec testpga(500000);

PL/SQL procedure successfully completed.

SQL>
SQL> column COMP format a12
SQL>
SQL> select substr(COMPONENT, 0, 10) COMP, CURRENT_SIZE CS,
USER_SPECIFIED_SIZE US from v$memory_dynamic_components where
CURRENT_SIZE!=0;

COMP                 CS         US
------------ ---------- ----------

shared poo     88080384          0
large pool     83886080          0
java pool       4194304          0
SGA Target    188743680          0
DEFAULT bu      4194304          0
PGA Target    125829120          0

6 rows selected.

SQL>
SQL> select substr(COMPONENT, 0, 10) COMP, FINAL_SIZE,
OPER_TYPE, OPER_MODE, status from v$memory_resize_ops order by
START_TIME;

COMP         FINAL_SIZE OPER_TYPE OPER_MODE STATUS
------------ ---------- --------- --------- ---------
DEFAULT bu     96468992 INITIALIZ IMMEDIATE COMPLETE
                        ING
```

Practice 13-1: Using Automatic Memory Management (continued)

```
RECYCLE bu            0 STATIC                COMPLETE
KEEP buffe            0 STATIC                COMPLETE
DEFAULT 32            0 STATIC                COMPLETE
shared poo     75497472 GROW       IMMEDIATE COMPLETE
shared poo     71303168 GROW       IMMEDIATE COMPLETE
shared poo     67108864 STATIC                COMPLETE
large pool      4194304 STATIC                COMPLETE
large pool      4194304 GROW       IMMEDIATE COMPLETE
java pool       4194304 STATIC                COMPLETE
streams po            0 STATIC                COMPLETE
SGA Target    188743680 STATIC                COMPLETE
PGA Target    125829120 STATIC                COMPLETE
DEFAULT bu     96468992 STATIC     IMMEDIATE COMPLETE
DEFAULT bu    100663296 STATIC     IMMEDIATE COMPLETE
ASM Buffer            0 STATIC                COMPLETE
DEFAULT bu    104857600 STATIC                COMPLETE
DEFAULT bu    104857600 STATIC     IMMEDIATE COMPLETE
DEFAULT 2K            0 STATIC                COMPLETE
DEFAULT 4K            0 STATIC                COMPLETE
DEFAULT 8K            0 STATIC                COMPLETE
DEFAULT 16            0 STATIC                COMPLETE
DEFAULT bu     92274688 SHRINK     IMMEDIATE COMPLETE
shared poo     79691776 GROW       IMMEDIATE COMPLETE
DEFAULT bu     88080384 SHRINK     IMMEDIATE COMPLETE
shared poo     83886080 GROW       IMMEDIATE COMPLETE
shared poo     88080384 GROW       DEFERRED  COMPLETE
DEFAULT bu     83886080 SHRINK     DEFERRED  COMPLETE
DEFAULT bu     79691776 SHRINK     IMMEDIATE COMPLETE
large pool      8388608 GROW       IMMEDIATE COMPLETE
DEFAULT bu     75497472 SHRINK     IMMEDIATE COMPLETE
DEFAULT bu     71303168 SHRINK     IMMEDIATE COMPLETE
DEFAULT bu     67108864 SHRINK     IMMEDIATE COMPLETE
DEFAULT bu     62914560 SHRINK     IMMEDIATE COMPLETE
DEFAULT bu     58720256 SHRINK     IMMEDIATE COMPLETE
DEFAULT bu     54525952 SHRINK     IMMEDIATE COMPLETE
large pool     12582912 GROW       IMMEDIATE COMPLETE
large pool     16777216 GROW       IMMEDIATE COMPLETE
large pool     20971520 GROW       IMMEDIATE COMPLETE
large pool     25165824 GROW       IMMEDIATE COMPLETE
large pool     29360128 GROW       IMMEDIATE COMPLETE
large pool     33554432 GROW       IMMEDIATE COMPLETE
DEFAULT bu     37748736 SHRINK     IMMEDIATE COMPLETE
large pool     75497472 GROW       IMMEDIATE COMPLETE
DEFAULT bu     46137344 SHRINK     IMMEDIATE COMPLETE
DEFAULT bu     50331648 SHRINK     IMMEDIATE COMPLETE
DEFAULT bu     54525952 SHRINK     IMMEDIATE COMPLETE
large pool     83886080 GROW       IMMEDIATE COMPLETE
large pool     79691776 GROW       IMMEDIATE COMPLETE
DEFAULT bu      4194304 SHRINK     IMMEDIATE COMPLETE
```

Practice 13-1: Using Automatic Memory Management (continued)

```
DEFAULT bu       8388608 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu      12582912 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu      16777216 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu      16777216 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu      20971520 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu      25165824 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu      29360128 SHRINK    IMMEDIATE COMPLETE
DEFAULT bu      33554432 SHRINK    IMMEDIATE COMPLETE
large pool      33554432 GROW      IMMEDIATE COMPLETE
large pool      37748736 GROW      IMMEDIATE COMPLETE
large pool      41943040 GROW      IMMEDIATE COMPLETE
large pool      46137344 GROW      IMMEDIATE COMPLETE
large pool      50331648 GROW      IMMEDIATE COMPLETE
large pool      54525952 GROW      IMMEDIATE COMPLETE
large pool      58720256 GROW      IMMEDIATE COMPLETE
large pool      62914560 GROW      IMMEDIATE COMPLETE
large pool      67108864 GROW      IMMEDIATE COMPLETE
large pool      71303168 GROW      IMMEDIATE COMPLETE
large pool      71303168 GROW      IMMEDIATE COMPLETE
DEFAULT bu      41943040 SHRINK    IMMEDIATE COMPLETE

70 rows selected.
```

```
SQL> exit
$
```

11) From a terminal window, restart Enterprise Manager Database Control.

```
$ emctl start dbconsole

Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009 Oracle Corporation.  All rights
reserved.
https://edrsr37p1.us.oracle.com:1158/em/console/aboutApplicati
on
Starting Oracle Enterprise Manager 11g Database Control
.............. started.
--------------------------------------------------------------
----
Logs are generated in directory
/u01/app/oracle/product/11.2.0/dbhome_1/edrsr37p1.us.oracle.co
m_orcl/sysman/log
$
```

12) In Enterprise Manager look at the memory variations that happened during this lab. What do you observe?

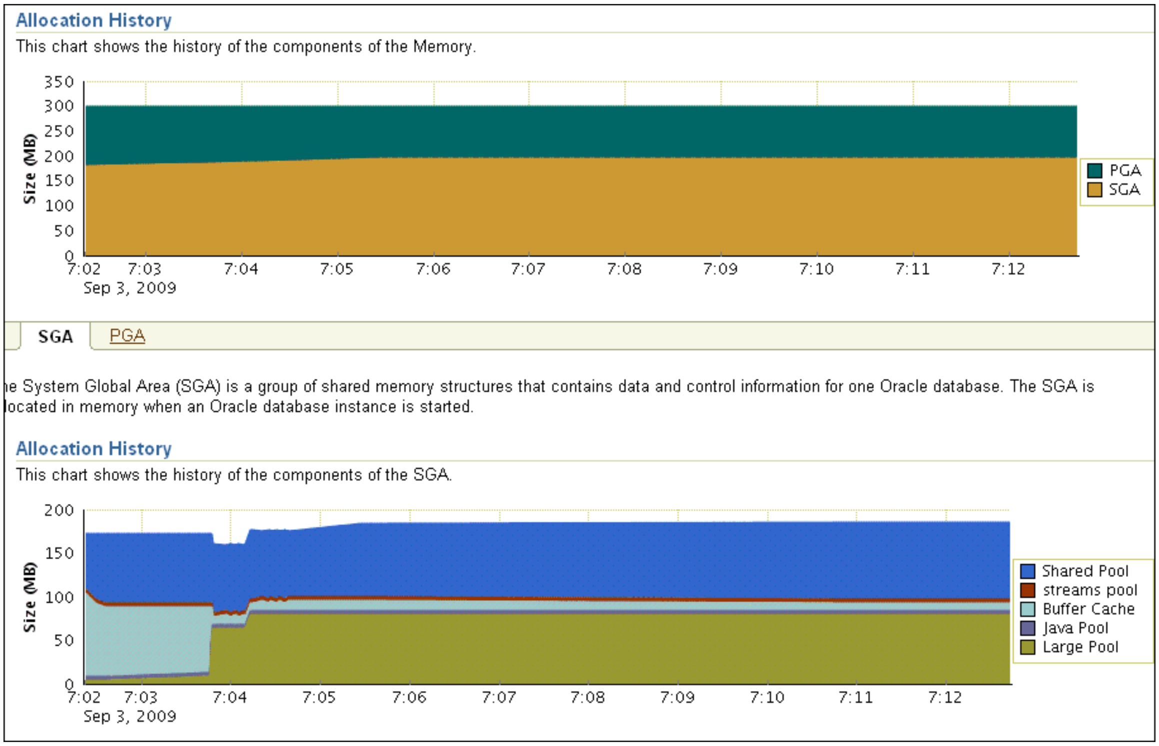
a) Logged into Enterprise Manager as the SYSDBA, click the Server tab.

Practice 13-1: Using Automatic Memory Management (continued)

13) On the Server tabbed page, click Memory Advisors in the Database Configuration section.

14) On the Memory Advisors page, look at the first two graphics.

15) You should see modifications of the memory components in the second graph, that the large pool grew and shrank.



16) Logout and exit from Enterprise Manager.

17) To clean up your environment, shut down your database instance, restore the original `SPFILE`, turn on archiving and restart your `orcl` database instance. To do all that, execute the `amm_cleanup.sh` script.

```
$ cd ~/labs
$ ./amm_cleanup.sh
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009 Oracle Corporation.  All rights
reserved.
https://edrsr37p1.us.oracle.com:1158/em/console/aboutApplicati
on
Stopping Oracle Enterprise Manager 11g Database Control ...
 ...  Stopped.

SQL*Plus: Release 11.2.0.1.0 Production on Thu Sep 3 19:21:43
2009

Copyright (c) 1982, 2009, Oracle.  All rights reserved.
```

Practice 13-1: Using Automatic Memory Management (continued)

```
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining
and Real Application Testing options

SQL> SQL>
User dropped.

SQL>
Tablespace dropped.

SQL>
Tablespace dropped.

SQL> SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
File created.

SQL> SQL> ORACLE instance started.

Total System Global Area  481259520 bytes
Fixed Size                  1337352 bytes
Variable Size             373295096 bytes
Database Buffers          100663296 bytes
Redo Buffers                5963776 bytes
Database mounted.
SQL>
Database altered.

SQL>
Database altered.

SQL> Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining
and Real Application Testing options
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009 Oracle Corporation.  All rights
reserved.
https://edrsr37p1.us.oracle.com:1158/em/console/aboutApplicati
on
```

Practice 13-1: Using Automatic Memory Management (continued)

```
Starting Oracle Enterprise Manager 11g Database Control
............... started.
------------------------------------------------------------------
----
Logs are generated in directory
/u01/app/oracle/product/11.2.0/dbhome_1/edrsr37p1.us.oracle.co
m_orcl/sysman/log
$
```

Practice 14-1: Monitoring Services

In your database there are several running applications. You want to monitor the resources that are being used by each application. Create a service configuration for each application or application function that uses your database.

In this practice, you create the following configuration in the `orcl` database:

Service Name	Usage	Response Time (sec)– Warning/Critical
SERV1	Client service	0.4, 1.0

1) Use the `DBMS_SERVICE` package to create a service called `SERV1`. Then make sure that you add your service name to your `tnsnames.ora` file.

   a) The recommended method for adding a service name to the `tnsnames.ora` file is to use Net Manager. For this exercise, execute the `sv1_add.sh` script. Review the `tnsnames.ora` file at `$ORACLE_HOME/network/admin` to confirm that the following lines are included. Substitute the output of the `hostname` command for `<hostname>` below.

```
SERV1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)
      (HOST = <hostname>.ua.oracle.com)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = SERV1.example.com)
    )
  )
```

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
$ cd /home/oracle/labs
$ ./sv1_add.sh
edrsr37p1.us.oracle.com
$
```

   b) Use the `DBMS_SERVICE.CREATE_SERVICE` procedure to create a service. (The command is entered on one line.)

```
$ sqlplus / as sysdba
```

```
SQL> EXEC
DBMS_SERVICE.CREATE_SERVICE('SERV1','SERV1.example.com')

PL/SQL procedure successfully completed.

SQL> exit;
```

2) After you have created your services, try connecting to your database by using your service name. What happens? Why?

   Answer: You cannot connect using your service because although it is defined, it is not started on your instance. You can verify this by looking at the `SERVICE_NAME` initialization parameter and by looking at the services known to the listener.

```
$ lsnrctl services
LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 26-JUL-
2009 16:23:46

Copyright (c) 1991, 2009, Oracle.  All rights reserved.

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
Services Summary...
Service "+ASM" has 1 instance(s).
  Instance "+ASM", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "DEDICATED" established:10098 refused:0 state:ready
         LOCAL SERVER
Service "orcl.us.oracle.com" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "DEDICATED" established:3657 refused:0 state:ready
         LOCAL SERVER
Service "orclXDB.us.oracle.com" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "D000" established:0 refused:0 current:0 max:1022
state:ready
         DISPATCHER <machine: edrsr37p1.us.oracle.com, pid:
6610>

(ADDRESS=(PROTOCOL=tcp)(HOST=edrsr37p1.us.oracle.com)(PORT=296
69))
Service "rcat.example.com" has 1 instance(s).
  Instance "rcat", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "DEDICATED" established:3 refused:0 state:ready
         LOCAL SERVER
Service "rcatXDB.example.com" has 1 instance(s).
```

```
  Instance "rcat", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "D000" established:0 refused:0 current:0 max:1022
state:ready
         DISPATCHER <machine: edrsr37p1.us.oracle.com, pid:
20809>

(ADDRESS=(PROTOCOL=tcp)(HOST=edrsr37p1.us.oracle.com)(PORT=620
66))
The command completed successfully
$
```

```
$ sqlplus / as sysdba

SQL> show parameter service

NAME                             TYPE        VALUE
------------------------------ ----------- ----------------
service_names                    string      orcl.oracle.com
```

```
SQL> connect system@SERV1
Enter password: oracle_4U <<< not displayed
ERROR:
ORA-12514: TNS:listener does not currently know of service
requested in connect descriptor

Warning: You are no longer connected to ORACLE.
SQL>
```

3) How would you make sure that you can connect using your service? Do it and connect to your instance by using your service.

   Answer: You must start your service on your instance.

```
$ connect / as sysdba
Connected.

SQL> show parameter service

NAME                             TYPE        VALUE
------------------------------ ----------- ----------------
service_names                    string      orcl.us.oracle.com

SQL> EXEC DBMS_SERVICE.START_SERVICE('SERV1')

PL/SQL procedure successfully completed.

SQL> show parameter service

NAME                         TYPE        VALUE
------------------------- ----------- --------------------
```

```
service_names                string      SERV1.us.oracle.com

SQL> host lsnrctl services
LSNRCTL for Linux: Version 11.2.0.1.0 - Production on 26-JUL-
2009 16:30:36

Copyright (c) 1991, 2009, Oracle.  All rights reserved.

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
Services Summary...
Service "+ASM" has 1 instance(s).
  Instance "+ASM", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "DEDICATED" established:10105 refused:0 state:ready
         LOCAL SERVER
Service "SERV1.example.com" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "DEDICATED" established:0 refused:0 state:ready
         LOCAL SERVER
Service "orcl.us.oracle.com" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "DEDICATED" established:0 refused:0 state:ready
         LOCAL SERVER
Service "orclXDB.us.oracle.com" has 1 instance(s).
  Instance "orcl", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "D000" established:0 refused:0 current:0 max:1022
state:ready
         DISPATCHER <machine: edrsr37p1.us.oracle.com, pid:
6610>

(ADDRESS=(PROTOCOL=tcp)(HOST=edrsr37p1.us.oracle.com)(PORT=296
69))
Service "rcat.example.com" has 1 instance(s).
  Instance "rcat", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "DEDICATED" established:3 refused:0 state:ready
         LOCAL SERVER
Service "rcatXDB.example.com" has 1 instance(s).
  Instance "rcat", status READY, has 1 handler(s) for this
service...
    Handler(s):
      "D000" established:0 refused:0 current:0 max:1022
state:ready
```

```
         DISPATCHER <machine: edrsr37p1.us.oracle.com, pid:
20809>

(ADDRESS=(PROTOCOL=tcp)(HOST=edrsr37p1.us.oracle.com)(PORT=620
66))
The command completed successfully

SQL>
```

```
SQL> connect system@SERV1
Enter password: oracle_4U <<< not displayed

Connected.
SQL> exit
```

4) Execute the `sv1_load.sh` script as `SYSDBA`. This script creates a new `SV_USER` user. Then you connect to your instance as this user and the `SERV1` service. Create workload activity by executing the `sv1_load2.sql` script. If this script finishes before you completed the next step, then use the `sv1_sel.sql` script to executes the following query:

```
SELECT COUNT(*) FROM DBA_OBJECTS,DBA_OBJECTS,DBA_OBJECTS
```

```
$ cd ~/labs
$ ./sv1_load.sh

SQL> SQL> SQL> SQL> SQL> drop user sv_user cascade
          *
ERROR at line 1:
ORA-01918: user 'SV_USER' does not exist


SQL> SQL>   2    3
User created.

SQL> SQL>
Grant succeeded.
$
```

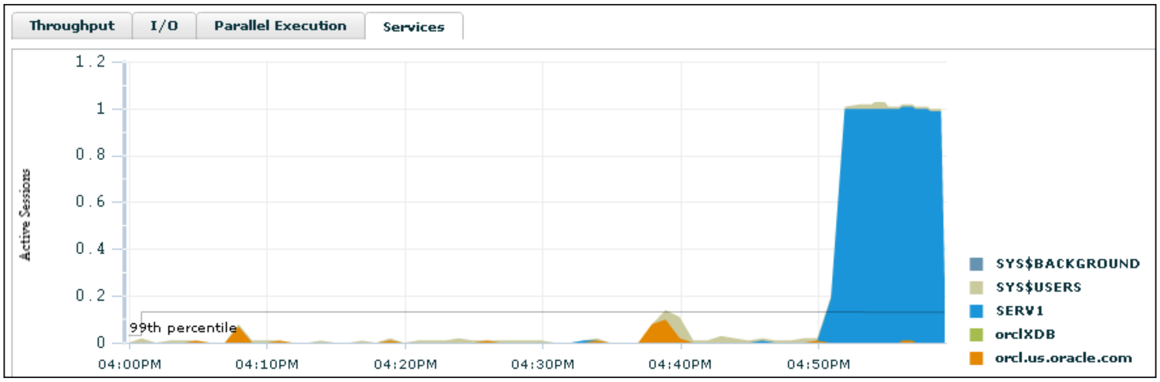
Note: Do not wait for the script to complete before proceeding to the next step.

```
$ sqlplus sv_user@SERV1
Enter password: oracle_4U <<< not displayed
Connected.

SQL> @sv1_load2.sql
SQL> DECLARE
  2  t number;
  3  BEGIN
  4  for i in 1..2000 loop
  5    select count(*) into t from dba_objects;
  6  end loop;
```

```
  7   END;
  8   /
```

5) After the execution starts, access the EM Top Consumers page from the Performance tabbed page, and check if `SERV1` is using more resources. Also, check the statistics on your service with `V$SERVICE_STATS` from a SQL*Plus session connected as `SYSDBA`.

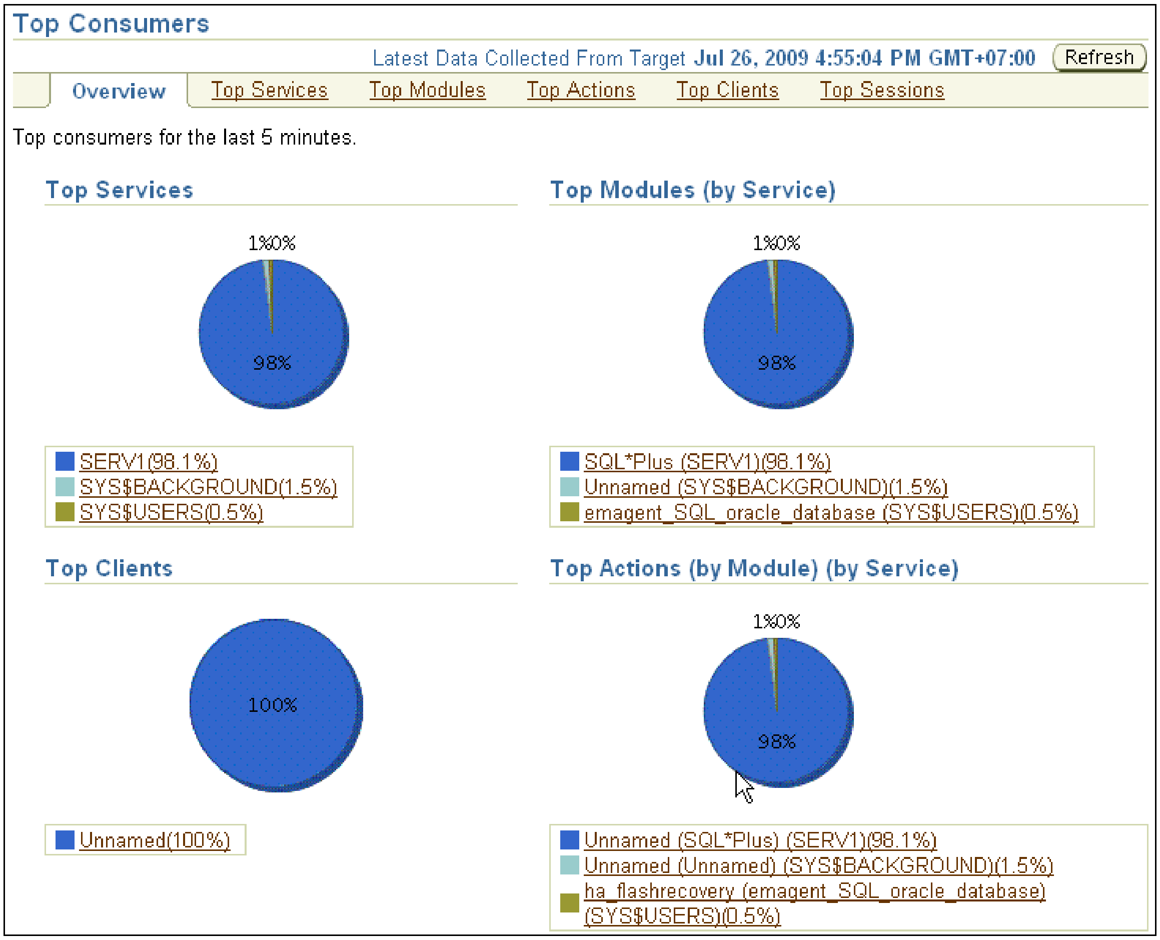
   a) On the home page, click the Performance tab. Towards the bottom of the Performance page, click the Services tab.



An Active Session graph with the activity aggregated by service name is displayed. The network service name of each connection is recorded as a separate service. So all the connections made without a service name are aggregated, as are all the connections made as `SERV1`.

   b) Click the Top Consumers link in the Additional Monitoring Links section. Refresh the Top Consumers Overview page several times.

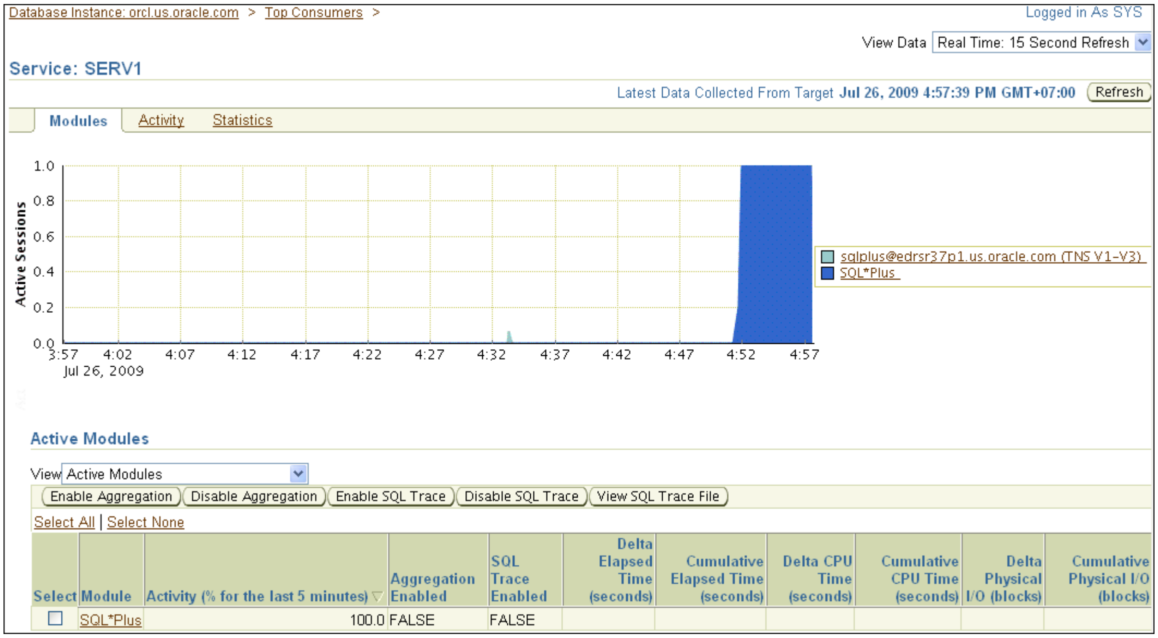
Practice 14-1: Monitoring Services (continued)



The names and number of services listed in the Top Services Graph depends on the number and type of connections to the database.

c) You can also see the detailed statistics by navigating to the Top Services tab > SERV1 link > Statistics tab.

Practice 14-1: Monitoring Services (continued)



6) If the `sv1_load2.sql` script finishes before you completed this step, then use the `sv1_sel.sql` script to continue creating a workload. – When you completed the tasks, make sure that you stop your running workload by pressing Ctrl + C in your terminal window.

```
SQL> @sv1_sel.sql
SQL> select count(*) from dba_objects,dba_objects,dba_objects
                                      *
ERROR at line 1:
ORA-01013: user requested cancel of current operation

SQL> exit
```

Practices for Lesson 15

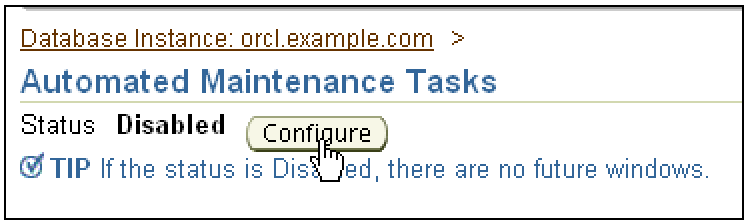
By default, Automatic SQL Tuning executes automatically during each nightly maintenance window. For this practice, you simulate the execution of Automatic SQL Tuning, and explore its results

Practice 15-1: Using Automatic SQL Tuning

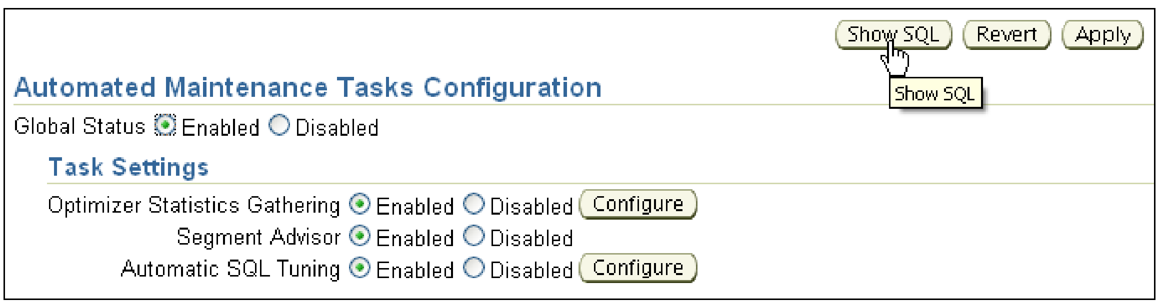
In this practice, you manually launch Automatic SQL Tuning to automatically tune a small application workload. You then investigate the outcomes and configuration possibilities.

1) In EM, navigate to Server > Automated Maintenance Tasks (in the Oracle Scheduler section).

   a) To check if the task settings are enabled, click Configure, (no matter if the status is Enabled or Disabled).



   b) Toggle Enabled and Disabled, then click Show SQL when the Global Status is Enabled.



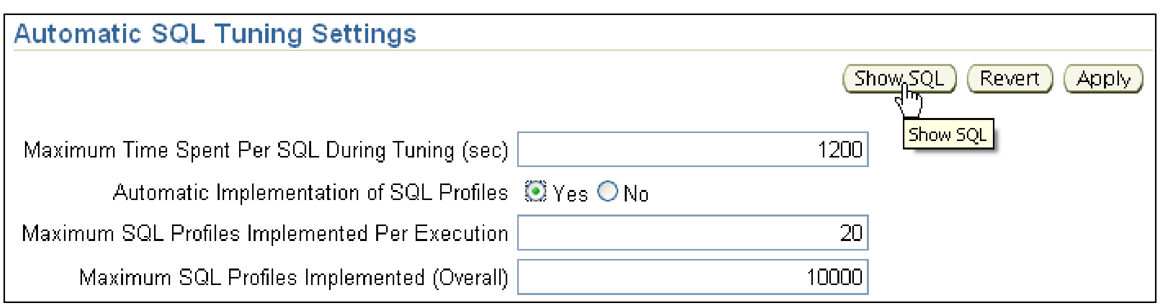
   c) Review the command and click Return.

```
BEGIN
dbms_auto_task_admin.enable();
END;
```

   d) On the Automated Maintenance Tasks Configuration page, click Apply.

   e) Click the Configure button next to Automatic SQL Tuning.

   f) Select Yes for "Automatic Implementation of SQL Profiles" and click Show SQL.



   g) Review the command and click Return.

Return

```
BEGIN
dbms_sqltune.set_auto_tuning_task_parameter( 'ACCEPT_SQL_PROFILES', 'TRUE');
END;
```

h) On the Automatic SQL Tuning Settings page, click Apply.

You should receive a success message.

2) In a terminal window connected as the `oracle` user, point to the `ORCL` instance, review and execute the `ast_setup.sh` script .This script creates the AST user, turns off automatic maintenance tasks, and drops any existing profiles on queries executed by the AST user.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
$ cat ast_setup.sh
#!/bin/bash
#  For training only - execute as oracle OS user

sqlplus / as sysdba <<EOF!
set echo on

drop user ast cascade;
create user ast identified by "oracle_4U";
grant dba to ast;

alter system flush shared_pool;
--
-- Turn off AUTOTASK
--
alter system set "_enable_automatic_maintenance"=0;

--
-- Clear out old executions of auto-sqltune
--
exec
dbms_sqltune.reset_tuning_task('SYS_AUTO_SQL_TUNING_TASK');

--
-- Drop any profiles on AST queries
--
declare
  cursor prof_names is
    select name from dba_sql_profiles where sql_text like
'%AST%';
begin
  for prof_rec in prof_names loop
    dbms_sqltune.drop_sql_profile(prof_rec.name);
  end loop;
end;
/
```

```
EOF!
$
```

```
$ ./ast_setup.sh

SQL> SQL> SQL> drop user ast cascade
          *
ERROR at line 1:
ORA-01918: user 'AST' does not exist

SQL>
User created.

SQL>
Grant succeeded.

SQL> SQL>
System altered.

SQL> SQL> SQL> SQL>
System altered.

SQL> SQL> SQL> SQL> SQL>
PL/SQL procedure successfully completed.

SQL> SQL> SQL> SQL> SQL>   2    3    4    5    6    7    8
9
PL/SQL procedure successfully completed.

SQL> SQL>
$
```

3) In preparation for the practice, you should log in as the `AST` user with the `oracle_4U` password and execute a workload. Execute the `ast_workload_stream.sql` script. This script executes, multiple times a query that is not correctly optimized. The query in question uses hints that force the optimizer to pick a suboptimal execution plan. The script executes for approximately 30 to 60 seconds. (O*utput has been reduced to minimize clutter*.)

```
$ sqlplus ast
Enter password: oracle_4U <<< not displayed
SQL> @ast_workload_stream.sql
Sun Aug  2 23:31:33 GMT-7 2009
no rows selected
no rows selected
no rows selected
.
.
.
no rows selected
no rows selected
```

Practice 15-1: Using Automatic SQL Tuning (continued)

```
Sun Aug  2 23:31:55 GMT-7 2009
SQL> exit
$
```

4) Automatic SQL Tuning is implemented using an automated task that runs during maintenance windows. However, you are not going to wait for the next maintenance window to open. This might take too long. Instead, you will force the opening of your next maintenance window now. This will automatically trigger the Automatic SQL Tuning task. Review and execute the `ast_run.sh` script to do that. The script's execution takes about ten minutes (most likely).

```
$ cat ast_run.sh
#!/bin/bash
# For training only - execute as oracle OS user
date

sqlplus / as sysdba <<EOF!
set echo on
set serveroutput on

exec dbms_workload_repository.create_snapshot;

variable window varchar2(20);
begin
 select upper(to_char(sysdate,'fmday'))||'_WINDOW' into
:window from dual;
end;
/
print window;

--
-- Open the corresponding maintenance window, but with other
clients disabled
--
alter system set "_enable_automatic_maintenance"=1
/
exec dbms_auto_task_admin.disable( -
  'auto optimizer stats collection', null, :window);

exec dbms_auto_task_admin.disable( -
  'auto space advisor', null, :window);

exec dbms_scheduler.open_window(:window, null, true);

--
-- Close the maintenance window when sqltune is done
--
exec dbms_lock.sleep(60);

declare
  running number;
```

```
begin
  loop
    select count(*)
    into   running
    from   dba_advisor_executions
    where  task_name = 'SYS_AUTO_SQL_TUNING_TASK' and
           status = 'EXECUTING';
    if (running = 0) then
      exit;
    end if;
    dbms_lock.sleep(60);
  end loop;
  dbms_scheduler.close_window(:window);
end;
/
alter system set "_enable_automatic_maintenance"=0
/

-- Re-enable the other guys so they look like they are enabled
in EM.
-- Still they will be disabled because we have set the
underscore.
--

exec dbms_auto_task_admin.enable( -
  'auto optimizer stats collection', null, :window);

exec dbms_auto_task_admin.enable( -
  'auto space advisor', null, :window);

EOF!

date

$
```

```
$ ./ast_run.sh
Sun Aug  2 23:42:54 GMT-7 2009

SQL> SQL> SQL> SQL>
PL/SQL procedure successfully completed.

SQL> SQL> SQL>   2    3    4
PL/SQL procedure successfully completed.

SQL>
WINDOW
--------------------------------
SUNDAY_WINDOW

SQL> SQL> SQL> SQL> SQL>   2
```

Practice 15-1: Using Automatic SQL Tuning (continued)

```
System altered.

SQL> >
PL/SQL procedure successfully completed.

SQL> SQL> >
PL/SQL procedure successfully completed.

SQL> SQL>
PL/SQL procedure successfully completed.

SQL> SQL> SQL> SQL> SQL>
PL/SQL procedure successfully completed.

SQL> SQL>   2    3    4    5    6    7    8    9   10   11
12   13   14   15   16   17
PL/SQL procedure successfully completed.

SQL>   2
System altered.

SQL> SQL> SQL> SQL> SQL> SQL> >
PL/SQL procedure successfully completed.

SQL> SQL> >
PL/SQL procedure successfully completed.

SQL> SQL>
Sun Aug  2 23:43:57 GMT-7 2009
$
```

Some of your output, like the WINDOW, may look different.

5) Execute the `ast_workload_stream.sh` script again. What do you observe?

   a) You should see that the execution time for `ast_workload_stream.sh` is much faster than the original execution. This is probably due to the fact that Automatic SQL Tuning implemented a profile for your statement automatically.

```
$ sqlplus ast
Enter password: oracle_4U <<< not displayed
SQL> @ast_workload_stream.sql
Sun Aug  2 23:48:54 GMT-7 2009
no rows selected
no rows selected
no rows selected
.
.
.
no rows selected
no rows selected

Sun Aug  2 23:49:16 GMT-7 2009
```

```
SQL> exit
$
```

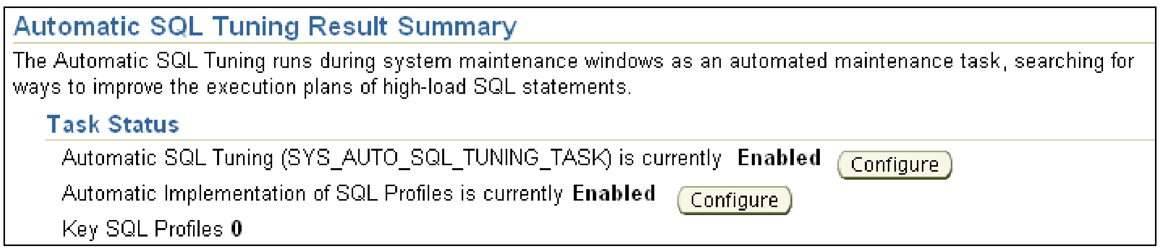
6) Logged in as the `AST` user, force the creation of an AWR snapshot.

```
$ sqlplus ast
Enter password: oracle_4U <<< not displayed
SQL> set echo on
SQL> exec dbms_workload_repository.create_snapshot;
PL/SQL procedure successfully completed.

SQL> exit
$
```

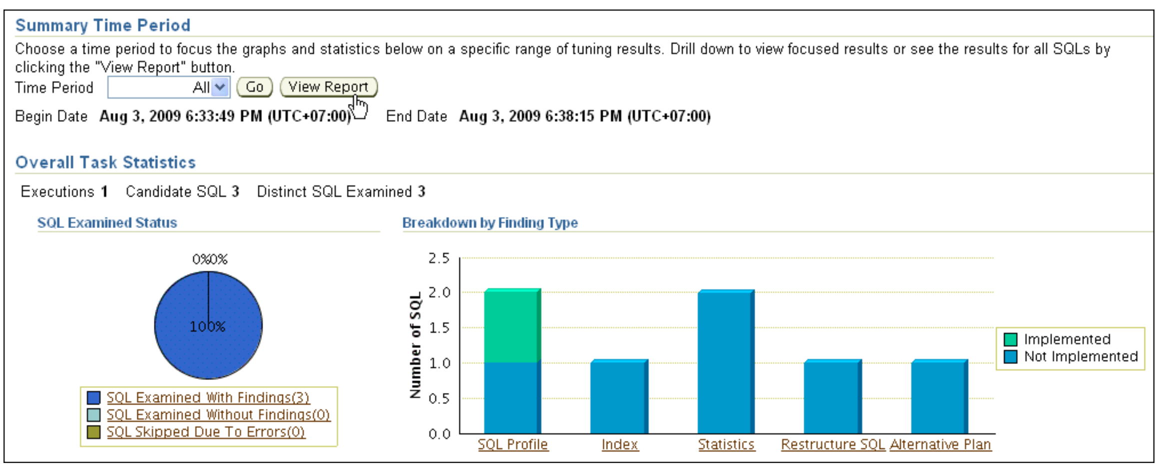
7) How can you confirm that a SQL Profile was automatically implemented?

   a) In Enterprise Manager, navigate to Server > Automated Maintenance Tasks (Oracle Scheduler) > Automatic SQL Tuning.

   b) On the Automatic SQL Tuning summary page, view the tuning results.



The task has already run in one maintenance window and has results ready to be viewed.

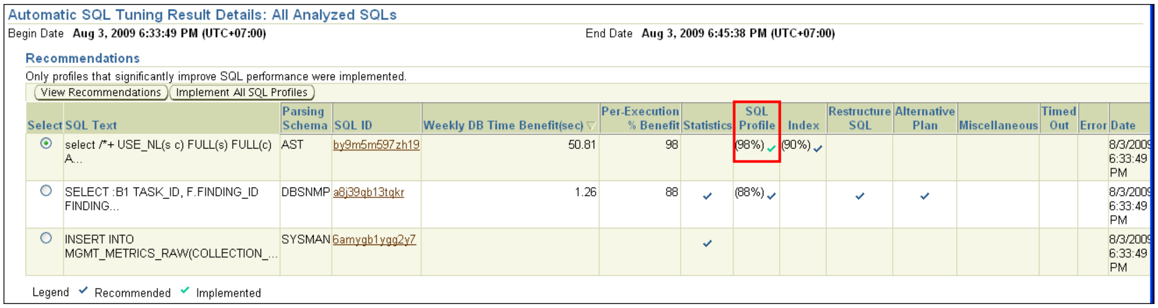


   c) Look at the graphs on the Automatic SQL Tuning Result Summary page. (*If you do not see any graphs, return to step 5, execute the work load twice, then continue with step 6 and 7.*)

   d) Focus on understanding the pie chart and the bar graph next to it. You should be able to get a feeling for the general findings breakdown, as well as the number of SQL profiles implemented by the task.

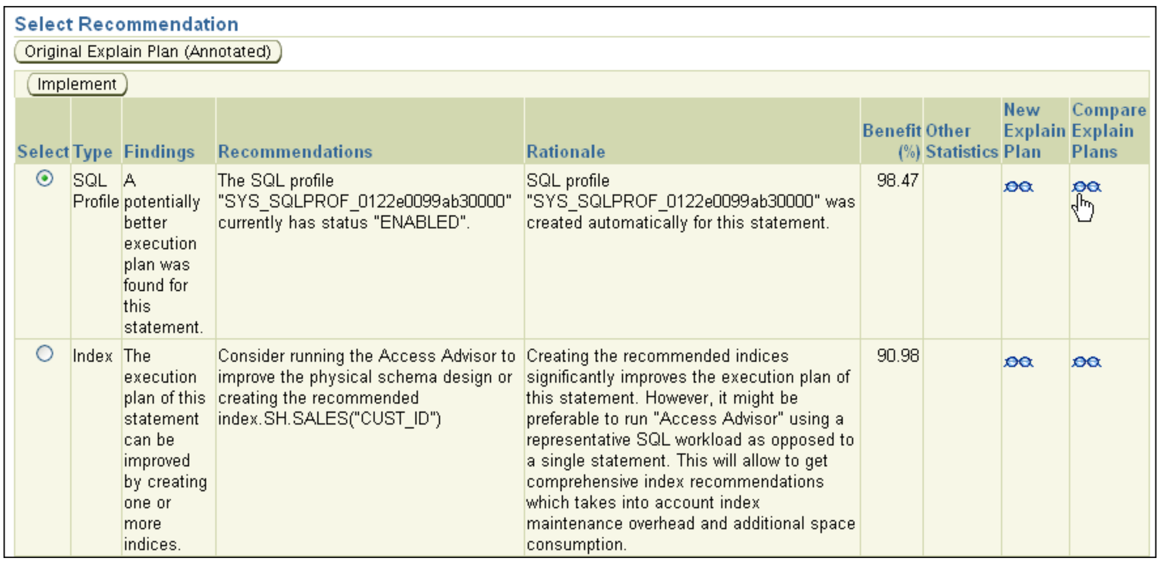
Practice 15-1: Using Automatic SQL Tuning (continued)

e) Click View Report to see a detailed SQL-level report.

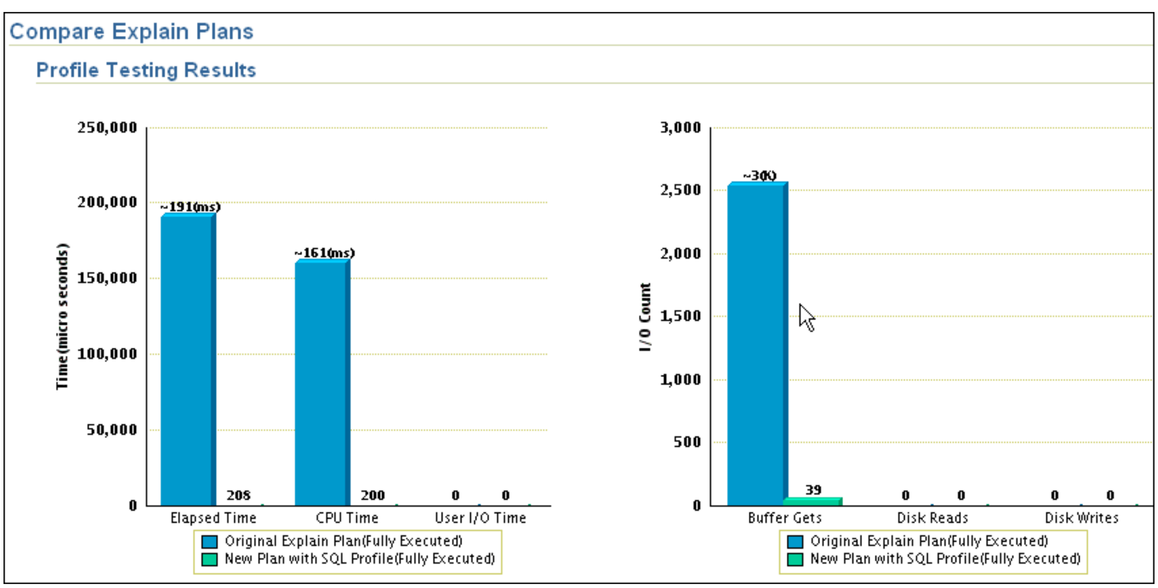


f) Find and select the SQL that ran in the AST schema. Note the green check mark meaning that the profile was implemented.

g) Click the View Recommendations button.



h) Click the Compare Explain Plans eyeglass icon for the SQL Profile entry.



Practice 15-1: Using Automatic SQL Tuning (continued)

i) Scroll down the page.

Original Explain Plan (Annotated)

○ Indicates an adjustment from the original plan by the SQL Tuning Advisor

Plan Hash Value **4005616876**

Expand All | Collapse All

Operation	Line ID	Object	Object Type	Order	Rows	Bytes	Cost	Time	CPU Cost	I/O Cost
▼ SELECT STATEMENT	0			6		0.013	○ 893	11	○ 269,312,608	○ 885
▼ HASH GROUP BY	1			5		0.013	○ 893	11	○ 269,312,608	○ 885
▼ NESTED LOOPS	2			4		0.013	○ 892	11	○ 236,425,824	○ 885
TABLE ACCESS FULL	3	SH.CUSTOMERS	TABLE	1		0.005	○ 405	5	○ 21,682,460	○ 404
▼ PARTITION RANGE ALL	4			3		0.008	○ 488	6	○ 214,743,360	○ 481
TABLE ACCESS FULL	5	SH.SALES	TABLE	2		0.008	○ 488	6	○ 214,743,360	○ 481

New Explain Plan With SQL Profile

Plan Hash Value **3070788227**

Expand All | Collapse All

Operation	Line ID	Object	Object Type	Order	Rows	Bytes	Cost	Time	CPU Cost	I/O Cost
▼ SELECT STATEMENT	0			8		0.013	55	1	33,327,052	54
▼ HASH GROUP BY	1			7		0.013	55	1	33,327,052	54
▼ NESTED LOOPS	2			6		0.013	54	1	440,260	54
▼ PARTITION RANGE ALL	3			4		0.008	54	1	438,310	54
▼ TABLE ACCESS BY LOCAL INDEX ROWID	4	SH.SALES	TABLE	3		0.008	54	1	438,310	54
▼ BITMAP CONVERSION TO ROWIDS	5			2						
BITMAP INDEX RANGE SCAN	6	SH.SALES_CUST_BIX	INDEX (BITMAP)	1						
INDEX UNIQUE SCAN	7	SH.CUSTOMERS_PK	INDEX (UNIQUE)	5		0.005	0	1	1,950	0

j) Look at the old and new explain plans for the query.

k) Then click the "Recommendations for SQL ID" locator link (the last of the breadcrumbs on top of the page) to return to the previous screen.

SQL Text

select /*+ USE_NL(s c) FULL(s) FULL(c) AST */ c.cust_id, sum(s.quantity_sold) from sh.sales s, sh.customers c where s.cust_id = c.cust_id and c.cust_id < 2 group by c.cust_id

l) Investigate a SQL profile. While still on the "Recommendations for `SQL_ID`" page, click the SQL text to go to the SQL Details page for this SQL.

Statistics | Activity | Plan | Plan Control | **Tuning History** | SQL Monitoring

SQL Tuning History

The following SQL tuning tasks provide the recommendations to tune this SQL statement.

Advisor Task Name	Advisor Task Owner	Task Completion
SYS_AUTO_SQL_TUNING_TASK	SYS	Aug 3, 2009 6:34:13 PM

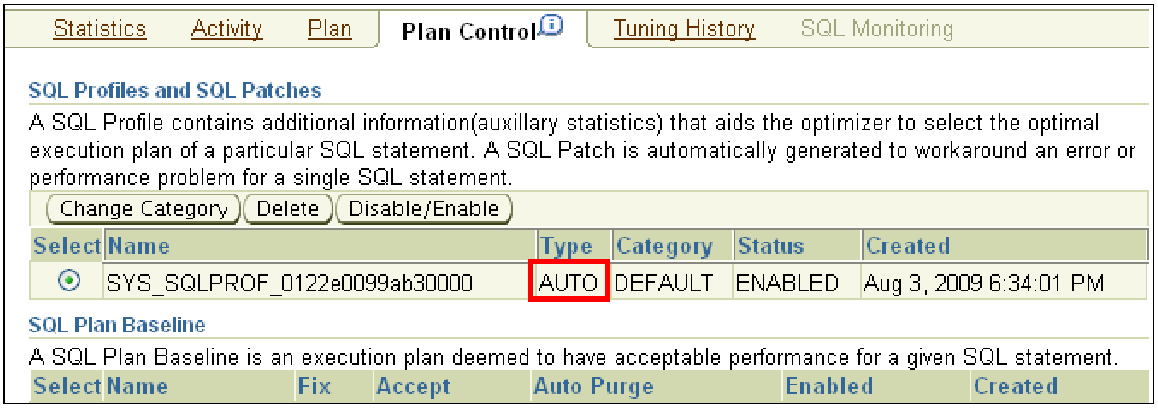
ADDM Findings for this SQL during historic period

Finding Name	Occurrences (during selected historical period)

Practice 15-1: Using Automatic SQL Tuning (continued)

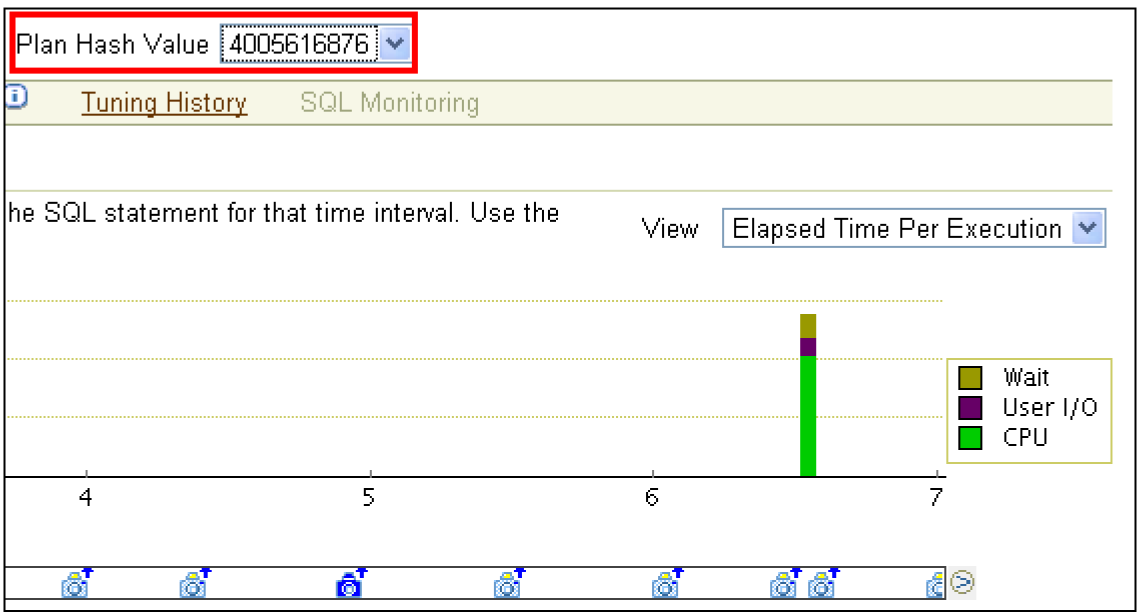
m) On the SQL Details - Tuning History page note the link to `SYS_AUTO_SQL_TUNING_TASK` that is there to show that the SQL was tuned by this tuning task.

n) Click the Plan Control tab.



o) Note that a profile was created automatically for this SQL. The type of `AUTO` means it was automatically created.

p) Click the Statistics tab to take a look at the execution history for this SQL.



q) Depending on the speed of your machine, you may not see two hash values. If that is the case, ignore this step and the following one. Select Real Time: Manual Refresh from the View Data and then each of possible two Plan Hash Values from the corresponding drop-down list. Choose one after the other and wait for the page to refresh each time.

r) Depending on the speed of your environment, you should see one statement with a relatively high elapsed time per execution, and one with very low elapsed time per execution. This shows the improved plan. If you select All from the Plan Hash Values drop-down list, you might not be able to see the execution corresponding to the statement after tuning on the Summary graph. This might be because the workload was too short to execute.

8) Generate a text report for more in-depth information. From the command line, execute `ast_task_report.sh` script. What do you observe?

   a) Notice the first queries that fetch execution name and object number from the advisor schema, followed by the final query that gets the text report. In the text report, look for the section about the SQL profile finding and peruse the Validation Results section. This shows you the execution statistics observed during test-execute and allows you to get more of a feeling about the profile's quality. You can also use the `report_auto_tuning_task` API to get reports that span multiple executions of the task.

```
$ cat ast_task_report.sh
#!/bin/bash
# For training only - execute as oracle OS user

sqlplus / as sysdba <<EOF!
set echo on
set long 1000000000
set longchunksize 1000
set serveroutput on

--
-- Check the execution names
--
alter session set nls_date_format = 'MM/DD/YYYY HH24:MI:SS';

select execution_name, status, execution_start
 from   dba_advisor_executions
 where  task_name = 'SYS_AUTO_SQL_TUNING_TASK'
 order by execution_start;

variable last_exec varchar2(30);

begin
  select max(execution_name) keep (dense_rank last order by
execution_start)
  into   :last_exec
  from   dba_advisor_executions
  where  task_name = 'SYS_AUTO_SQL_TUNING_TASK';
end;
/

print :last_exec

--
```

Practice 15-1: Using Automatic SQL Tuning (continued)

```
-- Find the object ID for query AST with sql_id by9m5m597zh19
--
variable obj_id number;

begin
  select object_id
  into   :obj_id
  from   dba_advisor_objects
  where  task_name = 'SYS_AUTO_SQL_TUNING_TASK' and
         execution_name = :last_exec and
         type = 'SQL' and
         attr1 = 'by9m5m597zh19';
end;
/

print :obj_id

--
-- Get a text report to drill down on this one query
--
set pagesize 0
select dbms_sqltune.report_auto_tuning_task(
  :last_exec, :last_exec, 'TEXT', 'TYPICAL', 'ALL', :obj_id)
from dual;

EOF!

$
```

```
$ ./ast_task_report.sh

SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL>
Session altered.

SQL> SQL>   2    3    4
EXECUTION_NAME                 STATUS      EXECUTION_START
------------------------------ ----------- -------------------
EXEC_176                       COMPLETED   08/03/2009 18:33:49

SQL> SQL> SQL> SQL>   2    3    4    5    6    7
PL/SQL procedure successfully completed.

SQL> SQL>
LAST_EXEC
--------------------------------
EXEC_176

SQL> SQL> SQL> SQL> SQL> SQL> SQL>   2    3    4    5    6
7    8    9   10
PL/SQL procedure successfully completed.
```

Practice 15-1: Using Automatic SQL Tuning (continued)

```
SQL> SQL>
    OBJ_ID
----------
         3

SQL> SQL> SQL> SQL> SQL> SQL>   2    3  GENERAL INFORMATION
SECTION
-------------------------------------------------------------
-----------------
Tuning Task Name                       :
SYS_AUTO_SQL_TUNING_TASK
Tuning Task Owner                      : SYS
Workload Type                          : Automatic High-Load
SQL Workload
Scope                                  : COMPREHENSIVE
Global Time Limit(seconds)             : 3600
Per-SQL Time Limit(seconds)            : 1200
Completion Status                      : COMPLETED
Started at                             : 08/03/2009 18:33:49
Completed at                           : 08/03/2009 18:34:13
Number of Candidate SQLs               : 3
Cumulative Elapsed Time of SQL (s)     : 53

-------------------------------------------------------------
-----------------
Object ID  : 3
Schema Name: AST
SQL ID     : by9m5m597zh19
SQL Text   : select /*+ USE_NL(s c) FULL(s) FULL(c) AST */
c.cust_id,
             sum(s.quantity_sold) from sh.sales s,
sh.customers c where
             s.cust_id = c.cust_id and c.cust_id < 2 group by
c.cust_id

-------------------------------------------------------------
-----------------
FINDINGS SECTION (2 findings)
-------------------------------------------------------------
-----------------

1- SQL Profile Finding (see explain plans section below)
-------------------------------------------------------
  A potentially better execution plan was found for this
statement.
  SQL profile "SYS_SQLPROF_0122e0099ab30000" was created
automatically for
  this statement.

  Recommendation (estimated benefit: 98.47%)
  ------------------------------------------
```

```
  - An automatically-created SQL profile is present on the
system.
    Name:   SYS_SQLPROF_0122e0099ab30000
    Status: ENABLED

  Validation results
  ------------------
  The SQL profile was tested by executing both its plan and
the original plan
  and measuring their respective execution statistics. A plan
may have been
  only partially executed if the other could be run to
completion in less time.

                           Original Plan  With SQL Profile  %
Improved
                           -------------  ----------------  --
--------
  Completion Status:            COMPLETE          COMPLETE
  Elapsed Time(us):               191076               208
99.89 %
  CPU Time(us):                   160775               200
99.87 %
  User I/O Time(us):                   0                 0
  Buffer Gets:                      2541                39
98.46 %
  Physical Read Requests:              0                 0
  Physical Write Requests:             0                 0
  Physical Read Bytes:                 0                 0
  Physical Write Bytes:                0                 0
  Rows Processed:                      0                 0
  Fetches:                             0                 0
  Executions:                          1                 1

  Notes
  -----
  1. The original plan was first executed to warm the buffer
cache.
  2. Statistics for original plan were averaged over next 4
executions.
  3. The SQL profile plan was first executed to warm the
buffer cache.
  4. Statistics for the SQL profile plan were averaged over
next 9 executions.

2- Index Finding (see explain plans section below)
--------------------------------------------------
  The execution plan of this statement can be improved by
creating one or more
  indices.

  Recommendation (estimated benefit: 90.98%)
```

Practice 15-1: Using Automatic SQL Tuning (continued)

```
  ------------------------------------------
  - Consider running the Access Advisor to improve the
physical schema design
    or creating the recommended index.
    create index SH.IDX$$_00010001 on SH.SALES("CUST_ID");

  Rationale
  ---------
    Creating the recommended indices significantly improves
the execution plan
    of this statement. However, it might be preferable to run
"Access Advisor"
    using a representative SQL workload as opposed to a single
statement. This
    will allow to get comprehensive index recommendations
which takes into
    account index maintenance overhead and additional space
consumption.

--------------------------------------------------------------
-----------------
EXPLAIN PLANS SECTION
--------------------------------------------------------------
-----------------

1- Original With Adjusted Cost
------------------------------
Plan hash value: 4005616876

--------------------------------------------------------------
------------------
-------------------
| Id  | Operation             | Name      | Rows  | Bytes |
Cost (%CPU)| Time
  | Pstart| Pstop |
--------------------------------------------------------------
------------------
-------------------
|   0 | SELECT STATEMENT      |           |     1 |    13 |
893   (1)| 00:00:1
1 |       |       |
|   1 |  HASH GROUP BY        |           |     1 |    13 |
893   (1)| 00:00:1
1 |       |       |
|   2 |   NESTED LOOPS        |           |     1 |    13 |
892   (1)| 00:00:1
1 |       |       |
|*  3 |    TABLE ACCESS FULL  | CUSTOMERS |     1 |     5 |
405   (1)| 00:00:0
5 |       |       |
|   4 |    PARTITION RANGE ALL|           |     1 |     8 |
488   (2)| 00:00:0
```

```
6 |      1 |     28 |
|*  5 |     TABLE ACCESS FULL | SALES     |     1 |     8 |
488   (2)| 00:00:0
6 |      1 |     28 |
--------------------------------------------------------------
------------------
-------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   3 - filter("C"."CUST_ID"<2)
   5 - filter("S"."CUST_ID"<2 AND "S"."CUST_ID"="C"."CUST_ID")

2- Using SQL Profile
--------------------
Plan hash value: 3070788227

--------------------------------------------------------------
------------------
--------------------------------------
| Id  | Operation                            | Name
| Rows  | Bytes |
Cost (%CPU)| Time     | Pstart| Pstop |
--------------------------------------------------------------
------------------
--------------------------------------
|   0 | SELECT STATEMENT                     |
|     1 |    13 |
   55   (2)| 00:00:01 |       |       |
|   1 |  HASH GROUP BY                       |
|     1 |    13 |
   55   (2)| 00:00:01 |       |       |
|   2 |   NESTED LOOPS                       |
|     1 |    13 |
   54   (0)| 00:00:01 |       |       |
|   3 |    PARTITION RANGE ALL               |
|     1 |     8 |
   54   (0)| 00:00:01 |     1 |    28 |
|   4 |     TABLE ACCESS BY LOCAL INDEX ROWID| SALES
|     1 |     8 |
   54   (0)| 00:00:01 |     1 |    28 |
|   5 |      BITMAP CONVERSION TO ROWIDS     |
|       |       |
           |          |       |       |
|*  6 |       BITMAP INDEX RANGE SCAN        | SALES_CUST_BIX
|       |       |
           |          |     1 |    28 |
|*  7 |    INDEX UNIQUE SCAN                 | CUSTOMERS_PK
|     1 |     5 |
    0   (0)| 00:00:01 |       |       |
```

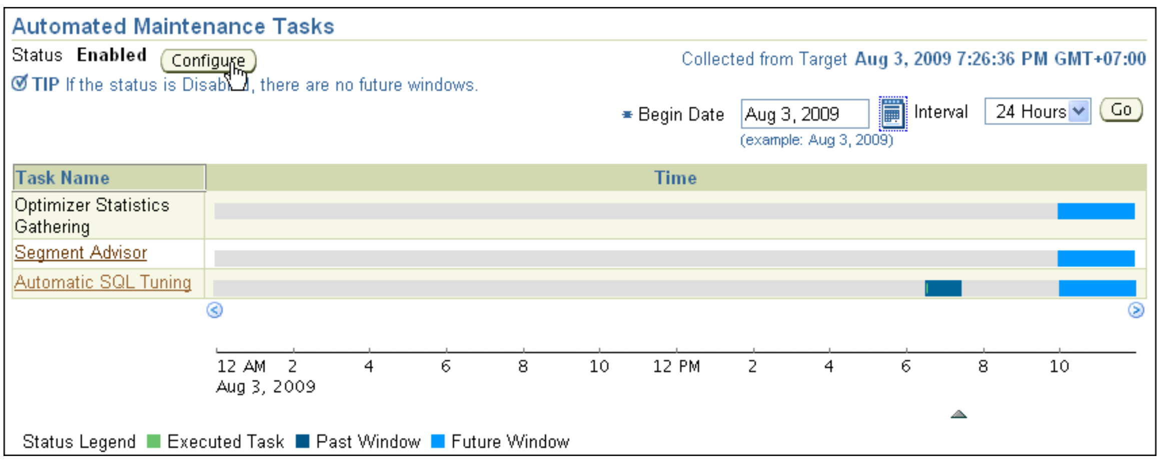
```
--------------------------------------------------------------
-----------------
--------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   6 - access("S"."CUST_ID"<2)
       filter("S"."CUST_ID"<2)
   7 - access("S"."CUST_ID"="C"."CUST_ID")
       filter("C"."CUST_ID"<2)

3- Using New Indices
--------------------
Plan hash value: 1871796534

--------------------------------------------------------------
-----------------
--------------------------------------
| Id  | Operation                          | Name
| Rows  | Bytes |
Cost (%CPU)| Time     | Pstart| Pstop |
--------------------------------------------------------------
-----------------
--------------------------------------
|   0 | SELECT STATEMENT                   |
|     1 |    13 |
    5   (0)| 00:00:01 |       |       |
|   1 |  SORT GROUP BY NOSORT              |
|     1 |    13 |
    5   (0)| 00:00:01 |       |       |
|   2 |   NESTED LOOPS                     |
|       |       |
           |          |       |       |
|   3 |    NESTED LOOPS                    |
|     1 |    13 |
    5   (0)| 00:00:01 |       |       |
|*  4 |     INDEX RANGE SCAN               | CUSTOMERS_PK
|     1 |     5 |
    2   (0)| 00:00:01 |       |       |
|*  5 |     INDEX RANGE SCAN               | IDX$$_00010001
|     1 |       |
    2   (0)| 00:00:01 |       |       |
|   6 |    TABLE ACCESS BY GLOBAL INDEX ROWID| SALES
|     1 |     8 |
    3   (0)| 00:00:01 | ROWID | ROWID |
--------------------------------------------------------------
-----------------
--------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------
```

```
   4 - access("C"."CUST_ID"<2)
   5 - access("S"."CUST_ID"="C"."CUST_ID")
       filter("S"."CUST_ID"<2)

--------------------------------------------------------------
-----------------

SQL> SQL>
$
```

9) Investigate how to configure Automatic SQL Tuning with Enterprise Manager.

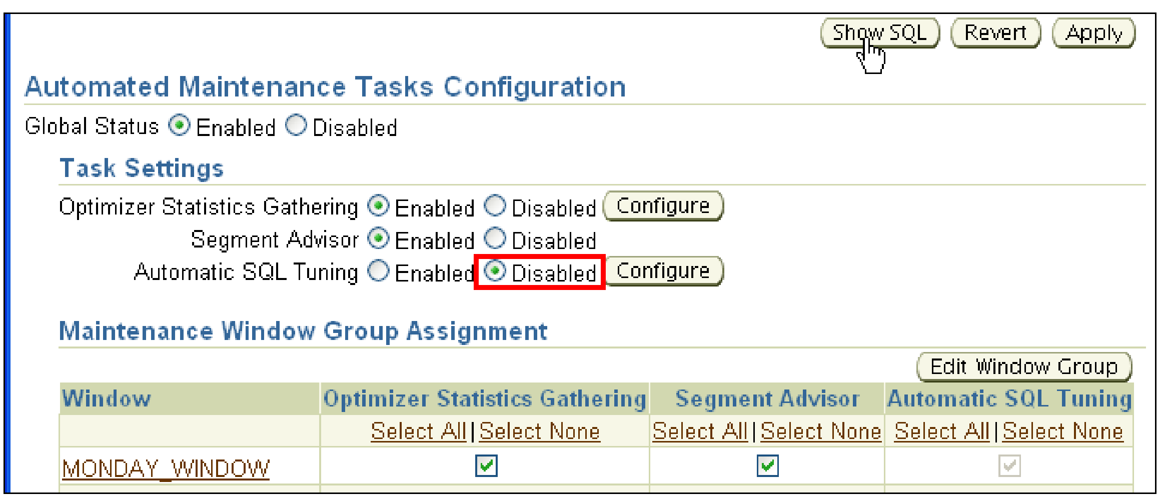
   a) Logged into Enterprise Manager as the SYS user, navigate to Server > Automated Maintenance Tasks.

   b) The chart here shows times in the past when each client was executed, and times in the future when they are scheduled to run again.



   c) Modify the graph's begin and end points with the widgets in the upper right.

   d) Click the Configure button.



Practice 15-1: Using Automatic SQL Tuning (continued)

On the Automated Maintenance Tasks Configuration page, you can disable individual clients and change which windows they run in.

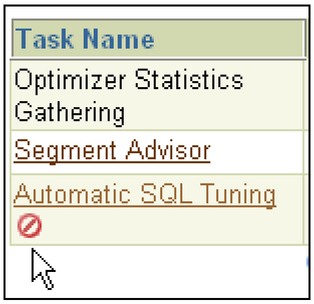
e) Disable the Automatic SQL Tuning client entirely and click Show SQL.

```
BEGIN
dbms_auto_task_admin.disable(client_name => 'sql tuning advisor',
operation => NULL, window_name => NULL);
END;
```

f) Review the commands and then click Return.

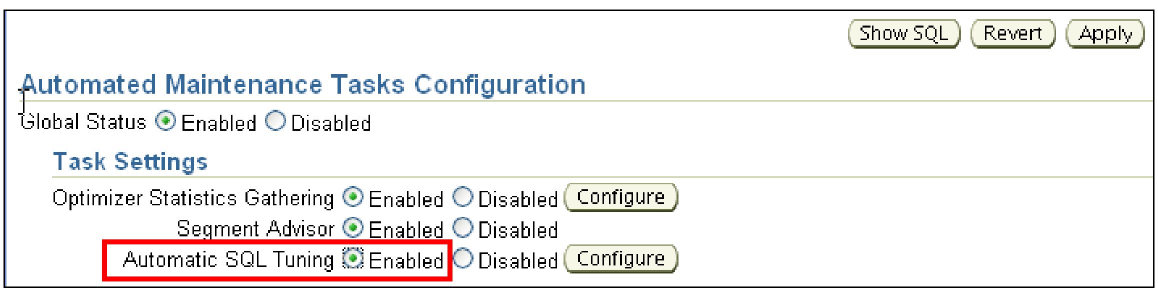
g) On the Automated Maintenance Tasks Configuration page, click Apply. You should receive a success message.

h) Click the Automated Maintenance Tasks locator link.



i) Notice the forbidden sign right next to the task name.

j) Click Configure.



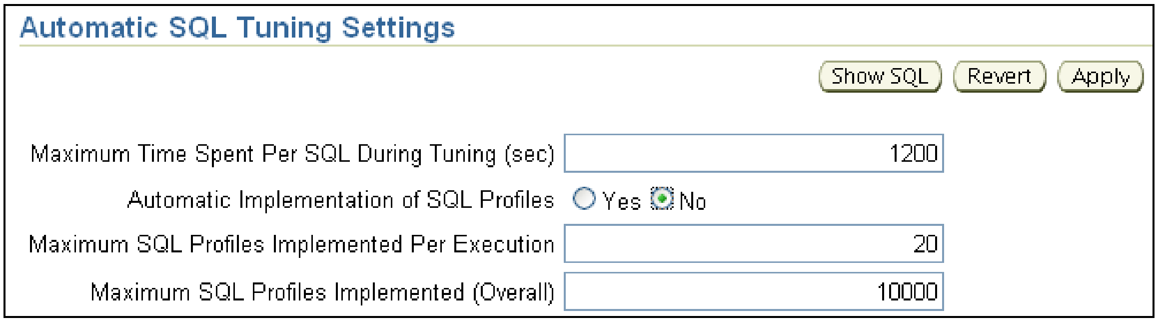
k) Enable the task again.

```
BEGIN
dbms_auto_task_admin.enable(client_name => 'sql tuning advisor',
operation => NULL, window_name => NULL);
END;
```

l) Optionally, click Show SQl, review the commands and then click Return.

m) Then click Apply to enable Automatic SQL Tuning. You should receive a success message.

n) Navigate to the Automatic SQL Tuning page. If you are on the Automated Maintenance Tasks Configuration page, click the Configure button for Automatic SQL Tuning.



o) On the Automatic SQL Tuning Settings page, select No in front of the “Automatic Implementation of SQL Profiles” field, and click Show SQL.

```
BEGIN
dbms_sqltune.set_auto_tuning_task_parameter( 'ACCEPT_SQL_PROFILES',
'FALSE');
END;
```

p) Review the command, click Return, and then click Apply. You should receive a success message.

10) OPTIONAL: You can investigate how to configure Automatic SQL Tuning using PL/SQL looking at the following script: `ast_manual_config.sh` and script. **Note:** In your case, the task executes quickly because the workload to take into account is really small. However, you could use the `ast_interrupt_task.sh` script from another session to stop the task, should it last too long.

Practices for Lesson 16

You received complaints that certain batch jobs are using too many system resources and that a specific user is known to start data warehouse processes during regular business hours. You decide to use the Database Resource Manager for better system-resource utilization and control.

Your first effort to balance the situation includes creating an `APPUSER` consumer group and assigning it to the default `DEFAULT_PLAN` resource plan. You then map a couple of Oracle users and your major OS user to resource groups. Activate the resource plan and test your assignments. Regularly click Show SQL to review all statements that are new to you.

Practice 16-1: Managing Resources

In this practice, you create an `APPUSER` consumer group and assign it to the default `DEFAULT_PLAN` resource plan. Then you map a couple of Oracle users and your major OS user to resource groups. Activate the resource plan and test your assignments.

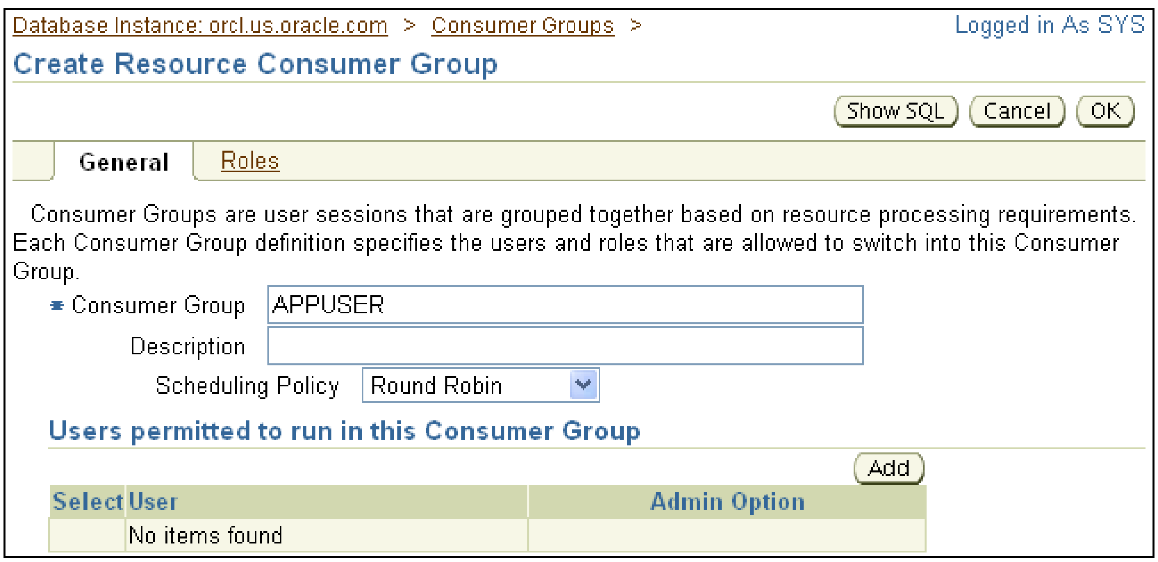
Log in as the `SYS` user (with `oracle_4U` password, connect as `SYSDBA`) and perform the necessary tasks through Enterprise Manager Database Control or through SQL*Plus. All scripts for this practice are in the `/home/oracle/labs` directory.

Whenever you open a new terminal window, execute the `oraenv` script to set environment variables for the `orcl` database.

1) Using Enterprise Manager Database Control, create a resource group called `APPUSER`. At this point, do not add users to the group.

   a) In Enterprise Manager, select Server > Consumer Groups (in the Resource Manager section).

   b) On the Consumer Groups page, click the Create button.

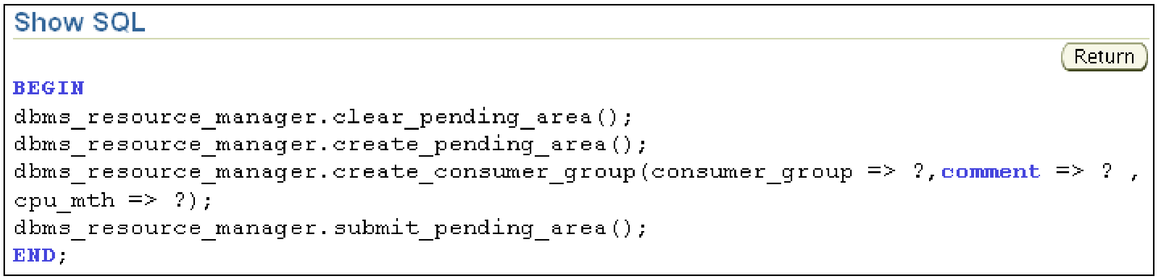


   c) Enter `APPUSER` as Consumer Group and ensure that the Scheduling Policy is set to Round Robin.

   Question 1: What does the `ROUND-ROBIN` parameter value mean?

   Possible Answer: `ROUND-ROBIN` indicates that CPU resources are fairly allocated to the `APPUSER` consumer group, according to the active resource plan directives.

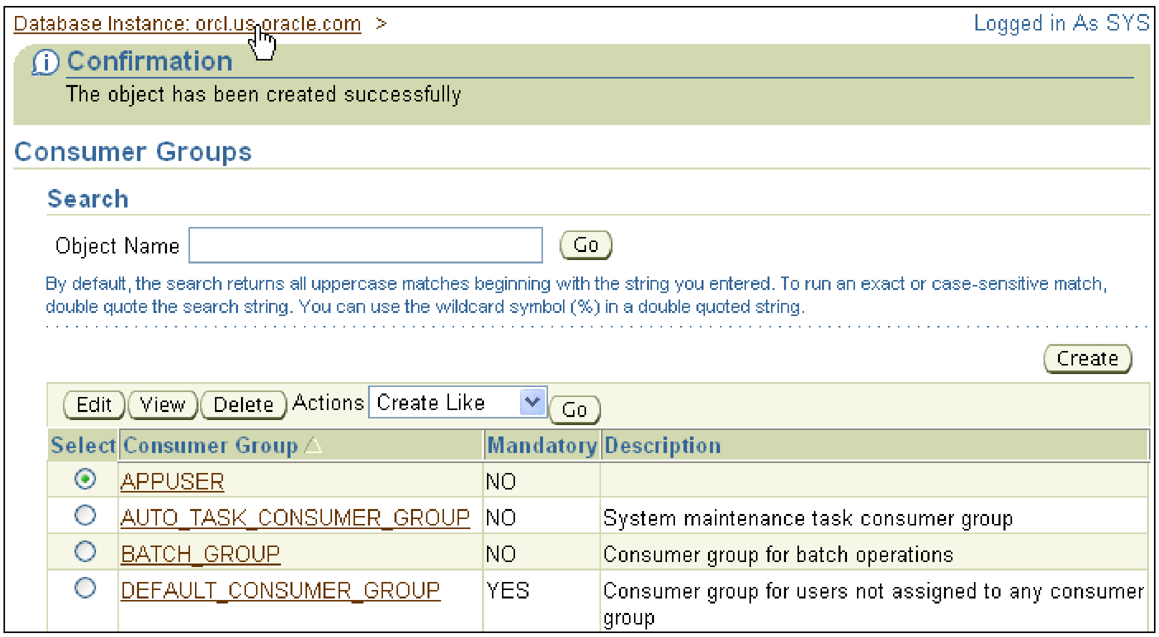
   d) Optionally, click Show SQL, review the statements, and then click Return



The bind variables are not displayed for security reasons.

e) On the Create Resource Consumer Group page, click OK to create the consumer group.

f) A confirmation message appears and the new consumer group is displayed. After confirming its creation, click the “Database Instance:orcl.us.oracle.com” link.



2) Add the APPUSER and LOW_GROUP consumer groups to the DEFAULT_PLAN resource plan. Change the level 3 CPU resource allocation percentages: 60% for the APPUSER consumer group and 40% for the LOW_GROUP consumer group.

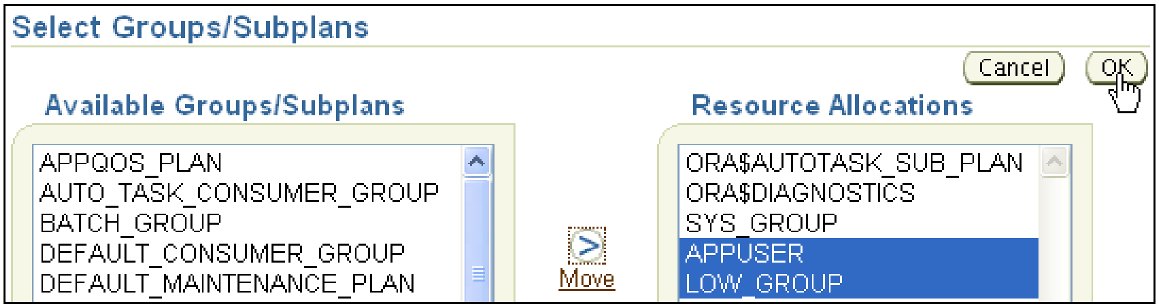
a) In Enterprise Manager, select Server > Plans.

b) On the Resource Plans page, select DEFAULT_PLAN and click the Edit button.

c) Click Modify.

d) On the Select Groups/Subplans page, move APPUSER and LOW_GROUP to the “Resource Allocations.”

Practice 16-1: Managing Resources (continued)



e) Click OK.

f) Enter `60` for `APPUSER` Level 3 and `40` for `LOW_GROUP` Level 3.

g) Click Show SQL.

```
DECLARE
spfileValue VARCHAR2(1000);
scopeValue VARCHAR2(10) := 'MEMORY';
planName VARCHAR2(100) :=?;
BEGIN
dbms_resource_manager.clear_pending_area();
dbms_resource_manager.create_pending_area();
dbms_resource_manager.create_plan_directive(
    plan => ?,
    group_or_subplan => ?,
    comment => ?,
    mgmt_p1 => ?, mgmt_p2 => ?, mgmt_p3 => ?, mgmt_p4 => ?,
    mgmt_p5 => ?, mgmt_p6 => ?, mgmt_p7 => ?, mgmt_p8 => ? ,
    parallel_degree_limit_p1 => ? ,
    switch_io_reqs => ? ,
   switch_io_megabytes => ?
,
    active_sess_pool_p1 => ?,
    queueing_p1 => ?,
    switch_group => ?,
    switch_time => ?,
    switch_estimate => case ? when 'false' then false when
'true' then true else false end,
    max_est_exec_time => ?,
    undo_pool => ? ,
    max_idle_time => ?,
    max_idle_blocker_time => ?,
    switch_for_call => case ? when 'false' then false when
'true' then true else false end

);
dbms_resource_manager.create_plan_directive(
    plan => ?,
    group_or_subplan => ?,
    comment => ?,
    mgmt_p1 => ?, mgmt_p2 => ?, mgmt_p3 => ?, mgmt_p4 => ?,
    mgmt_p5 => ?, mgmt_p6 => ?, mgmt_p7 => ?, mgmt_p8 => ? ,
```

```
    parallel_degree_limit_p1 => ? ,
    switch_io_reqs => ? ,
   switch_io_megabytes => ?
,
    active_sess_pool_p1 => ?,
    queueing_p1 => ?,
    switch_group => ?,
    switch_time => ?,
    switch_estimate => case ? when 'false' then false when
'true' then true else false end,
    max_est_exec_time => ?,
    undo_pool => ? ,
    max_idle_time => ?,
    max_idle_blocker_time => ?,
    switch_for_call => case ? when 'false' then false when
'true' then true else false end

);
dbms_resource_manager.submit_pending_area();
select value into spfileValue from v$parameter where name =
'spfile';
IF spfileValue IS NOT NULL then
EXECUTE IMMEDIATE
'alter system set resource_manager_plan = '||planName||' scope
=BOTH';
END IF;
dbms_resource_manager.switch_plan( plan_name => ? , sid => ? ,
allow_scheduler_plan_switches => FALSE );
END;
```

h) Review the code, then click Return.

Modify

Group/Subplan	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7
APPUSER			60				
LOW_GROUP			40				
ORA$AUTOTASK_SUB_PLAN		5					
ORA$DIAGNOSTICS		5					
OTHER_GROUPS		90					
SYS_GROUP	95						

General | Parallelism | Session Pool | Undo Pool | Threshold | Idle Time

i) On the Edit Resource Plan: DEAFULT_PLAN page, click Apply to assign the APPUSER and the LOW_GROUP consumer groups to the DEFAULT_PLAN resource plan. (You activate this plan later.)

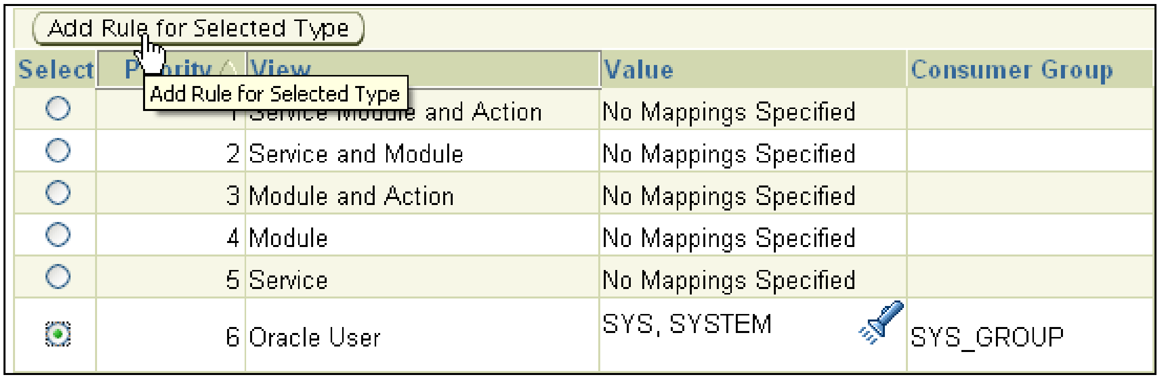
j) You should receive a message, that your update was successful.

Practice 16-1: Managing Resources (continued)

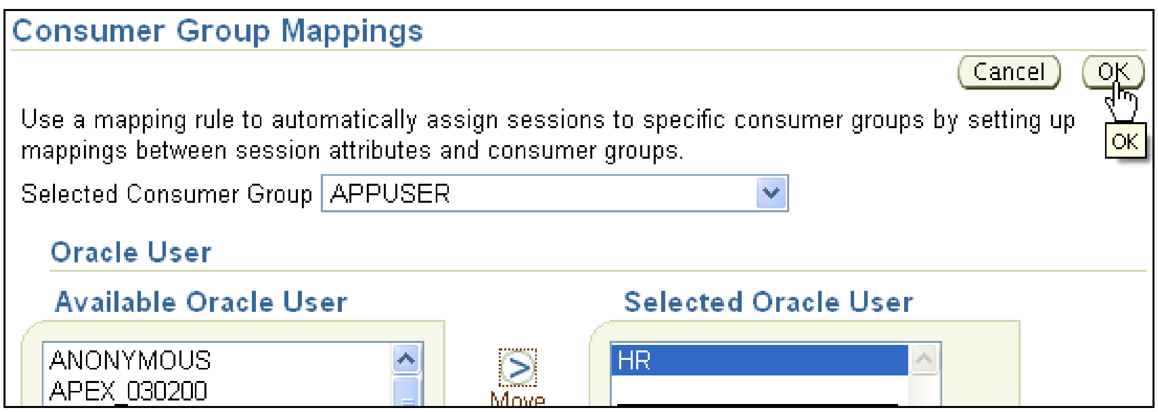
3) Configure Consumer Group Mappings, so that the `HR` Oracle user belongs to the `APPUSER` consumer group, and the `SCOTT` user to the `LOW_GROUP` consumer group. For the `SCOTT` user, confirm that his `ORACLE_USER` attribute has a higher priority than the `CLIENT_OS_USER` attribute.

   a) In Enterprise Manager, select Server > Consumer Group Mappings.

   b) Select Oracle User and click the "Add Rule for Selected Type" button.



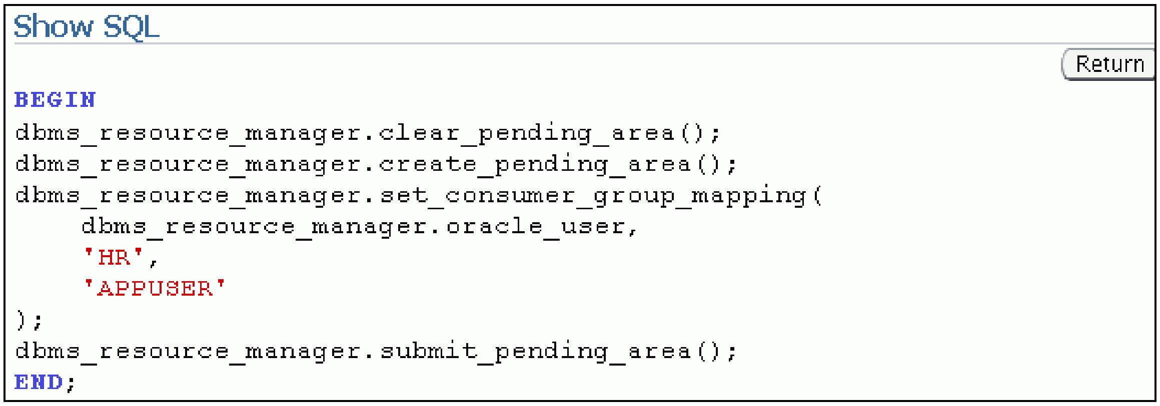
   c) On the Consumer Group Mappings page, ensure that `APPUSER` is selected as "Selected Consumer Group."



   d) Move the `HR` user into the Selected Oracle User region, and then click OK.

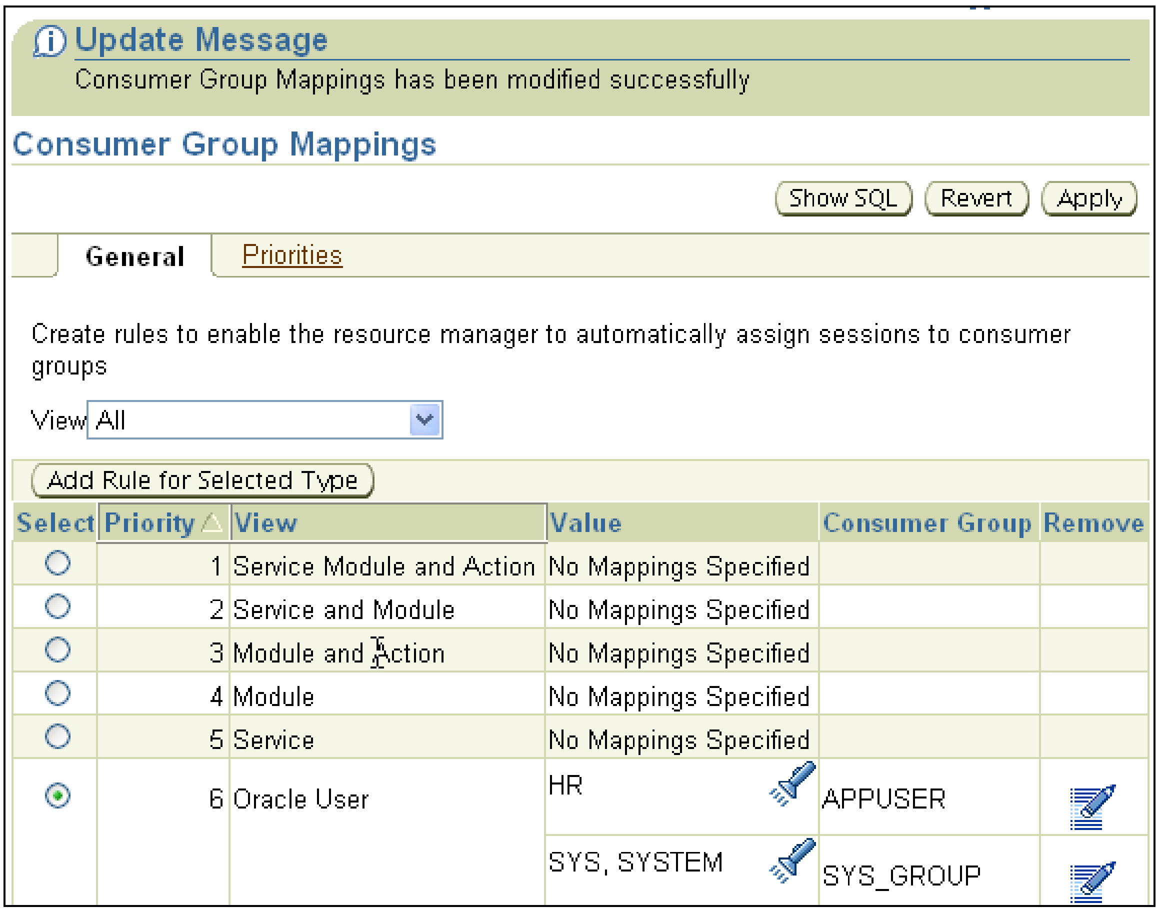
   e) You should receive a success message. On the Consumer Group Mappings General page, click Show SQL.

Practice 16-1: Managing Resources (continued)

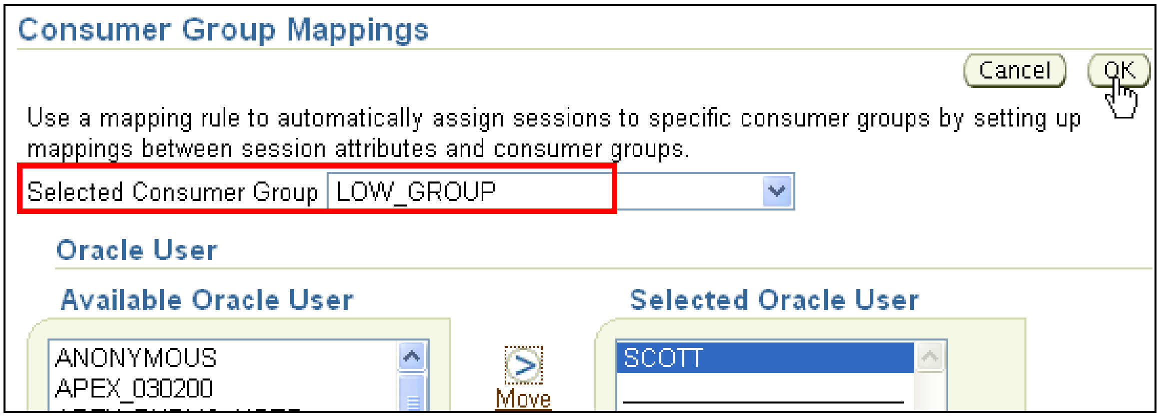


f) Review the statements and click Return.

g) Click Apply to assign the HR user to the APPUSER consumer group.

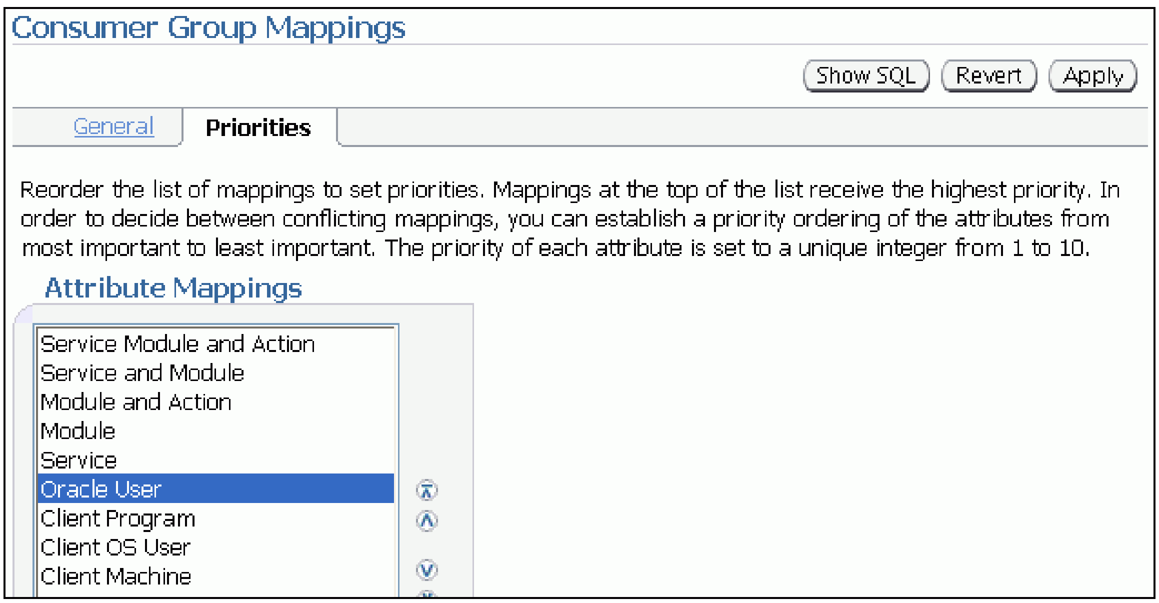


h) Select Oracle User again, and click the “Add Rule for Selected Type” button.



i) Select `LOW_GROUP` from the Selected Consumer Group drop-down list, and move `SCOTT` into the Selected Oracle User area, and then click OK.

j) You should receive a success message (but do not yet apply your changes.) On the Consumer Group Mappings page, click the Priorities tab.



k) Confirm that "Oracle User" has a higher priority than "Client OS User."

l) Click Apply to assign the `SCOTT` user to the `LOW_GROUP` consumer group.

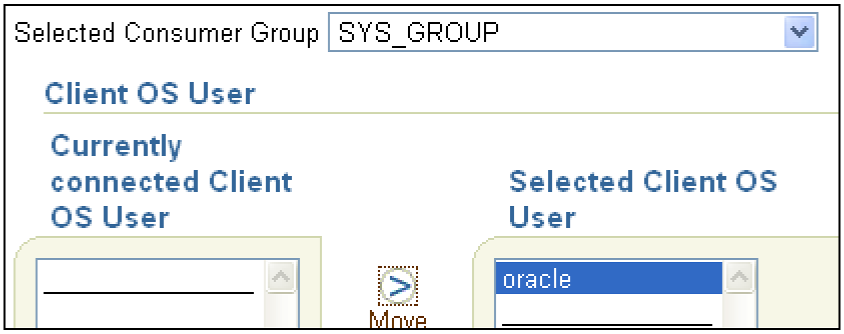
You should receive a success message.

4) Configure Consumer Group Mappings so that the `oracle` OS user belongs to the `SYS_GROUP` consumer group.

   a) Return to the Consumer Group Mappings –General page.

   b) Select Client OS User and click the "Add Rule for Selected Type" button.

Practice 16-1: Managing Resources (continued)



c) Select `SYS_GROUP` from the Selected Consumer Group drop-down list, move `oracle` into the Selected Client OS User area, and click OK.

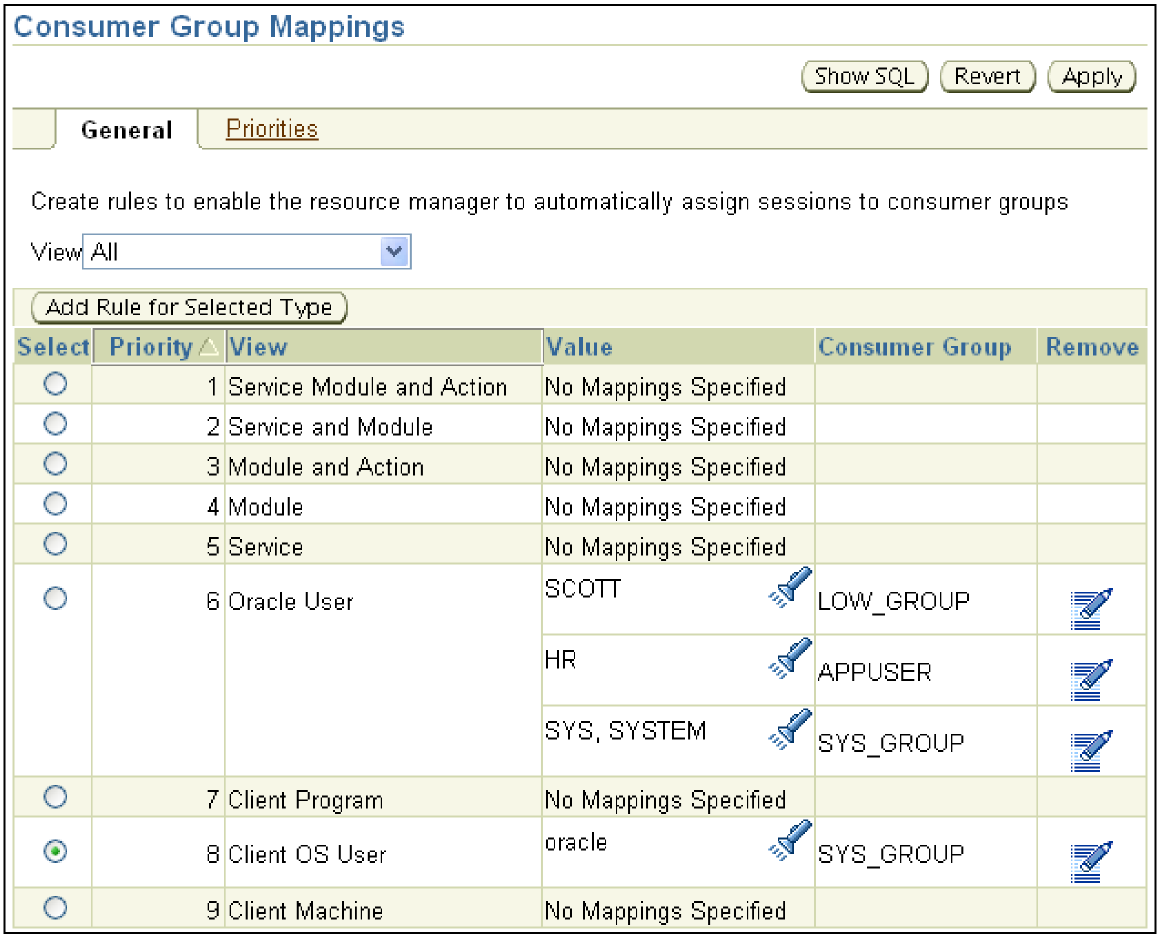
You should receive a success message.

Show SQL

```
BEGIN
dbms_resource_manager.clear_pending_area();
dbms_resource_manager.create_pending_area();
dbms_resource_manager.set_consumer_group_mapping(
    dbms_resource_manager.client_os_user,
    'oracle',
    'SYS_GROUP'
);
dbms_resource_manager.submit_pending_area();
END;
```

d) Optionally, click Show SQL, review the statements, and click Return.

Practice 16-1: Managing Resources (continued)



e) Click Apply to assign the `oracle` OS user to the `SYS_GROUP` consumer group.

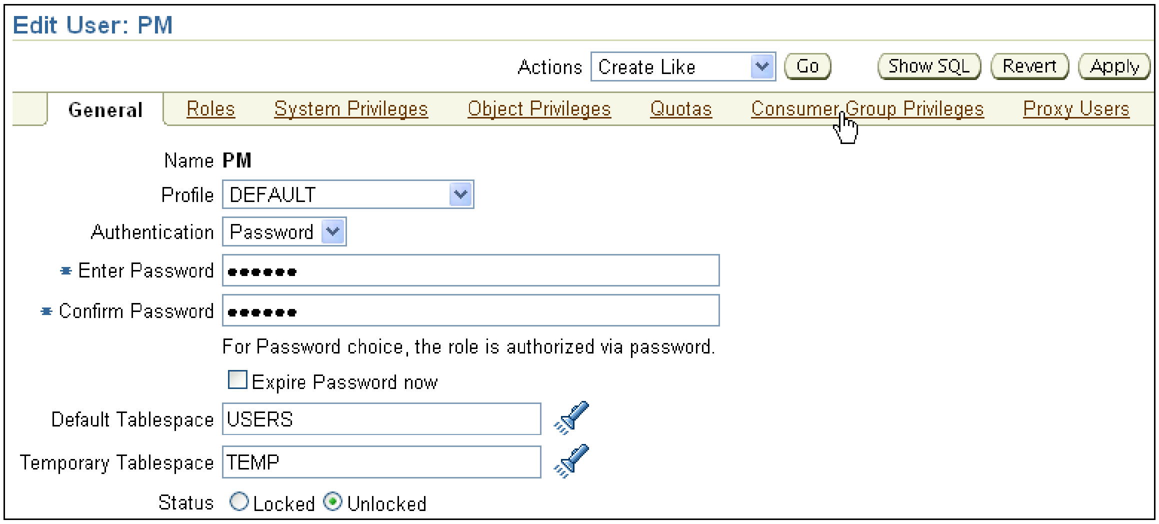
You should receive a success message.

5) Assign the `PM` Oracle user to the following consumer groups: `APPUSER`, `LOW_GROUP`, and `SYS_GROUP`.

a) In Enterprise Manager, select Server > Users (in the Security section).

b) Select the `PM` user and click the Edit button.

Practice 16-1: Managing Resources (continued)

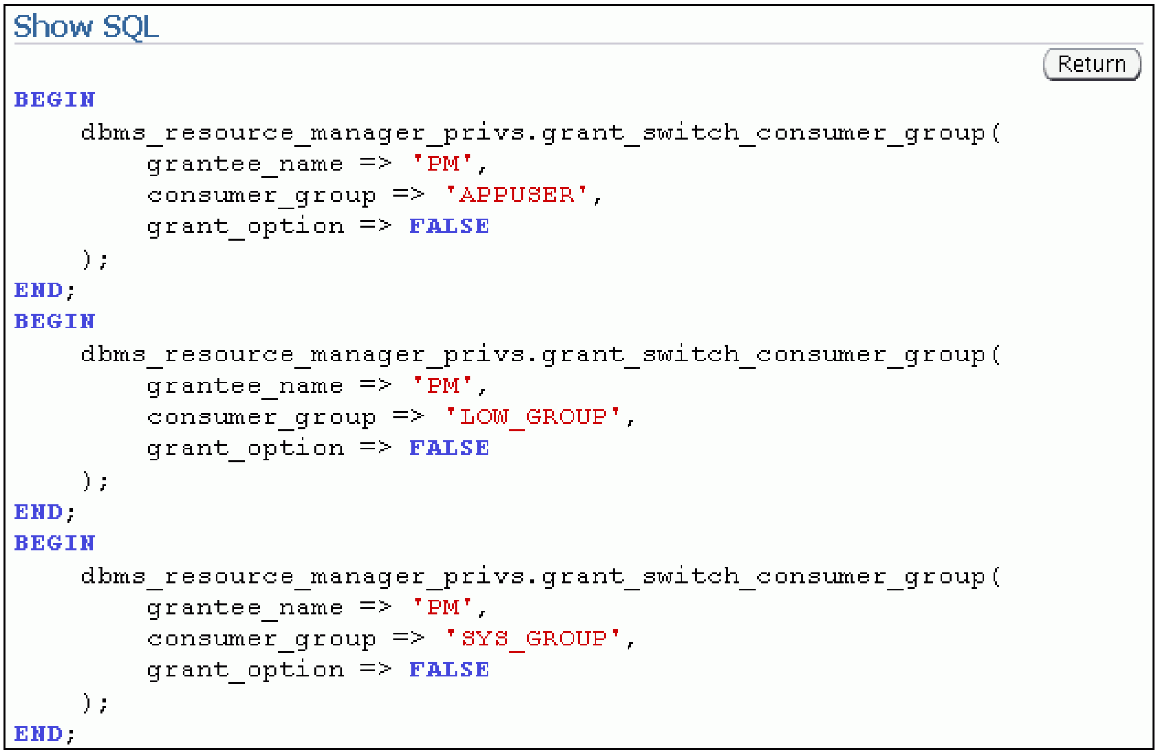


c) Click the Consumer Groups Privileges tab. If you see an error regarding the password for the `PM` user, enter `oracle_4U` in both the password fields.

d) Click the Edit List button.

e) Move the `APPUSER`, `LOW_GROUP`, and `SYS_GROUP` consumer groups to Selected Consumer Groups and click OK.

f) Click Show SQL.



Show SQL

Return

```
BEGIN
    dbms_resource_manager_privs.grant_switch_consumer_group(
        grantee_name => 'PM',
        consumer_group => 'APPUSER',
        grant_option => FALSE
    );
END;
BEGIN
    dbms_resource_manager_privs.grant_switch_consumer_group(
        grantee_name => 'PM',
        consumer_group => 'LOW_GROUP',
        grant_option => FALSE
    );
END;
BEGIN
    dbms_resource_manager_privs.grant_switch_consumer_group(
        grantee_name => 'PM',
        consumer_group => 'SYS_GROUP',
        grant_option => FALSE
    );
END;
```

g) Review the statements and click Return.

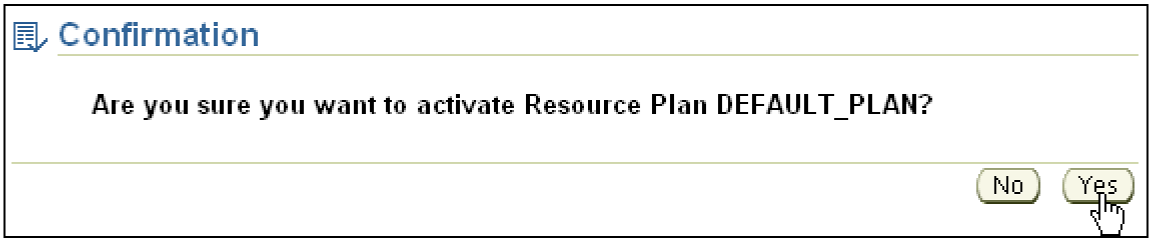
h) Click Apply to assign the `PM` user to these consumer groups.

You should receive a success message.

6) Activate the `DEFAULT_PLAN` resource plan.

a) In Enterprise Manager, select Server > Plans.

b) On the Resource Plans page, select `DEFAULT_PLAN`, select Activate from the Actions drop-down list, and click Go.



c) Click Yes to confirm your activation.

You should see a success message.

7) Test the consumer group mappings. Start two SQL*Plus sessions: the first with the `system/oracle_4U@orcl` connect string and the second with the `scott/oracle_4U@orcl` connect string.

a) As the `oracle` user in a terminal window, execute the `oraenv` script to set environment variables for the `orcl` database.

Practice 16-1: Managing Resources (continued)

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
$
```

Your output may be different depending on your previously executed tasks.

b) To start a SQL*Plus session with the `system/oracle_4U@orcl` connect string and to set your SQL prompt to "FIRST," enter:

```
$ sqlplus system@orcl
Enter password: oracle_4U <<< not displayed

SQL> SET SQLPROMPT "FIRST>"
FIRST>
```

c) As the `oracle` user in a terminal window, execute the `oraenv` script to set environment variables for the `orcl` database.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
$
```

Your output may be different depending on your previously executed tasks.

d) To start a SQL*Plus session with the `scott/oracle_4U@orcl` connect string and to set your SQL prompt to "SECOND," enter:

```
$ sqlplus scott@orcl
Enter password: oracle_4U <<< not displayed

SQL> SET SQLPROMPT "SECOND>"
SECOND>
```

e) In your `FIRST` SQL*Plus session, enter:

```
FIRST>@query_rsc_groups.sql

SCHEMANAME                     RESOURCE_CONSUMER_GROUP
------------------------------ ------------------------------
--
DBSNMP                         OTHER_GROUPS
DBSNMP                         OTHER_GROUPS
DBSNMP                         OTHER_GROUPS
SCOTT                          LOW_GROUP

FIRST>
```

f) ***Question:*** To which consumer group does the `SCOTT` user belong?

Answer: `SCOTT` is in the `LOW_GROUP` consumer group.

Note: Your output for this step (and the following steps) may not look exactly like the output shown. The information of concern here is for the specific users being mentioned.

Practice 16-1: Managing Resources (continued)

g) In the SECOND terminal window, connect as the PM user with the oracle_4U password:

```
SECOND>connect pm@orcl
Enter password: oracle_4U <<< not displayed

Connected.
SECOND>
```

h) In your FIRST SQL*Plus session, enter "/" to execute the previous SQL statement again.

```
FIRST>/

SCHEMANAME                     RESOURCE_CONSUMER_GROUP
------------------------------ ------------------------------
-
DBSNMP                         OTHER_GROUPS
DBSNMP                         OTHER_GROUPS
DBSNMP                         OTHER_GROUPS
PM                             SYS_GROUP

FIRST>
```

i) ***Question:*** To which consumer group does the PM user belong?

Answer: PM is in the SYS_GROUP consumer group.

j) In the SECOND terminal window, connect as the OE user with the oracle_4U password::

```
SECOND>connect oe@orcl
Enter password: oracle_4U <<< not displayed

Connected.
SECOND>
```

k) In your FIRST SQL*Plus session, enter "/" to execute the previous SQL statement again.

```
FIRST>/

SCHEMANAME                     RESOURCE_CONSUMER_GROUP
------------------------------ ------------------------------
DBSNMP                         OTHER_GROUPS
DBSNMP                         OTHER_GROUPS
DBSNMP                         OTHER_GROUPS
OE                             OTHER_GROUPS

FIRST> exit
```

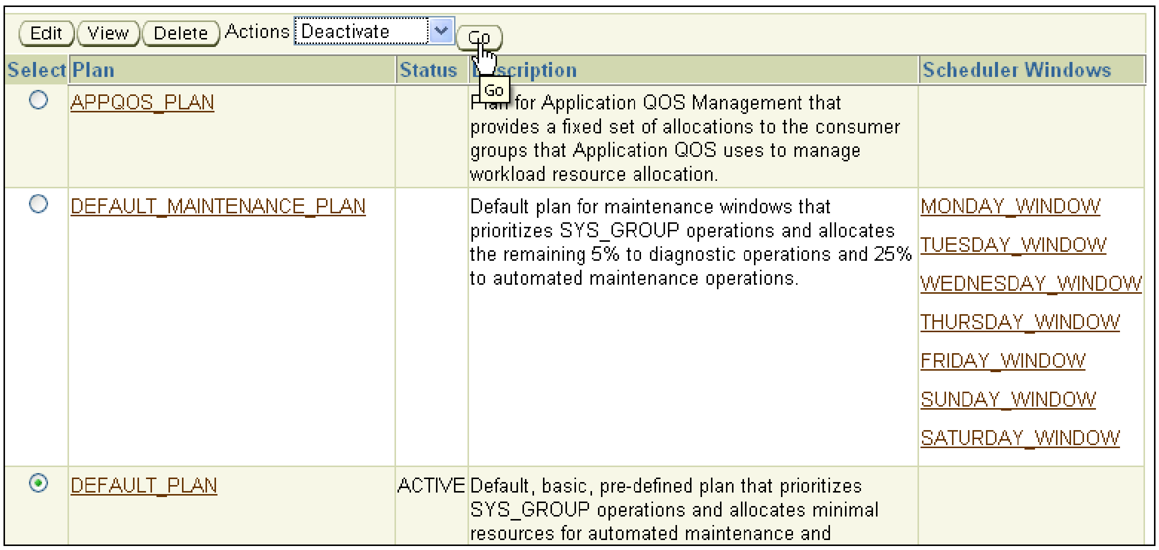
l) Exit both the SQL*Plus sessions.

m) *Question:* When testing your `OE` Oracle user, you notice that `OE` is in the `OTHER_GROUPS` consumer group. Why is that?

Possible Answer: The `OE` user is not explicitly assigned to another consumer resource group.

8) Revert to your original configuration by deactivating the `DEFAULT_PLAN` resource group, locking accounts, undoing all consumer group mappings, and finally by deleting the `APPUSER` resource group.

a) To deactivate the `DEFAULT_PLAN` resource plan in Enterprise Manager, select Server > Plans.



b) On the Resource Plans page, select the `DEFAULT_PLAN`, select Deactivate from the Actions drop-down list, and click Go.

c) Click Yes to confirm your deactivation.

You should receive a success message.

d) To reconfigure or undo all consumer group mappings, review and execute the `rsc_cleanup.sh` script from your working directory:

```
$ cat rsc_cleanup.sh
# Oracle Database 11g: Administration Workshop II
# Oracle Server Technologies - Curriculum Development
#
# ***Training purposes only***
# ***Not appropriate for production use***
#
# This script supports the Resource Manager practice session.
#   Start this script connected as OS user: oracle.

sqlplus "/ as sysdba" << EOF
```

```
PROMPT undo lab step 5

BEGIN
    dbms_resource_manager_privs.revoke_switch_consumer_group(
        revokee_name => 'PM',
        consumer_group => 'APPUSER'
    );
END;
/
BEGIN
    dbms_resource_manager_privs.revoke_switch_consumer_group(
        revokee_name => 'PM',
        consumer_group => 'LOW_GROUP'
    );
END;
/
BEGIN
    dbms_resource_manager_privs.revoke_switch_consumer_group(
        revokee_name => 'PM',
        consumer_group => 'SYS_GROUP'
    );
END;
/
PROMPT undo lab step 4

BEGIN
dbms_resource_manager.clear_pending_area();
dbms_resource_manager.create_pending_area();
dbms_resource_manager.set_consumer_group_mapping(
    dbms_resource_manager.client_os_user,
    'ORACLE',
    NULL
);
dbms_resource_manager.submit_pending_area();
END;
/
PROMPT  undo lab step 3

BEGIN
dbms_resource_manager.clear_pending_area();
dbms_resource_manager.create_pending_area();
dbms_resource_manager.set_consumer_group_mapping(
    dbms_resource_manager.oracle_user,
    'HR',
    NULL
);
dbms_resource_manager.set_consumer_group_mapping(
    dbms_resource_manager.oracle_user,
    'SCOTT',
    NULL
);
```

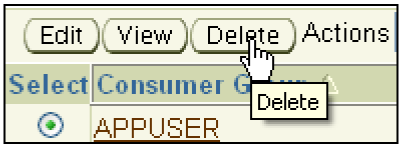
Practice 16-1: Managing Resources (continued)

```
dbms_resource_manager.set_consumer_group_mapping(
    dbms_resource_manager.oracle_user,
    'SYS',
    NULL
);
dbms_resource_manager.set_consumer_group_mapping(
    dbms_resource_manager.oracle_user,
    'SYSTEM',
    NULL
);
dbms_resource_manager.submit_pending_area();
END;
/
PROMPT  undo lab step 2

BEGIN
dbms_resource_manager.clear_pending_area();
dbms_resource_manager.create_pending_area();
dbms_resource_manager.delete_plan_directive('DEFAULT_PLAN',
'APPUSER');
dbms_resource_manager.delete_plan_directive('DEFAULT_PLAN',
'LOW_GROUP');
dbms_resource_manager.submit_pending_area();
dbms_resource_manager.switch_plan( plan_name => '', sid =>
'orcl', allow_scheduler_plan_switches => FALSE );
END;
/
exit
EOF
```

```
$ ./rsc_cleanup.sh

SQL> SQL> SQL> undo lab step 5
SQL> SQL>   2    3    4    5    6    7
PL/SQL procedure successfully completed.

SQL>   2    3    4    5    6    7
PL/SQL procedure successfully completed.

SQL>   2    3    4    5    6    7
PL/SQL procedure successfully completed.

SQL> undo lab step 4
SQL> SQL>   2    3    4    5    6    7    8    9   10   11
PL/SQL procedure successfully completed.

SQL> undo lab step 3
SQL> SQL>   2    3    4    5    6    7    8    9   10   11
12   13   14   15   16   17   18   19   20   21   22   23   24
25   26
```

```
PL/SQL procedure successfully completed.

SQL> undo lab step 2
SQL> SQL>   2    3    4    5    6    7    8    9
PL/SQL procedure successfully completed.

$
```

e) To delete the APPUSER resource group in Enterprise Manager, select Server > Consumer Groups.



f) On the Consumer Groups page, select APPUSER and click the Delete button.

g) Confirm your deletion by clicking Yes.

You should receive a success message.

h) Exit Enterprise Manager and close all terminal windows.

Practices for Lesson 17

In these practices, you explore Oracle Scheduler capabilities.

Practice 17-1: Creating Scheduler Components

In this practice, you use Enterprise Manager Database Control to create Scheduler objects in the `ORCL` database instance and automate tasks.

While logged in to the database as the HR user in Database Control, create a simple job that runs a SQL script:

- General:

  Name: `CREATE_LOG_TABLE_JOB`
  Owner: `HR`
  Description: Create the `SESSION_HISTORY` table for the next part of this practice
  Logging Level: `RUNS`
  Command Type: PL/SQL
  PL/SQL Block: `BEGIN execute immediate('create table session_history(snap_time TIMESTAMP WITH LOCAL TIME ZONE, num_sessions NUMBER)'); END;`

- Schedule:

  Repeating: `Do not Repeat`
  Start: `Immediately`

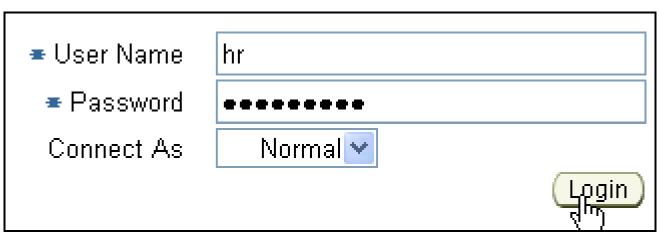
- Options:

  No special options

1) As the `SYS` user, grant `CONNECT`, `RESOURCE`, and `DBA` roles to the `HR` user.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl

$ sqlplus / as sysdba
Connected.
SQL> grant connect, resource, dba to hr;
Grant succeeded.
```

2) Log in to Enterprise Manager Database Control as the `HR` user with the `oracle_4U` password.



3) To create a job, navigate to Server > Jobs (in the Oracle Scheduler region).

   a) On the Scheduler Jobs page, click the Create button.

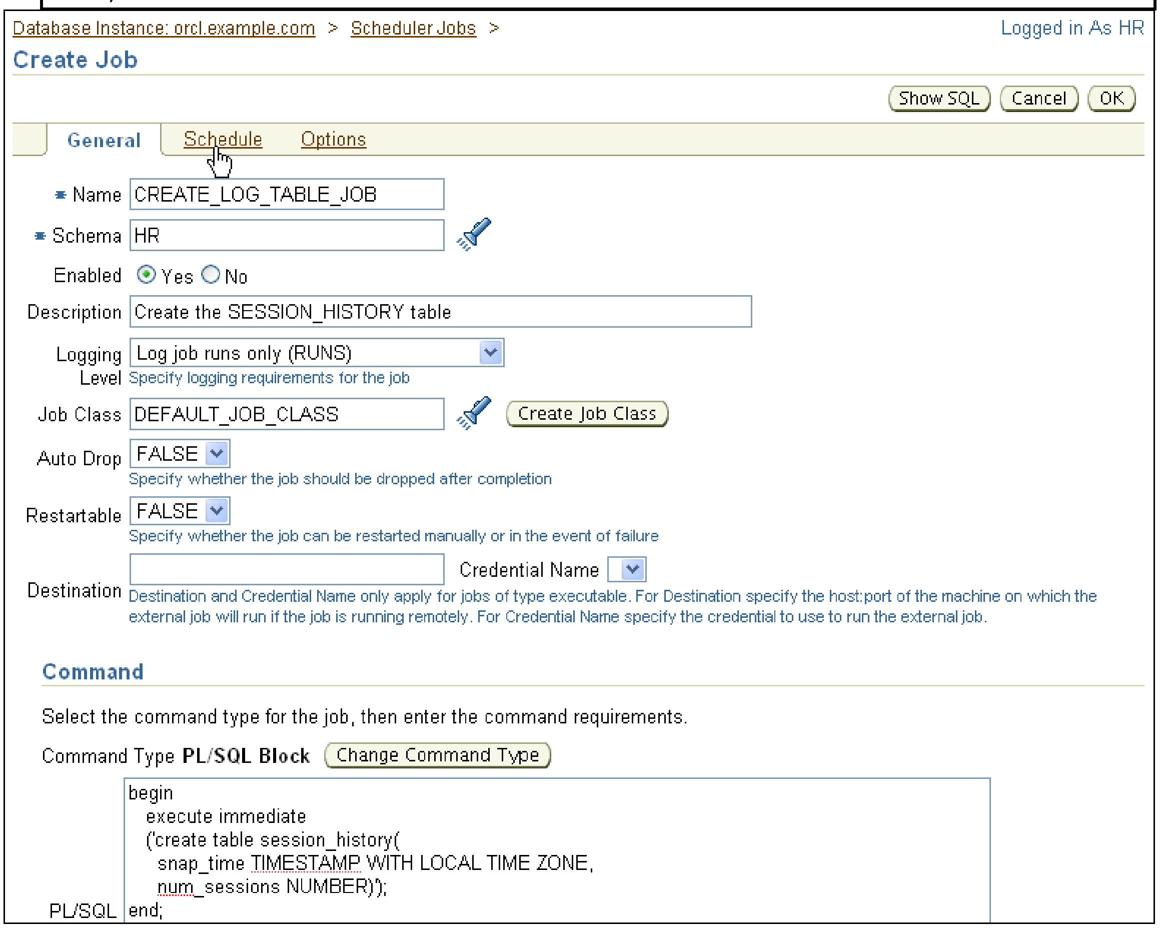
   b) On the Create Job - General page, enter and confirm the following values:

      Name: `CREATE_LOG_TABLE_JOB`
      Schema: `HR`

Practice 17-1: Creating Scheduler Components (continued)

Enabled: `Yes`
Description: Create the `SESSION_HISTORY` table
Logging Level: `Log job runs only (RUNS)`
Command Type: PL/SQL Block
PL/SQL Block:

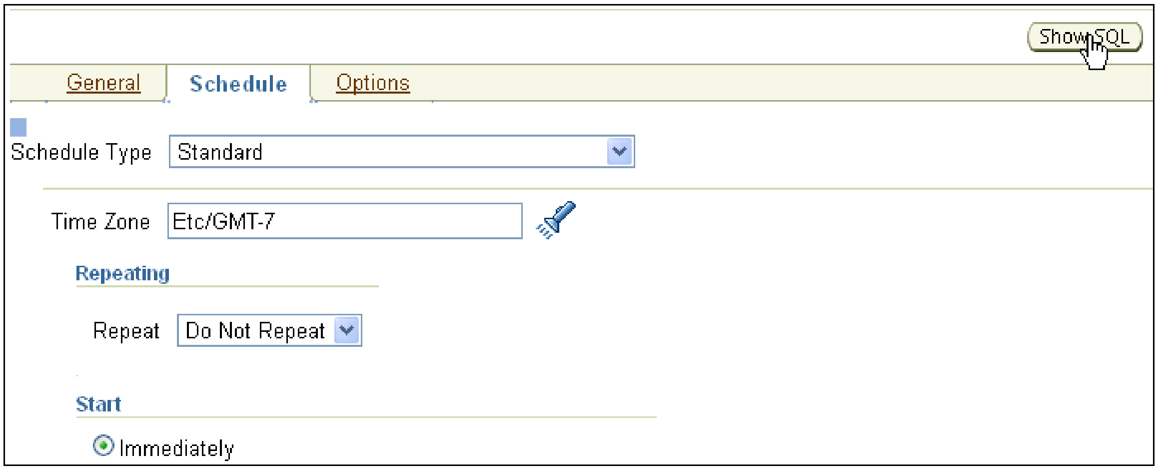
```
begin
   execute immediate
   ('create table session_history(
     snap_time TIMESTAMP WITH LOCAL TIME ZONE,
     num_sessions NUMBER)');
end;
```



c) On the Schedule folder tab, enter and confirm the following values:

Timezone: *Your_local_timezone*
Repeating: `Do not Repeat`
Start: `Immediately`

Practice 17-1: Creating Scheduler Components (continued)

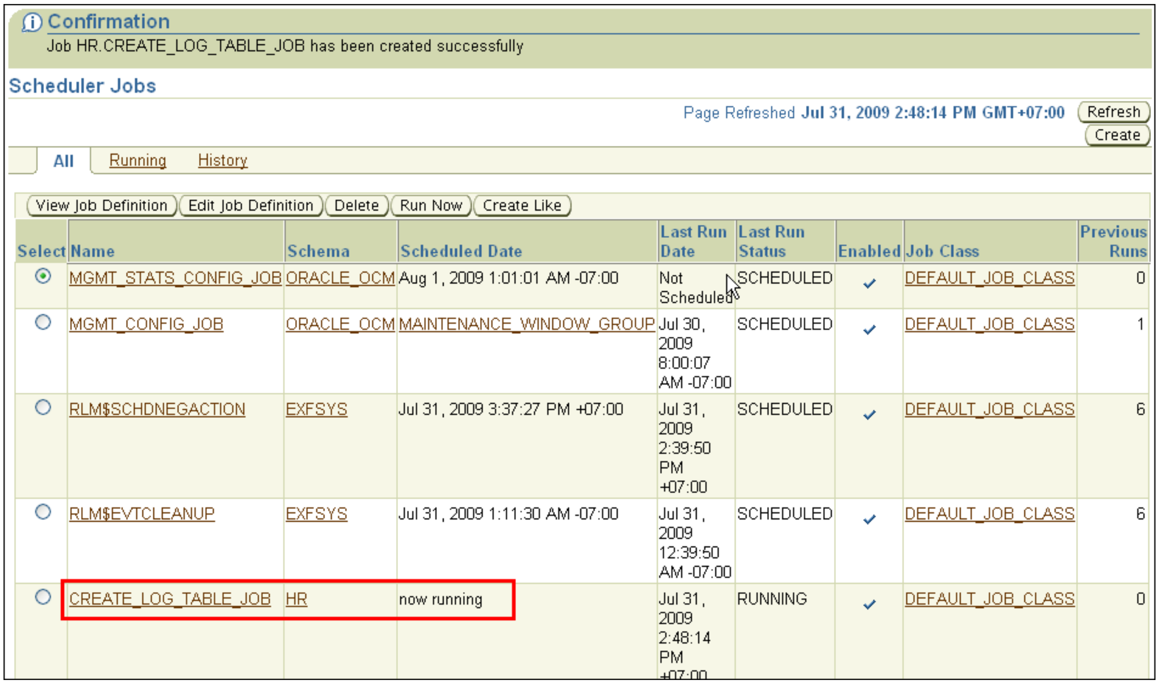


d) Click Show SQL if you want to view the SQL statement defining your job.

```
BEGIN
sys.dbms_scheduler.create_job(
job_name => '"HR"."CREATE_LOG_TABLE_JOB"',
job_type => 'PLSQL_BLOCK',
job_action => 'begin
   execute immediate
   (''create table session_history(
     snap_time TIMESTAMP WITH LOCAL TIME ZONE,
     num_sessions NUMBER)'');
end;
',
start_date => systimestamp at time zone 'Etc/GMT-7',
job_class => '"DEFAULT_JOB_CLASS"',
comments => 'Create the SESSION_HISTORY table',
auto_drop => FALSE,
enabled => TRUE);
END;
```

e) Review the statements (Your time zone might be different.) and click Return.

f) Click OK to create the job.

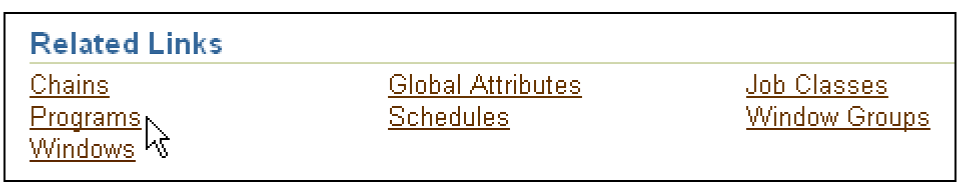


g) If the job does not appear on the Scheduler Jobs page, click the Refresh button until it succeeds. – Also, you may not see it “running”, but already with the Last Run Status of `SUCCEEDED`.

4) Create a program called `LOG_SESS_COUNT_PRGM` that logs the current number of database sessions into a table. Use the following code:

```
DECLARE
sess_count   NUMBER;
BEGIN
SELECT COUNT(*) INTO sess_count FROM V$SESSION;
INSERT INTO session_history VALUES (systimestamp,
sess_count);
COMMIT;
END;
```

a) Logged into Enterprise Manager as the HR user, naviagate to Server > Programs or click the Programs in the Related Links section on the Scheduler Jobs page.



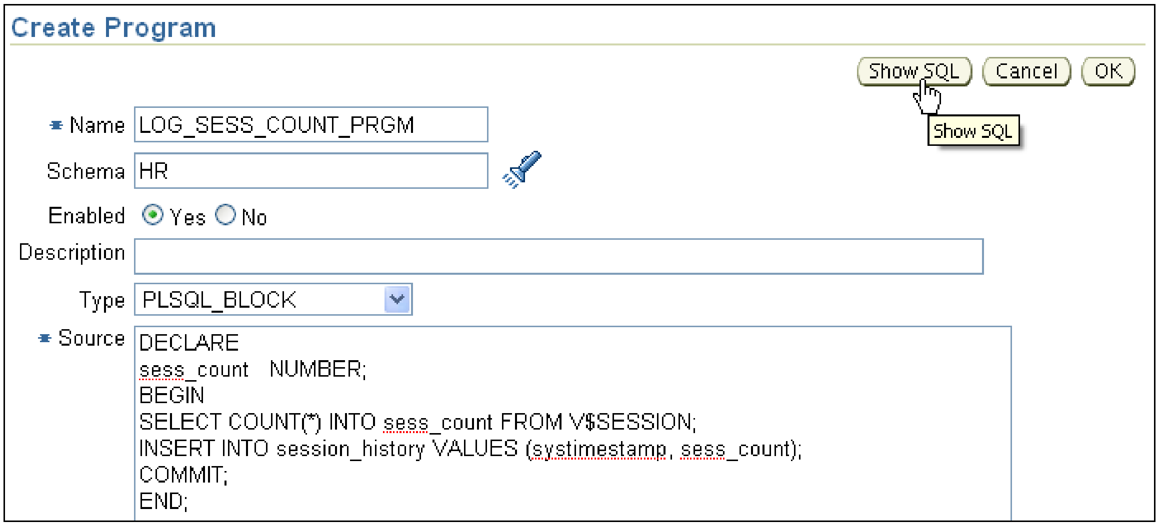
b) On the Scheduler Programs page, click the Create button.

c) On the Create Program page, enter and confirm the following values:

Name: `LOG_SESS_COUNT_PRGM`
Schema: `HR`
Enabled: Yes

Type: PLSQL_BLOCK
Source:

```
DECLARE
sess_count   NUMBER;
BEGIN
SELECT COUNT(*) INTO sess_count FROM V$SESSION;
INSERT INTO session_history VALUES (systimestamp,
sess_count);
COMMIT;
END;
```

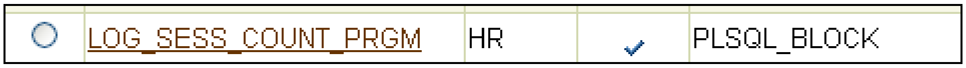


d) Click Show SQL.

```
BEGIN
DBMS_SCHEDULER.CREATE_PROGRAM(
program_name=>'"HR"."LOG_SESS_COUNT_PRGM"',
program_action=>'DECLARE
sess_count   NUMBER;
BEGIN
SELECT COUNT(*) INTO sess_count FROM V$SESSION;
INSERT INTO session_history VALUES (systimestamp, sess_count);
COMMIT;
END;
',
program_type=>'PLSQL_BLOCK',
number_of_arguments=>0,
comments=>'',
enabled=>TRUE);
END;
```

e) Review the statements, and then click Return.

f) Click OK to create the program.



○	LOG_SESS_COUNT_PRGM	HR	✓	PLSQL_BLOCK

You should see the program on the Scheduler Programs page.

5) Create a schedule named SESS_UPDATE_SCHED owned by HR that executes every three seconds. Use SQL*Plus and the DBMS_SCHEDULER.CREATE_SCHEDULE procedure to create the schedule.

```
BEGIN
  DBMS_SCHEDULER.CREATE_SCHEDULE (
   schedule_name => 'SESS_UPDATE_SCHED',
   start_date => SYSTIMESTAMP,
   repeat_interval => 'FREQ=SECONDLY;INTERVAL=3',
   comments => 'Every three seconds');
END;
/
```

Return to Enterprise Manager Database Control and verify that the SESS_UPDATE_SCHED schedule was created.

Hint: You may have to refresh the page for the Schedule to appear.

a) In a terminal window, enter:

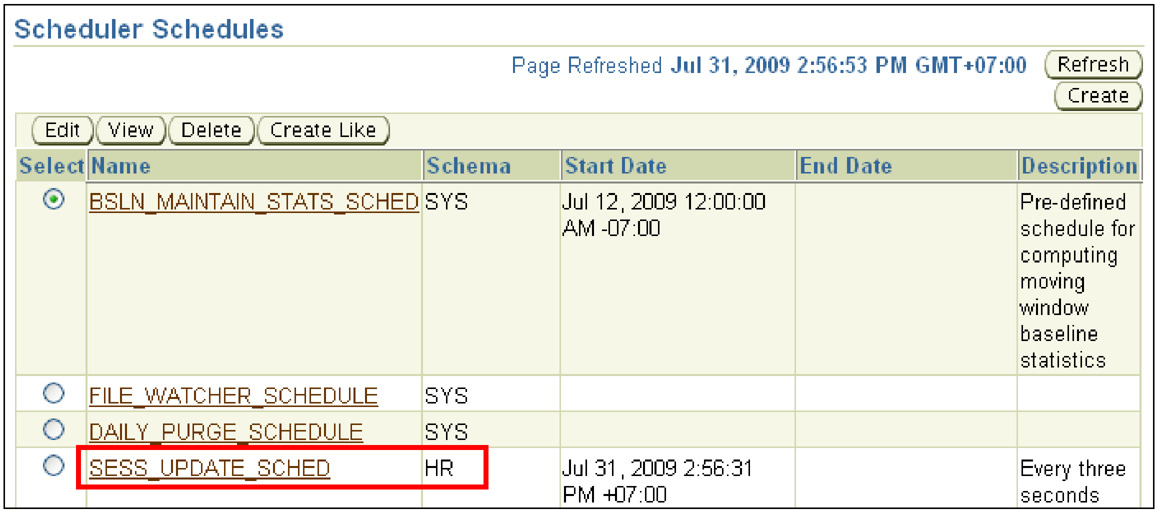
```
$ sqlplus hr
Enter password: oracle_4U <<< not displayed
```

b) In your SQL*Plus session, enter:

```
BEGIN
  DBMS_SCHEDULER.CREATE_SCHEDULE (
   schedule_name => 'SESS_UPDATE_SCHED',
   start_date => SYSTIMESTAMP,
   repeat_interval => 'FREQ=SECONDLY;INTERVAL=3',
   comments => 'Every three seconds');
END;
/
PL/SQL procedure successfully completed.
```

c) In Enterprise Manager, select Server > Schedules.

d) Verify that the SESS_UPDATE_SCHED schedule has been created. (You may have to refresh the page for the Schedule to appear.)



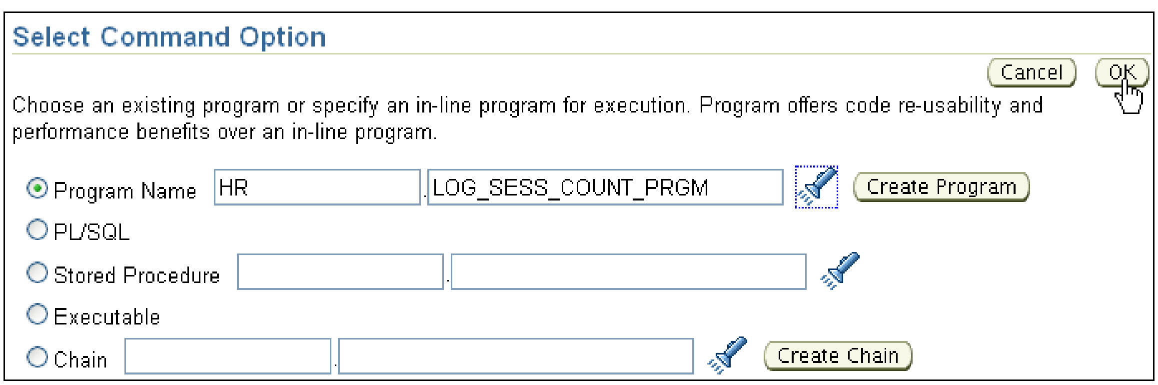
6) Using Enterprise Manager Database Control, create a job named LOG_SESSIONS_JOB that uses the LOG_SESS_COUNT_PRGM program and the SESS_UPDATE_SCHED schedule. Make sure that the job uses FULL logging.

a) In Enterprise Manager, select Server > Jobs, and then click the Create button.

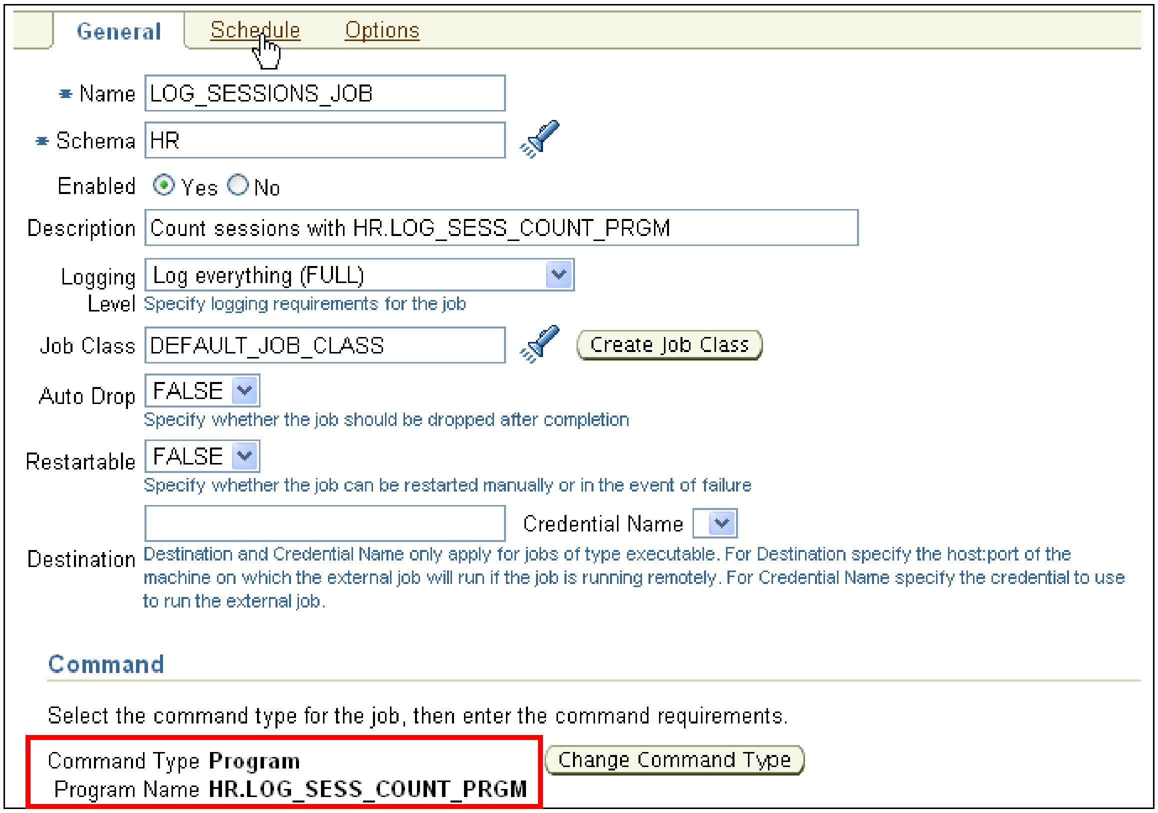
b) On the Create Job page, enter and confirm the following values:

Name: `LOG_SESSIONS_JOB`
Owner: `HR`
`Enabled: Yes`
Description: Count sessions with HR.LOG_SESS_COUNT_PRGM
Logging level: `Log everything (FULL)`

c) Click Change Command Type, and on the Select Command Option page, select Program Name, and enter `HR.LOG_SESS_COUNT_PRGM` in the field next to it, or use the Lookup (flashlight) icon to select the program.

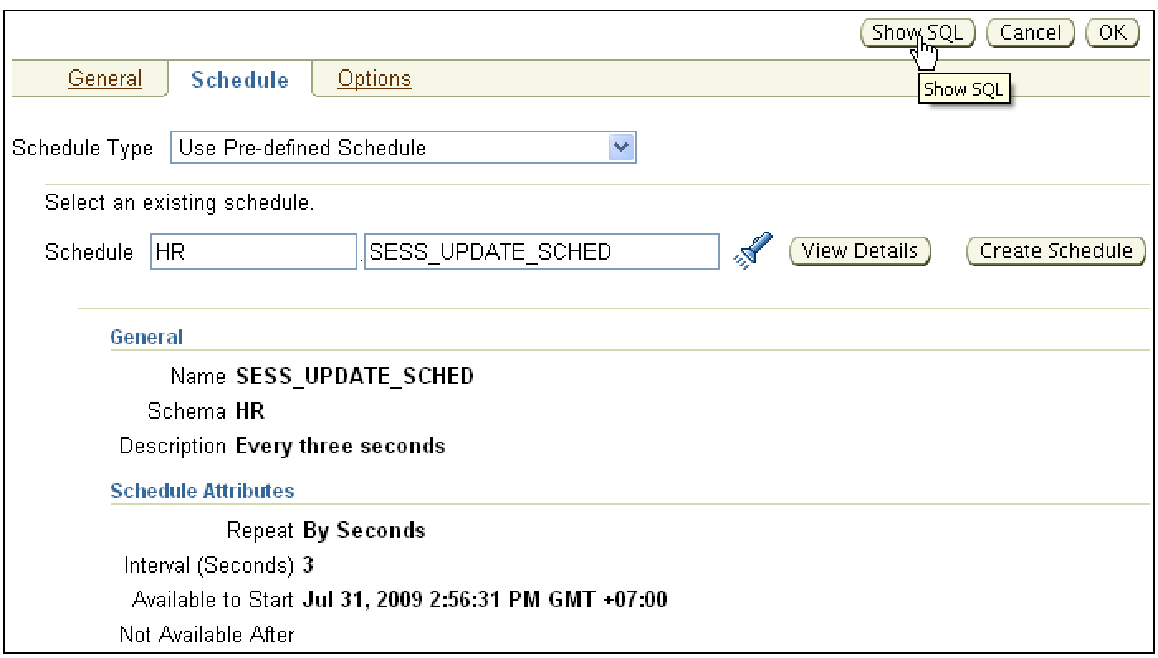


d) Click OK.



e) Back on the Create Job page, click the Schedule tab.

f) Change the Schedule Type to "Use Pre-Defined Schedule," and select the `HR.SESS_UPDATE_SCHED` schedule by using the flashlight icon.



g) Click Show SQL.

Practice 17-1: Creating Scheduler Components (continued)

```
BEGIN
sys.dbms_scheduler.create_job(
job_name => '"HR"."LOG_SESSIONS_JOB"',
program_name => '"HR"."LOG_SESS_COUNT_PRGM"',
schedule_name => '"HR"."SESS_UPDATE_SCHED"',
job_class => '"DEFAULT_JOB_CLASS"',
comments => 'Count sessions with HR.LOG_SESS_COUNT_PRGM',
auto_drop => FALSE,
enabled => FALSE);
sys.dbms_scheduler.set_attribute( name => '"HR"."LOG_SESSIONS_JOB"',
attribute => 'logging_level', value => DBMS_SCHEDULER.LOGGING_FULL);
sys.dbms_scheduler.enable( '"HR"."LOG_SESSIONS_JOB"' );
END;
```

h) Review the statements and then click Return.

i) On the Create Job page, click OK to create the job.

You should receive a success message and see the job on the Scheduler Jobs page.

Select	Name	Schema	Scheduled Date	Last Run Date	Last Run Status	Enabled	Job Class	Previous Runs
	MGMT_STATS_CONFIG_JOB	ORACLE_OCM	Aug 1, 2009 1:01:01 AM -07:00	Not Scheduled	SCHEDULED	✓	DEFAULT_JOB_CLASS	0
	MGMT_CONFIG_JOB	ORACLE_OCM	MAINTENANCE_WINDOW_GROUP	Jul 30, 2009 8:00:07 AM -07:00	SCHEDULED	✓	DEFAULT_JOB_CLASS	1
	RLM$SCHDNEGACTION	EXFSYS	Jul 31, 2009 3:37:27 PM +07:00	Jul 31, 2009 2:39:50 PM +07:00	SCHEDULED	✓	DEFAULT_JOB_CLASS	6
	RLM$EVTCLEANUP	EXFSYS	Jul 31, 2009 1:11:30 AM -07:00	Jul 31, 2009 12:39:50 AM -07:00	SCHEDULED	✓	DEFAULT_JOB_CLASS	6
	LOG_SESSIONS_JOB	HR	Jul 31, 2009 3:05:34 PM +07:00	Jul 31, 2009 3:05:31 PM +07:00	SCHEDULED	✓	DEFAULT_JOB_CLASS	29

Note that it quicly accumulates previous runs, because it executes every three seconds.

7) In your SQL*Plus session, check the `HR.SESSION_HISTORY` table for rows.

a) Enter:

```
SQL> SELECT * FROM SESSION_HISTORY ORDER BY snap_time;
```

Your result looks different but the second values should be three seconds apart:

```
SNAP_TIME
---------------------------------------------------------------
NUM_SESSIONS
------------
31-JUL-09 03.07.55.101299 PM
          41
```

Practice 17-1: Creating Scheduler Components (continued)

```
31-JUL-09 03.07.58.099194 PM
             41
```

Question: If there are rows in the table, are the time stamps three seconds apart?

Answer: Yes, there are rows. Yes, the time stamps are three seconds apart.

8) Use Enterprise Manager Database Control to alter the SESS_UPDATE_SCHED schedule from every three seconds to every three minutes. Then use SQL*Plus to verify that the rows are now being added every three minutes: query the HR.SESSION_HISTORY table, ordered by the SNAP_TIME column.

   a) In Enterprise Manager, select Server > Schedules.

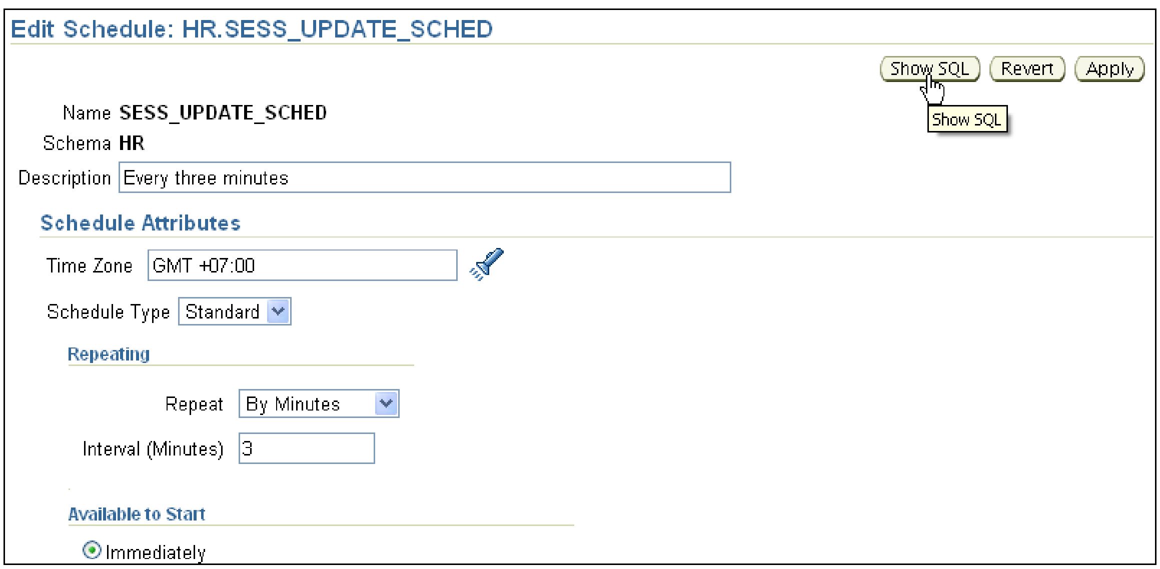
   b) Click the `SESS_UPDATE_SCHED` link.

   c) On the View Schedule page, click Edit.

   d) Change the description to "Every three minutes."

   e) Change Available to Start to Immediately.

   f) Change the value in the Repeat drop-down list from By Seconds to **By Minutes**.



   g) Ensure that the interval is 3 and then click Show SQL.

```
BEGIN
sys.dbms_scheduler.set_attribute( name => '"HR"."SESS_UPDATE_SCHED"', attribute =>
'repeat_interval', value => 'FREQ=MINUTELY;INTERVAL=3');
sys.dbms_scheduler.set_attribute( name => '"HR"."SESS_UPDATE_SCHED"', attribute =>
'start_date', value => systimestamp at time zone '+7:00');
sys.dbms_scheduler.set_attribute( name => '"HR"."SESS_UPDATE_SCHED"', attribute =>
'comments', value => 'Every three minutes');
END;
```

h) Review the statements, click Return, and then click Apply.

You should receive a success message.

9) In your SQL*Plus session, query the `HR.SESSION_HISTORY` table, ordered by the `SNAP_TIME` column. (Wait for three minutes after you update the schedule.) Enter:

```
SQL> SELECT * FROM HR.SESSION_HISTORY ORDER BY snap_time;
```

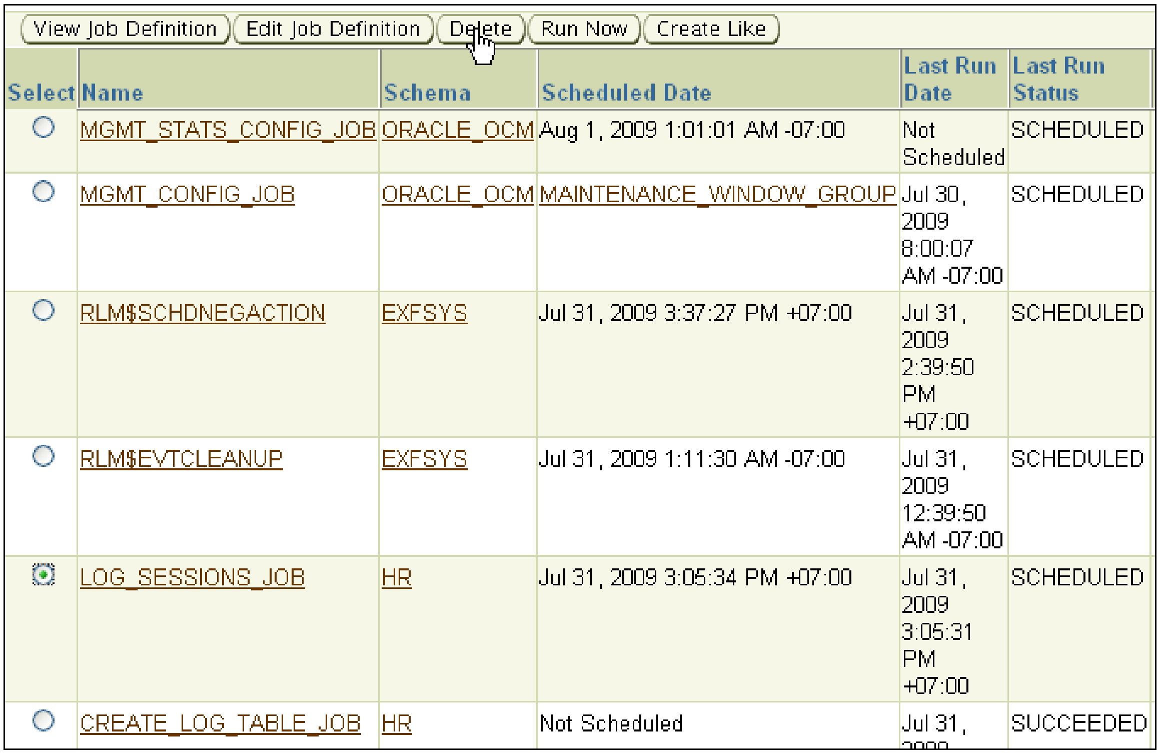
Your result looks different (but the minute values should be three minutes apart):

```
SNAP_TIME
---------------------------------------------------------------
NUM_SESSIONS
-------------
31-JUL-09 03.10.39.185103 PM
          41

31-JUL-09 03.13.38.927866 PM
          41
```

10) **This is your mandatory cleanup task.** Use Enterprise Manager to drop the `LOG_SESSIONS_JOB` and `CREATE_LOG_TABLE_JOB` jobs, the `LOG_SESS_COUNT_PRGM` program, and the `SESS_UPDATE_SCHED` schedule. Use SQL*Plus to drop the `SESSION_HISTORY` table, and exit from your session.

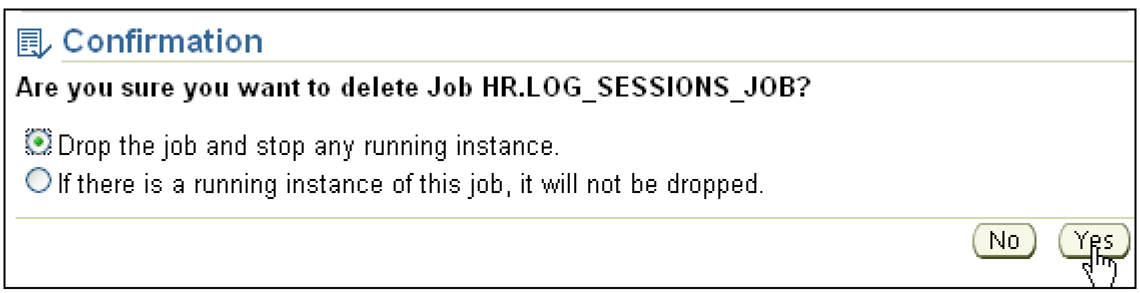
Note: Make sure that you do not delete the wrong schedule.

a) In Enterprise Manager, select Server > Jobs.



Select	Name	Schema	Scheduled Date	Last Run Date	Last Run Status
○	MGMT_STATS_CONFIG_JOB	ORACLE_OCM	Aug 1, 2009 1:01:01 AM -07:00	Not Scheduled	SCHEDULED
○	MGMT_CONFIG_JOB	ORACLE_OCM	MAINTENANCE_WINDOW_GROUP	Jul 30, 2009 8:00:07 AM -07:00	SCHEDULED
○	RLM$SCHDNEGACTION	EXFSYS	Jul 31, 2009 3:37:27 PM +07:00	Jul 31, 2009 2:39:50 PM +07:00	SCHEDULED
○	RLM$EVTCLEANUP	EXFSYS	Jul 31, 2009 1:11:30 AM -07:00	Jul 31, 2009 12:39:50 AM -07:00	SCHEDULED
◉	LOG_SESSIONS_JOB	HR	Jul 31, 2009 3:05:34 PM +07:00	Jul 31, 2009 3:05:31 PM +07:00	SCHEDULED
○	CREATE_LOG_TABLE_JOB	HR	Not Scheduled	Jul 31,	SUCCEEDED

b) With the `LOG_SESSIONS_JOB` job selected, click the Delete button.



c) Select "Drop the job and stop any running instance," and then click Yes.

d) Go back to the Scheduler Jobs page, select `CREATE_LOG_TABLE_JOB`, and click Delete. Select "Drop the job and stop any running instance," and then click Yes.

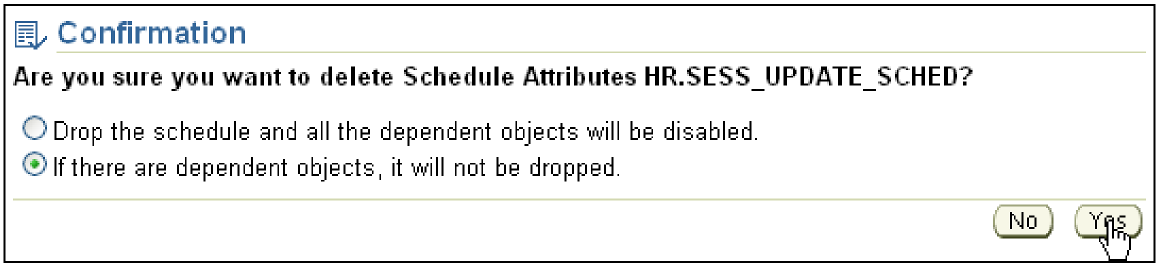
e) Click the Database Instance breadcrumb at the upper-left corner of the page to return to the Server page. Then click Programs.

f) With the `LOG_SESS_COUNT_PRGM` program selected, click the Delete button. Click Yes to confirm.

Practice 17-1: Creating Scheduler Components (continued)

g) Click Schedules in the Related Links section

h) With the `SESS_UPDATE_SCHED` schedule selected, click the Delete button. *Make sure that you do not delete the wrong schedule.*



i) Select “If there are dependent objects, it will not be dropped,” and then click Yes to confirm.

j) In your SQL*Plus session as the HR user, delete the SESSION_HISTORY table, and then exit the session. Enter:

```
SQL> DROP TABLE session_history PURGE;
Table dropped.

SQL> EXIT
```

Practice 17-2: Creating Lightweight Scheduler Jobs

In this optional practice, you create and run a lightweight scheduler job. View the metadata for a lightweight scheduler job. Navigate to your `$HOME/labs` directory.

1) Create a job template for the lightweight job. The template must be a PL/SQL procedure or a PL/SQL block. Run the `cr_test_log.sql` script to create the `test_log` table. Then run `prog_1.sql`. The `prog_1.sql` script in the `$HOME/labs` directory creates a job template.
   Note: The job template has a subset of the attributes of a scheduler program. Most of the attributes of a template cannot be changed for the job.

   a) Navigate to the `labs` directory.

```
$ cd ~/labs
$
```

   b) Execute the `cr_test_log.sql` and `prog_1.sql` scripts as the `system` user. The password for the `system` user is `oracle_4U`.

```
$ sqlplus system
Enter password: oracle_4U <<< not displayed

SQL> @cr_test_log.sql
SQL> -- cleanup previous runs
SQL> -- you will see an error the first time this script is
run
SQL> drop table system.test_log;
drop table system.test_log
                     *
ERROR at line 1:
ORA-00942: table or view does not exist


SQL>
SQL> -- create a table to hold timing information
SQL>
SQL> create table system.test_log
  2  (job_type       VARCHAR2(10),
  3  timemark        VARCHAR2(10),
  4  act_time        TIMESTAMP with TIME ZONE)
  5  /

Table created.
```

```
SQL> @prog_1.sql
SQL> REM For training only
SQL> set echo on
SQL>
SQL> BEGIN
  2  -- This will produce an error the first
  3  -- time it is run since PROG_1 does not exist
  4
```

Practice 17-2: Creating Lightweight Scheduler Jobs (continued)

```
  5  DBMS_SCHEDULER.DROP_PROGRAM (
  6     program_name           => '"SYSTEM"."PROG_1"');
  7  END;
  8  /
BEGIN
*
ERROR at line 1:
ORA-27476: "SYSTEM.PROG_1" does not exist
ORA-06512: at "SYS.DBMS_ISCHED", line 27
ORA-06512: at "SYS.DBMS_SCHEDULER", line 61
ORA-06512: at line 5


SQL> BEGIN
  2  DBMS_SCHEDULER.CREATE_PROGRAM(
  3  program_name=>'"SYSTEM"."PROG_1"'
  4  ,program_action=>'DECLARE
  5   time_now DATE;
  6  BEGIN
  7   INSERT INTO test_log
VALUES(''LWT'',''DONE'',SYSTIMESTAMP);
  8  END;'
  9  , program_type=>'PLSQL_BLOCK'
 10  , number_of_arguments=>0,
 11  comments=>'Insert a timestamp into the test_log'
 12  ,enabled=>TRUE);
 13  END;
 14  /
.
PL/SQL procedure successfully completed.
SQL>
```

2) Create a lightweight job, using the PL/SQL API. The job will run the `my_prog` template daily with an interval of 2, starting immediately.
Note: EM does not expose the `JOB_STYLE` setting at this time.

   a) Logged into SQL*Plus as the system user, execute the `my_lwt_job.sql` script.

```
SQL> @my_lwt_job.sql
SQL> REM For training only
SQL> set echo on
SQL> BEGIN
  2    -- the drop procedure will give and error the first
time
  3    -- this script is run
  4    sys.DBMS_SCHEDULER.DROP_JOB('my_lwt_job');
  5  END;
  6  /
BEGIN
*
```

Practice 17-2: Creating Lightweight Scheduler Jobs (continued)

```
ERROR at line 1:
ORA-27475: "SYSTEM.MY_LWT_JOB" must be a job
ORA-06512: at "SYS.DBMS_ISCHED", line 213
ORA-06512: at "SYS.DBMS_SCHEDULER", line 651
ORA-06512: at line 4


SQL>
SQL> DECLARE
  2     jobname VARCHAR2(30);
  3  BEGIN
  4  -- Create the Job
  5  jobname := 'my_lwt_job';
  6  sys.dbms_scheduler.create_job(
  7      job_name => '"SYSTEM"."MY_LWT_JOB"',
  8      program_name => '"SYSTEM"."PROG_1"',
  9      job_class => '"DEFAULT_JOB_CLASS"',
 10      job_style => 'LIGHTWEIGHT',
 11      repeat_interval => 'FREQ=DAILY;INTERVAL=2',
 12      comments => 'Lightweight job',
 13      enabled => TRUE);
 14  END;
 15  /

PL/SQL procedure successfully completed.

SQL>
```

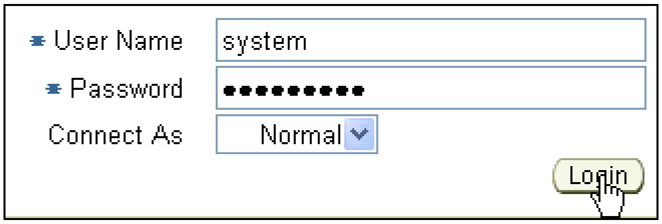
3) Check the Scheduler metadata views `USER_SCHEDULER_JOBS`, `_PROGRAMS`, `DBA_JOBS`. Select `JOB_NAME`, `JOB_STYLE`, and `PROGRAM_NAME` from `USER_SCHEDULER_JOBS`.

```
SQL> COL program_name format a12
SQL> SELECT job_name, job_style, program_name FROM
USER_SCHEDULER_JOBS;

JOB_NAME                       JOB_STYLE   PROGRAM_NAME
------------------------------ ----------- ------------
MY_LWT_JOB                     LIGHTWEIGHT PROG_1
```

4) Check the Enterprise Manager Scheduler Jobs page, find the `MY_LWT_JOB`, and view the attributes.

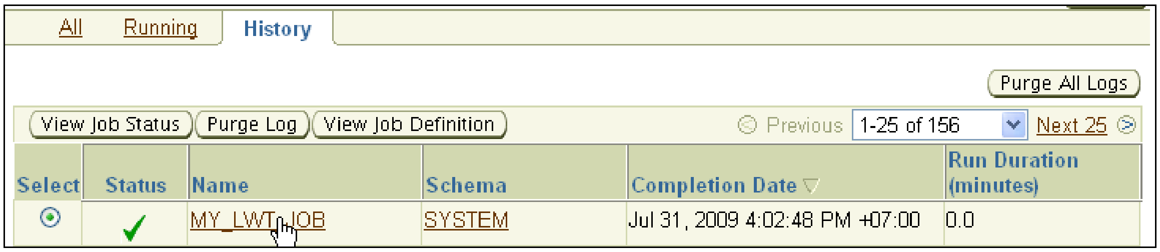
   a) Log into Enterprise Manager as the `SYSTEM` user with the `oracle_4U` password.

Practice 17-2: Creating Lightweight Scheduler Jobs (continued)

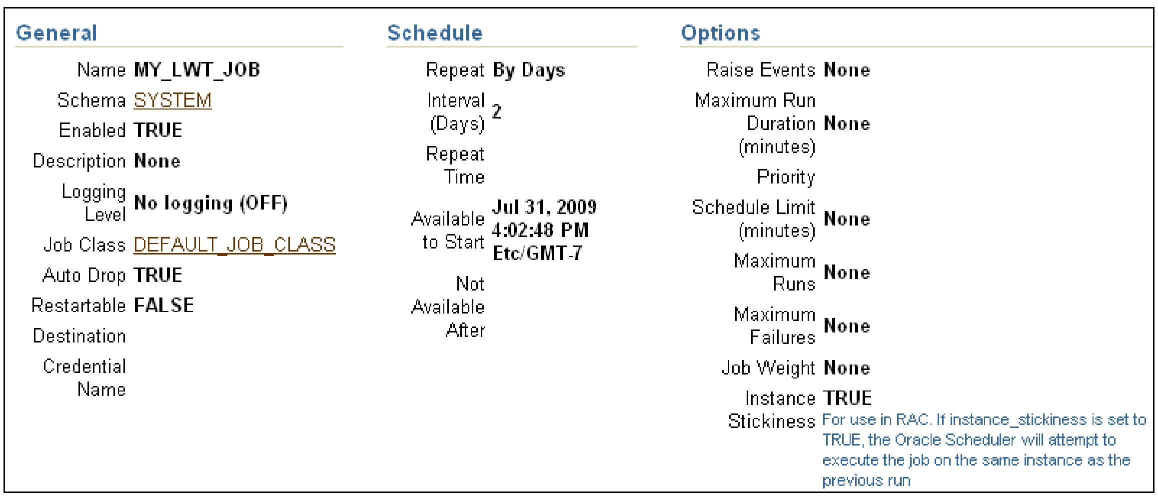


b) Navigate to the Scheduler Jobs page. Then click the History tab.

c) View the history of `MY_LWT_JOB`.



d) Click the Job Name to view the job attributes.

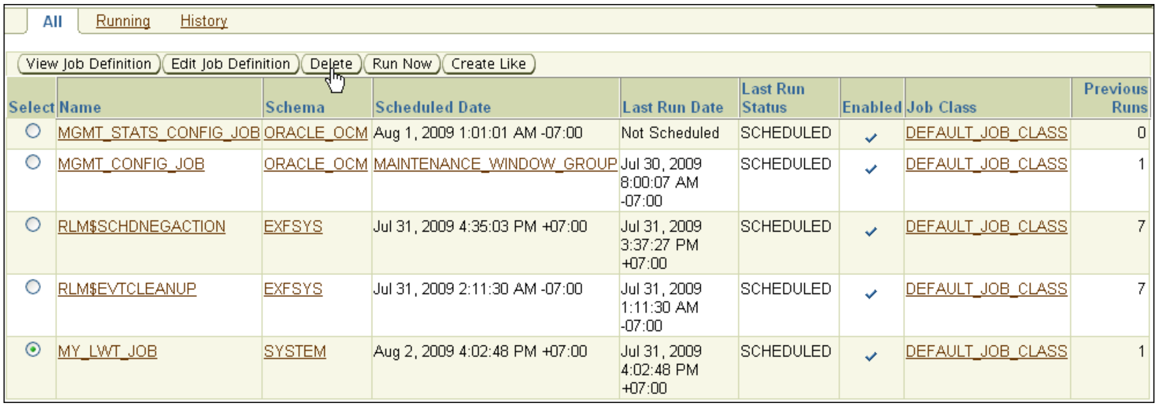


e) Click OK.

5) On the Scheduler Jobs, All page, delete the `MY_LWT_JOB` job

a) Navigate to the Scheduler Jobs, All page, select the `MY_LWT_JOB` job and click Delete.

Practice 17-2: Creating Lightweight Scheduler Jobs (continued)

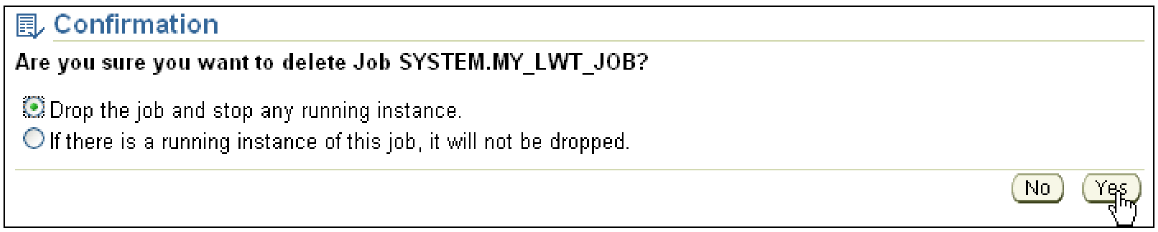


All | Running | History

View Job Definition | Edit Job Definition | Delete | Run Now | Create Like

Select	Name	Schema	Scheduled Date	Last Run Date	Last Run Status	Enabled	Job Class	Previous Runs
○	MGMT_STATS_CONFIG_JOB	ORACLE_OCM	Aug 1, 2009 1:01:01 AM -07:00	Not Scheduled	SCHEDULED	✓	DEFAULT_JOB_CLASS	0
○	MGMT_CONFIG_JOB	ORACLE_OCM	MAINTENANCE_WINDOW_GROUP	Jul 30, 2009 8:00:07 AM -07:00	SCHEDULED	✓	DEFAULT_JOB_CLASS	1
○	RLM$SCHDNEGACTION	EXFSYS	Jul 31, 2009 4:35:03 PM +07:00	Jul 31, 2009 3:37:27 PM +07:00	SCHEDULED	✓	DEFAULT_JOB_CLASS	7
○	RLM$EVTCLEANUP	EXFSYS	Jul 31, 2009 2:11:30 AM -07:00	Jul 31, 2009 1:11:30 AM -07:00	SCHEDULED	✓	DEFAULT_JOB_CLASS	7
⦿	MY_LWT_JOB	SYSTEM	Aug 2, 2009 4:02:48 PM +07:00	Jul 31, 2009 4:02:48 PM +07:00	SCHEDULED	✓	DEFAULT_JOB_CLASS	1

b) On the Confirmation page, select “Drop the job and any running instance.” Click Yes.



Practice 17-3: Monitoring the Scheduler

Background: Because your job tasks are regularly increasing, you decide to automate routine tasks. You first monitor existing scheduler elements, and then you create scheduler components and test them.

In this practice, use Enterprise Manager Database Control to define and monitor the Scheduler and automate tasks. Click Show SQL regularly to review all statements that are new to you.

Log in as the `SYS` user (with `oracle_4U` password, connect as `SYSDBA`) or as `HR` user (with `oracle_4U` password, connect as `Normal`), as indicated. Perform the necessary tasks either through Enterprise Manager Database Control or through SQL*Plus. All scripts for this practice are in the `/home/oracle/labs` directory.

1) Log in to Enterprise Manager Database Control as the `SYS` user with the `oracle_4U` password, connect as `SYSDBA`, and check the following roles for the `HR` user:

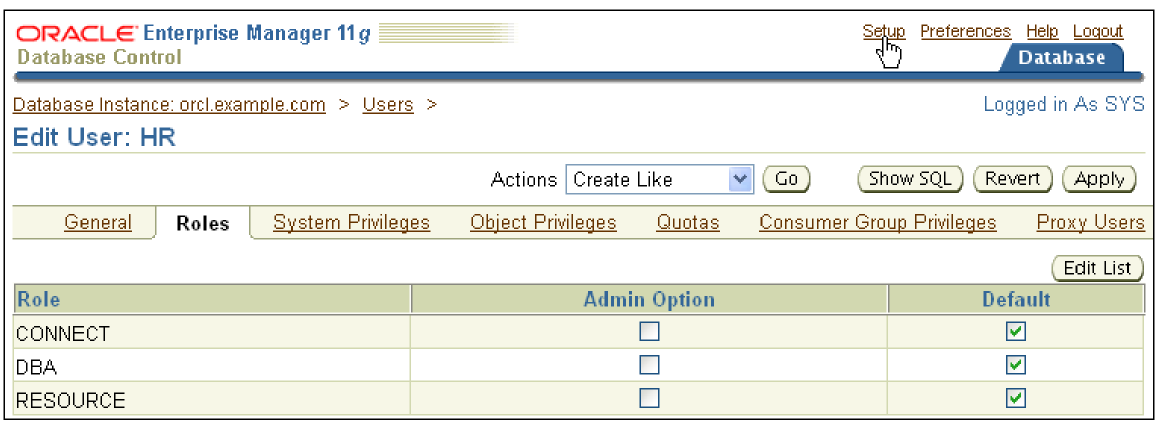
   - `CONNECT` role
   - `RESOURCE` role
   - `DBA` role

   Because you are going to use the `HR` user to administer jobs through Database Control, you need to make sure that `HR` is registered as a possible administrator.

   a) In Enterprise Manager, naviaget to Server > Users (in the Security section).

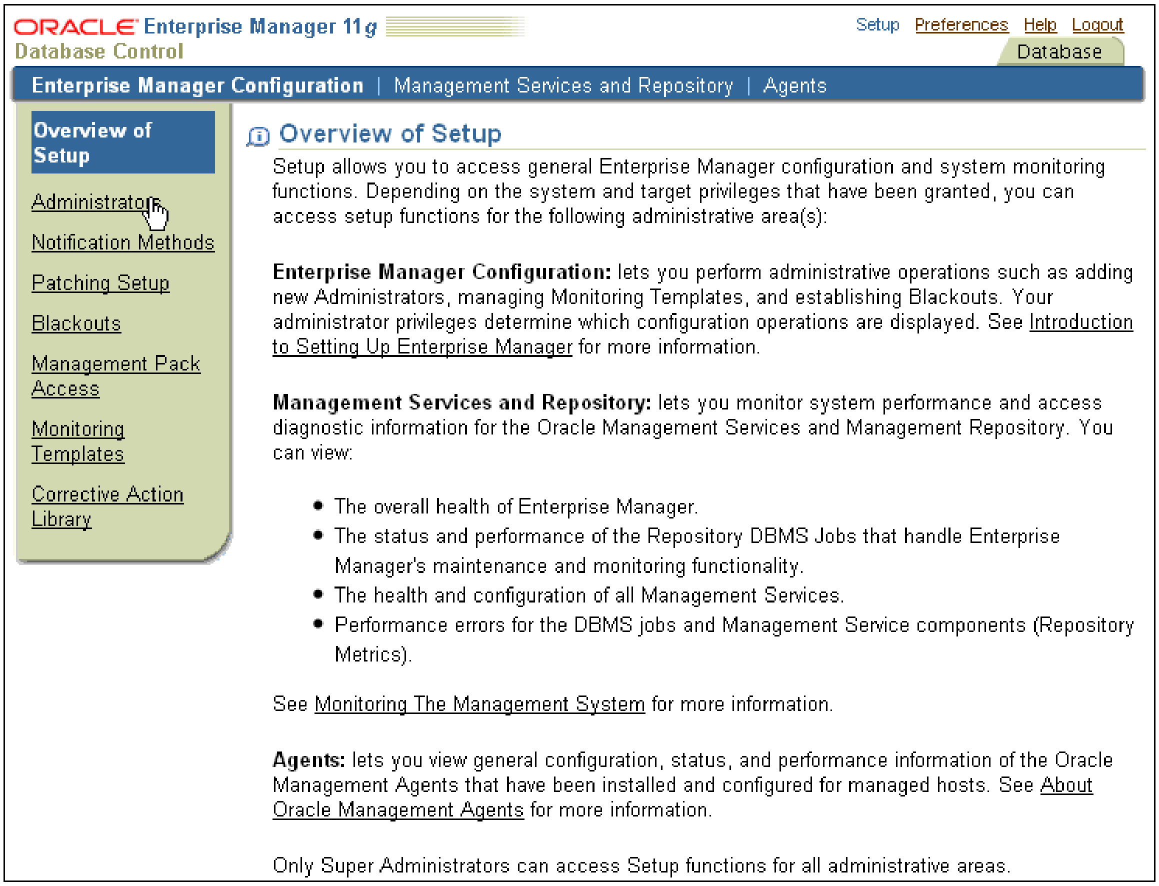
   b) On the Users page, select the `HR` user and click Edit.

   c) On the Edit User page, click the Roles tab.

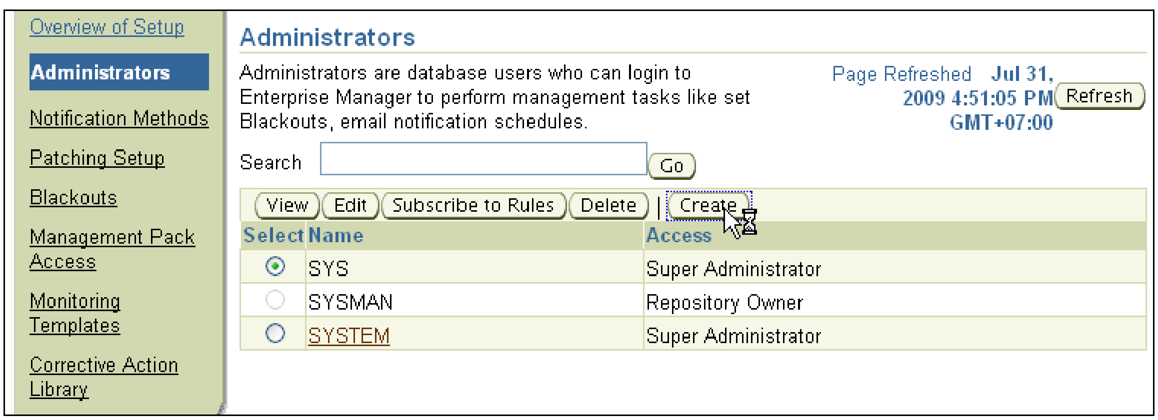


   d) If the roles are not yet selected, then click the Edit List button on the right side of the page. On the Modify Roles page, make sure that the `DBA`, `CONNECT`, and `RESOURCE` roles are selected. Click OK, then click Apply.

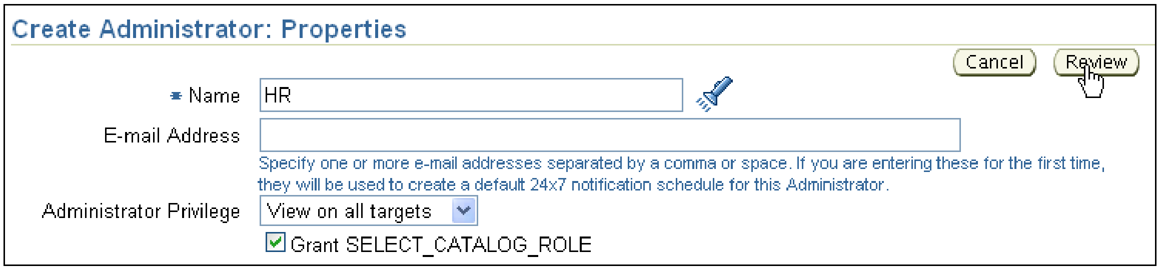
   e) Click the Setup link in the upper-right region of the page.



f) On the Enterprise Manager Configuration page, click the Administrators link.



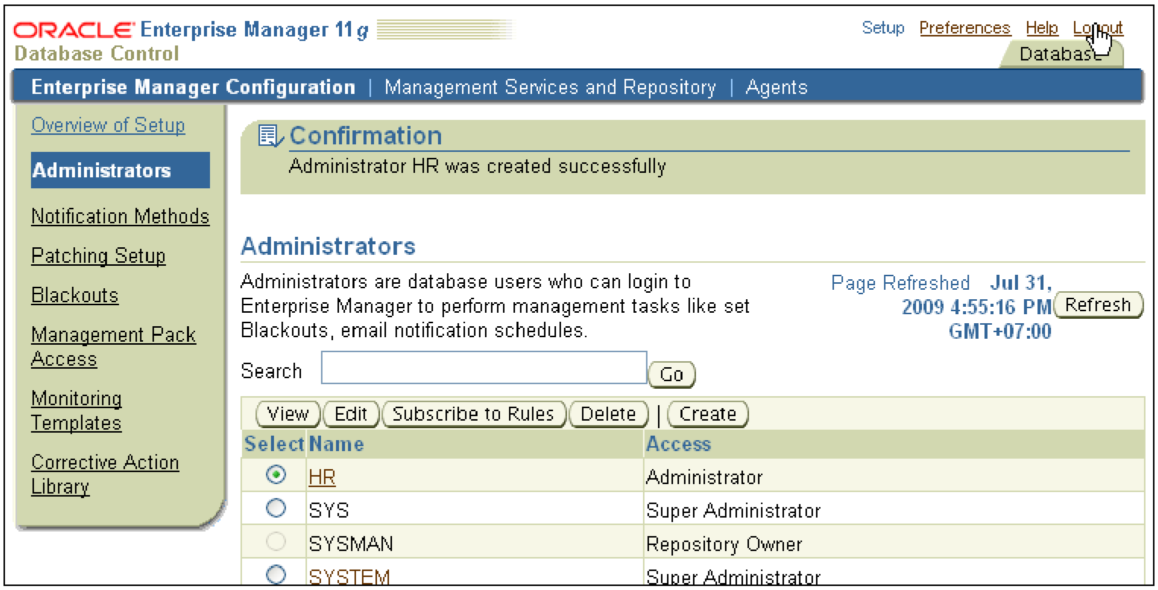
g) On the Administrators page, click the Create button.



h) On the Create Administrators: Properties page, enter `HR` as Name, ensure that the Grant SELECT_CATALOG_ROLE is selected and click Review.

i) On the Create Administrator HR: Review page, click the Finish button.

You should receive a success message



j) Click the Logout link at the upper-right corner of the page.

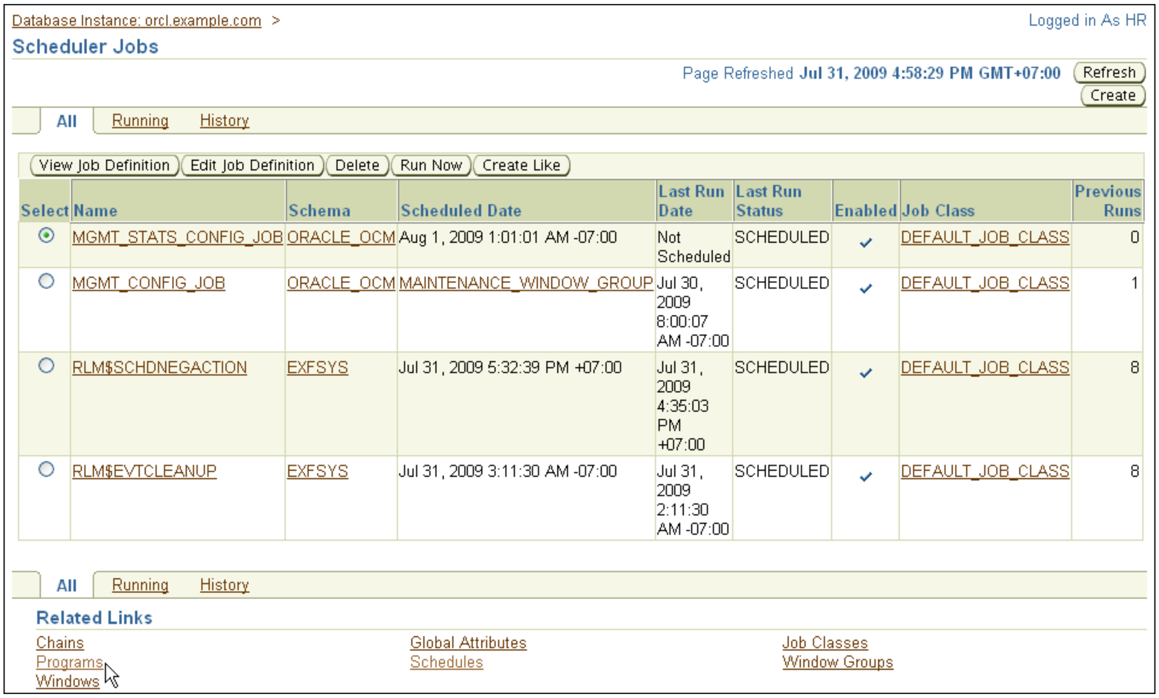
2) Log in to Enterprise Manager Database Control as the `HR` user. On the Server tabbed page, click the Jobs link in the Database Scheduler region. Are there any jobs?

a) Click the Login button to log in as the `HR` user.

b) Enter `HR` as username, `oracle_4U` as password, Connect As `Normal`, and click Login.

c) In Enterprise Manager, navigate to Server > Jobs.

Practice 17-3: Monitoring the Scheduler (continued)



Database Instance: orcl.example.com > Logged in As HR

Scheduler Jobs

Page Refreshed Jul 31, 2009 4:58:29 PM GMT+07:00 Refresh Create

All Running History

View Job Definition | Edit Job Definition | Delete | Run Now | Create Like

Select	Name	Schema	Scheduled Date	Last Run Date	Last Run Status	Enabled	Job Class	Previous Runs
⦿	MGMT_STATS_CONFIG_JOB	ORACLE_OCM	Aug 1, 2009 1:01:01 AM -07:00	Not Scheduled	SCHEDULED	✓	DEFAULT_JOB_CLASS	0
○	MGMT_CONFIG_JOB	ORACLE_OCM	MAINTENANCE_WINDOW_GROUP	Jul 30, 2009 8:00:07 AM -07:00	SCHEDULED	✓	DEFAULT_JOB_CLASS	1
○	RLM$SCHDNEGACTION	EXFSYS	Jul 31, 2009 5:32:39 PM +07:00	Jul 31, 2009 4:35:03 PM +07:00	SCHEDULED	✓	DEFAULT_JOB_CLASS	8
○	RLM$EVTCLEANUP	EXFSYS	Jul 31, 2009 3:11:30 AM -07:00	Jul 31, 2009 2:11:30 AM -07:00	SCHEDULED	✓	DEFAULT_JOB_CLASS	8

All Running History

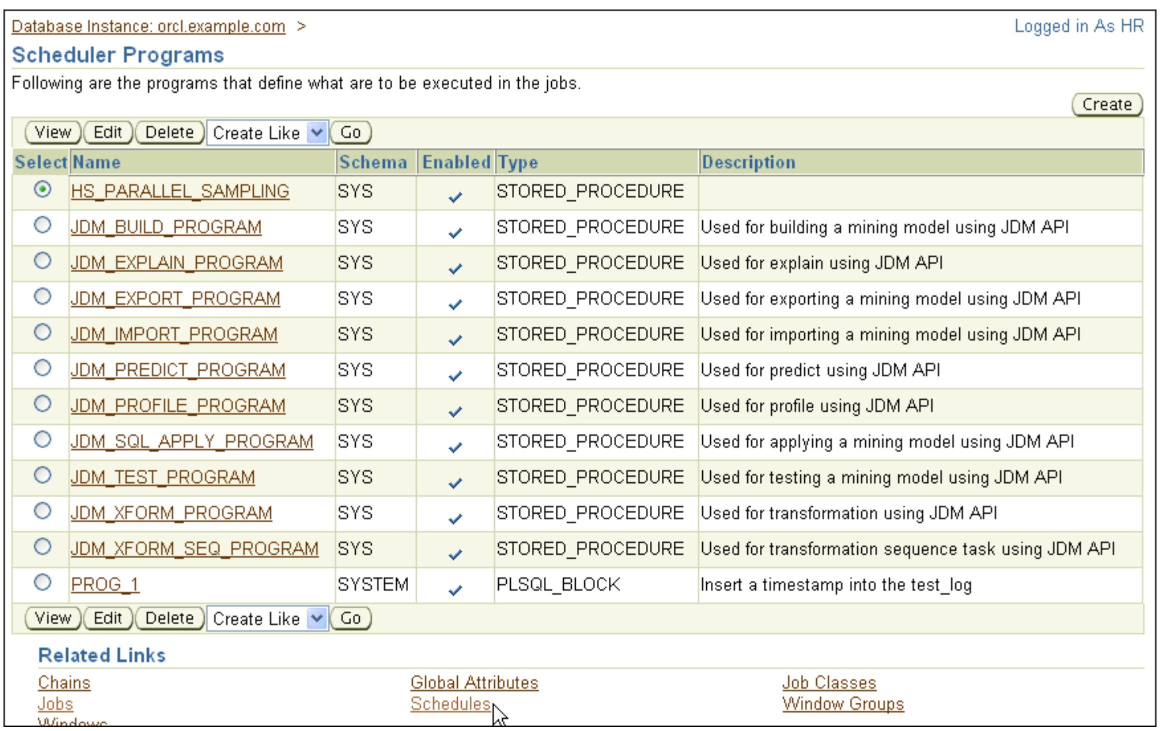
Related Links

Chains, Programs, Windows, Global Attributes, Schedules, Job Classes, Window Groups

Question: Are there any jobs?

Possible Answer: There are some jobs.

3) Click Programs in the Related Links section.



Database Instance: orcl.example.com > Logged in As HR

Scheduler Programs

Following are the programs that define what are to be executed in the jobs.

Create

View | Edit | Delete | Create Like | Go

Select	Name	Schema	Enabled	Type	Description
⦿	HS_PARALLEL_SAMPLING	SYS	✓	STORED_PROCEDURE	
○	JDM_BUILD_PROGRAM	SYS	✓	STORED_PROCEDURE	Used for building a mining model using JDM API
○	JDM_EXPLAIN_PROGRAM	SYS	✓	STORED_PROCEDURE	Used for explain using JDM API
○	JDM_EXPORT_PROGRAM	SYS	✓	STORED_PROCEDURE	Used for exporting a mining model using JDM API
○	JDM_IMPORT_PROGRAM	SYS	✓	STORED_PROCEDURE	Used for importing a mining model using JDM API
○	JDM_PREDICT_PROGRAM	SYS	✓	STORED_PROCEDURE	Used for predict using JDM API
○	JDM_PROFILE_PROGRAM	SYS	✓	STORED_PROCEDURE	Used for profile using JDM API
○	JDM_SQL_APPLY_PROGRAM	SYS	✓	STORED_PROCEDURE	Used for applying a mining model using JDM API
○	JDM_TEST_PROGRAM	SYS	✓	STORED_PROCEDURE	Used for testing a mining model using JDM API
○	JDM_XFORM_PROGRAM	SYS	✓	STORED_PROCEDURE	Used for transformation using JDM API
○	JDM_XFORM_SEQ_PROGRAM	SYS	✓	STORED_PROCEDURE	Used for transformation sequence task using JDM API
○	PROG_1	SYSTEM	✓	PLSQL_BLOCK	Insert a timestamp into the test_log

View | Edit | Delete | Create Like | Go

Related Links

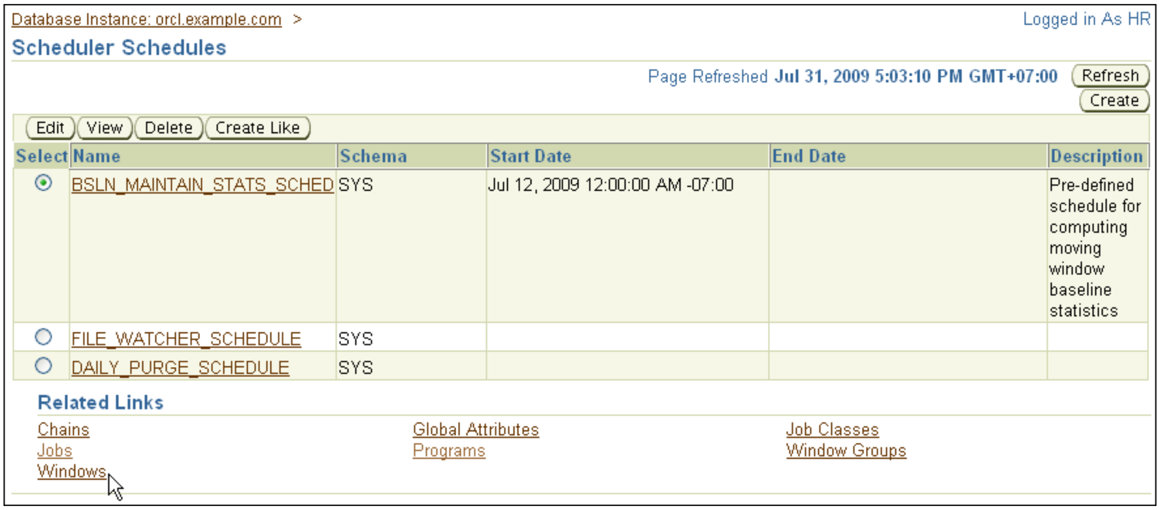
Chains, Jobs, Windows, Global Attributes, Schedules, Job Classes, Window Groups

Question: Are there any existing programs?

Practice 17-3: Monitoring the Scheduler (continued)

Answer: There are some existing programs.

4) Click Schedules in the Related Links section.

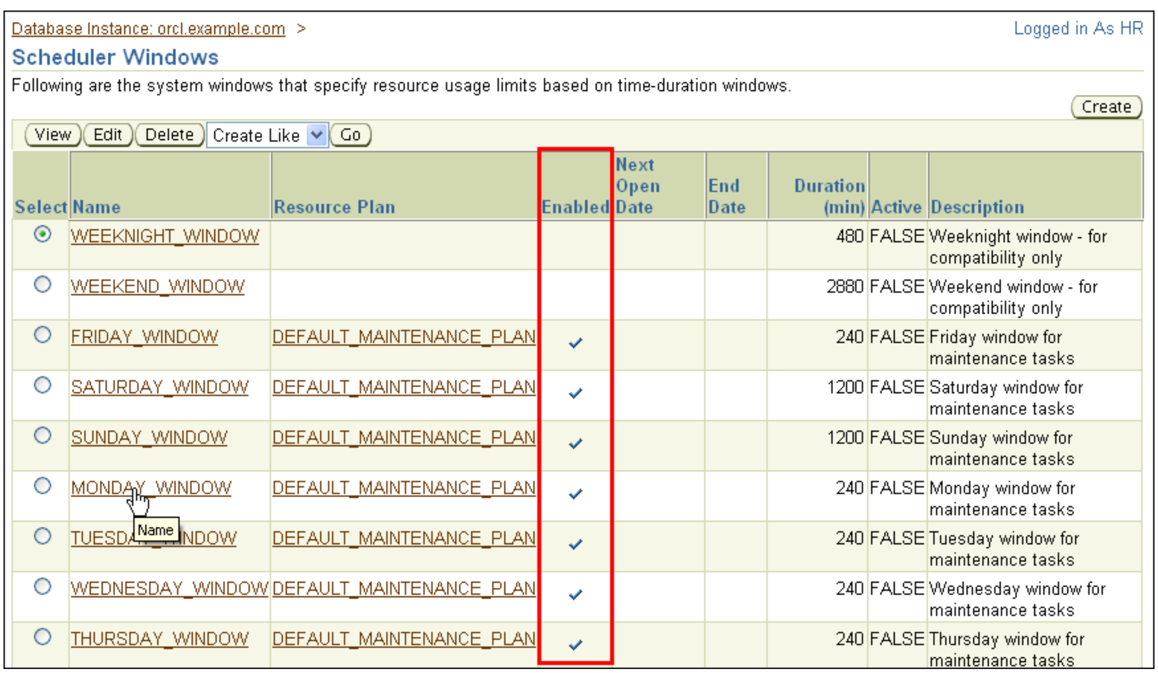


Select	Name	Schema	Start Date	End Date	Description
◉	BSLN_MAINTAIN_STATS_SCHED	SYS	Jul 12, 2009 12:00:00 AM -07:00		Pre-defined schedule for computing moving window baseline statistics
○	FILE_WATCHER_SCHEDULE	SYS			
○	DAILY_PURGE_SCHEDULE	SYS			

Question: Are there any existing schedules?

Answer: There are three schedules: `BSLN_MAINTAIN_STATS_SCHED`, `FILE_WATCHER_SCHEDULE` and `DAILY_PURGE_SCHEDULE`.

5) Click Windows in the Related Links section. Review the Scheduler Windows page in Enterprise Manager. Are there any existing windows? Which resource plan is associated with each window?



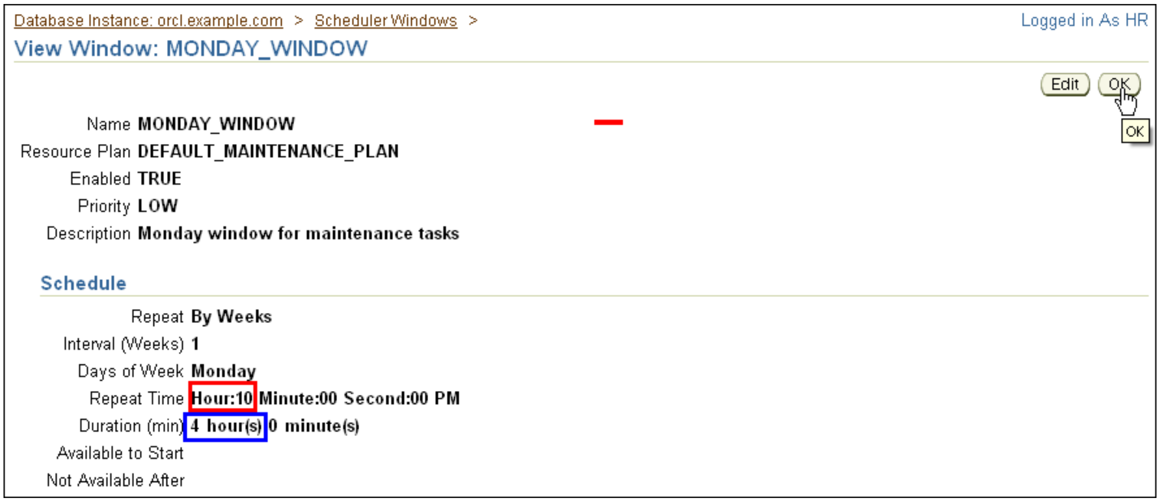
Select	Name	Resource Plan	Enabled	Next Open Date	End Date	Duration (min)	Active	Description
◉	WEEKNIGHT_WINDOW					480	FALSE	Weeknight window - for compatibility only
○	WEEKEND_WINDOW					2880	FALSE	Weekend window - for compatibility only
○	FRIDAY_WINDOW	DEFAULT_MAINTENANCE_PLAN	✓			240	FALSE	Friday window for maintenance tasks
○	SATURDAY_WINDOW	DEFAULT_MAINTENANCE_PLAN	✓			1200	FALSE	Saturday window for maintenance tasks
○	SUNDAY_WINDOW	DEFAULT_MAINTENANCE_PLAN	✓			1200	FALSE	Sunday window for maintenance tasks
○	MONDAY_WINDOW	DEFAULT_MAINTENANCE_PLAN	✓			240	FALSE	Monday window for maintenance tasks
○	TUESDAY_WINDOW	DEFAULT_MAINTENANCE_PLAN	✓			240	FALSE	Tuesday window for maintenance tasks
○	WEDNESDAY_WINDOW	DEFAULT_MAINTENANCE_PLAN	✓			240	FALSE	Wednesday window for maintenance tasks
○	THURSDAY_WINDOW	DEFAULT_MAINTENANCE_PLAN	✓			240	FALSE	Thursday window for maintenance tasks

Practice 17-3: Monitoring the Scheduler (continued)

Question 1: Are there any existing windows? Are any enabled?

Answer: There are several windows. All are enabled except `WEEKNIGHT_WINDOW` and `WEEKEND_WINDOW`.

6) Click the `MONDAY_WINDOW` link. Answer the questions, then click OK.



Question 1: At which time does this window open?

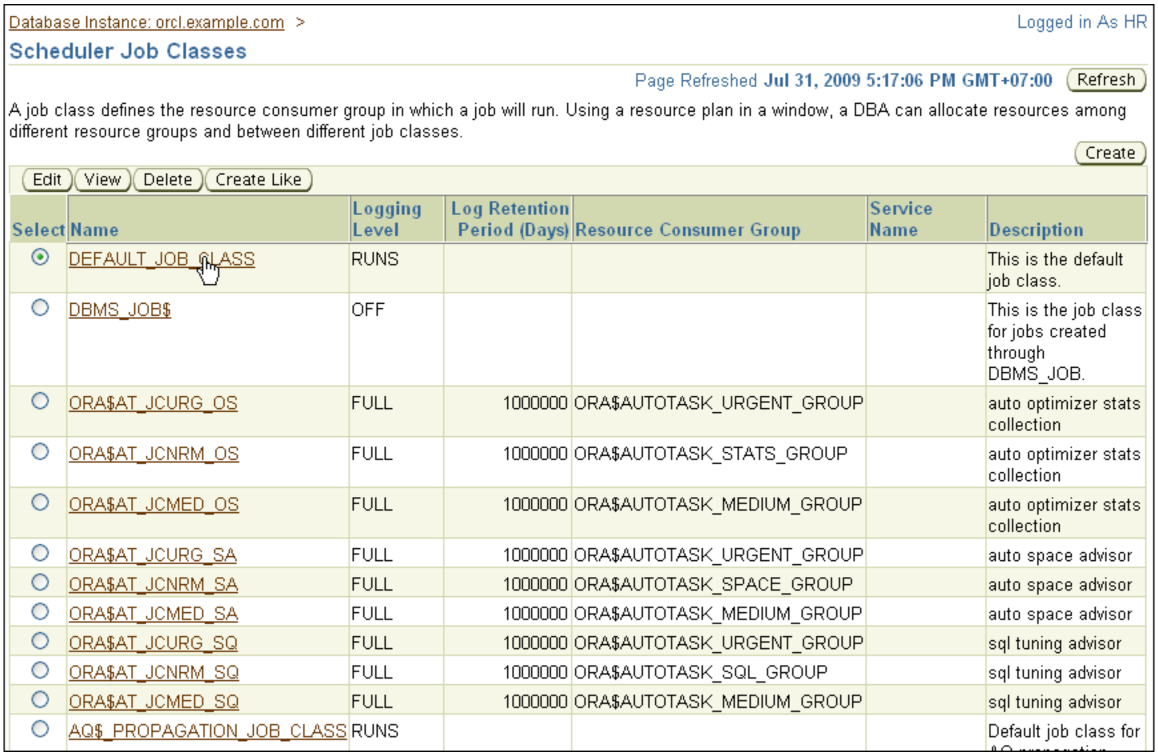
Possible Answer: 10 PM

Question 2: For how long does it stay open?

Possible Answer: for 4 hours

7) Click Job Classes in the Related Links section and review them.

Practice 17-3: Monitoring the Scheduler (continued)


Database Instance: orcl.example.com > Logged in As HR

Scheduler Job Classes

Page Refreshed Jul 31, 2009 5:17:06 PM GMT+07:00 Refresh

A job class defines the resource consumer group in which a job will run. Using a resource plan in a window, a DBA can allocate resources among different resource groups and between different job classes.

Create

Edit View Delete Create Like

Select	Name	Logging Level	Log Retention Period (Days)	Resource Consumer Group	Service Name	Description
⦿	DEFAULT_JOB_CLASS	RUNS				This is the default job class.
○	DBMS_JOB$	OFF				This is the job class for jobs created through DBMS_JOB.
○	ORA$AT_JCURG_OS	FULL	1000000	ORA$AUTOTASK_URGENT_GROUP		auto optimizer stats collection
○	ORA$AT_JCNRM_OS	FULL	1000000	ORA$AUTOTASK_STATS_GROUP		auto optimizer stats collection
○	ORA$AT_JCMED_OS	FULL	1000000	ORA$AUTOTASK_MEDIUM_GROUP		auto optimizer stats collection
○	ORA$AT_JCURG_SA	FULL	1000000	ORA$AUTOTASK_URGENT_GROUP		auto space advisor
○	ORA$AT_JCNRM_SA	FULL	1000000	ORA$AUTOTASK_SPACE_GROUP		auto space advisor
○	ORA$AT_JCMED_SA	FULL	1000000	ORA$AUTOTASK_MEDIUM_GROUP		auto space advisor
○	ORA$AT_JCURG_SQ	FULL	1000000	ORA$AUTOTASK_URGENT_GROUP		sql tuning advisor
○	ORA$AT_JCNRM_SQ	FULL	1000000	ORA$AUTOTASK_SQL_GROUP		sql tuning advisor
○	ORA$AT_JCMED_SQ	FULL	1000000	ORA$AUTOTASK_MEDIUM_GROUP		sql tuning advisor
○	AQ$_PROPAGATION_JOB_CLASS	RUNS				Default job class for

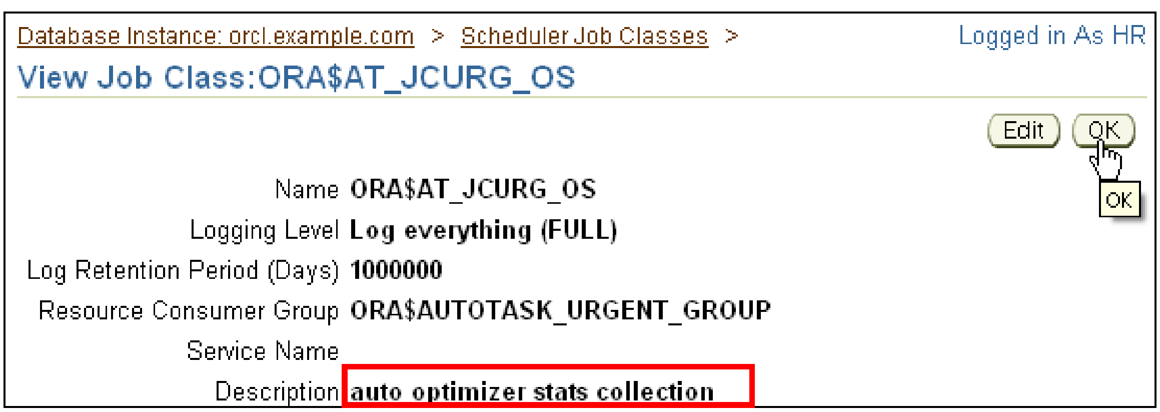
Question 1: Are there any existing job classes?

Possible Answer: There are many job classes.

Question 2: Which resource consumer group is associated with the `DEFAULT_JOB_CLASS` job class?

Possible Answer: None.

8) On the Scheduler Job classes page, click the `ORA$AT_JCURG_OS` link.



Question 1: Which resource consumer group is associated with the job class?

Possible Answer: `ORA$AT_JCURG_OS` is associated with `ORA$AUTOTASK_URGENT_GROUP`.

Practice 17-3: Monitoring the Scheduler (continued)

Question 2: For which task is this job class used?

Possible Answer: For automatic optimizer statistics collection

9) Click OK, and then exit Enterprise Manager.

Practices for Lesson 18

Background: To prepare for an upcoming merger, you want to set the warning and critical thresholds to a lower value than the default. Ensure that you receive early warnings to give you more time to react. When you finish your test case, drop the tablespace that you used.

Practice 18-1: Managing Storage

Access the `orcl` database as the `SYS` user (with the `oracle_4U` password, connect as `SYSDBA`) and perform the necessary tasks through Enterprise Manager Database Control or through SQL*Plus. All scripts for this practice are in the `/home/oracle/labs` directory.

1) Using the `DBMS_SERVER_ALERT.SET_THRESHOLD` procedure, reset the databasewide threshold values for the Tablespace Space Usage metric. Connect to a SQL*Plus session and execute the following procedure:

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
$ cd ~/labs
$ sqlplus / as sysdba
SQL> exec DBMS_SERVER_ALERT.SET_THRESHOLD(-
dbms_server_alert.tablespace_pct_full,-
NULL,NULL,NULL,NULL,1,1,NULL,-
dbms_server_alert.object_type_tablespace,NULL);
> > >
PL/SQL procedure successfully completed.

SQL>
```

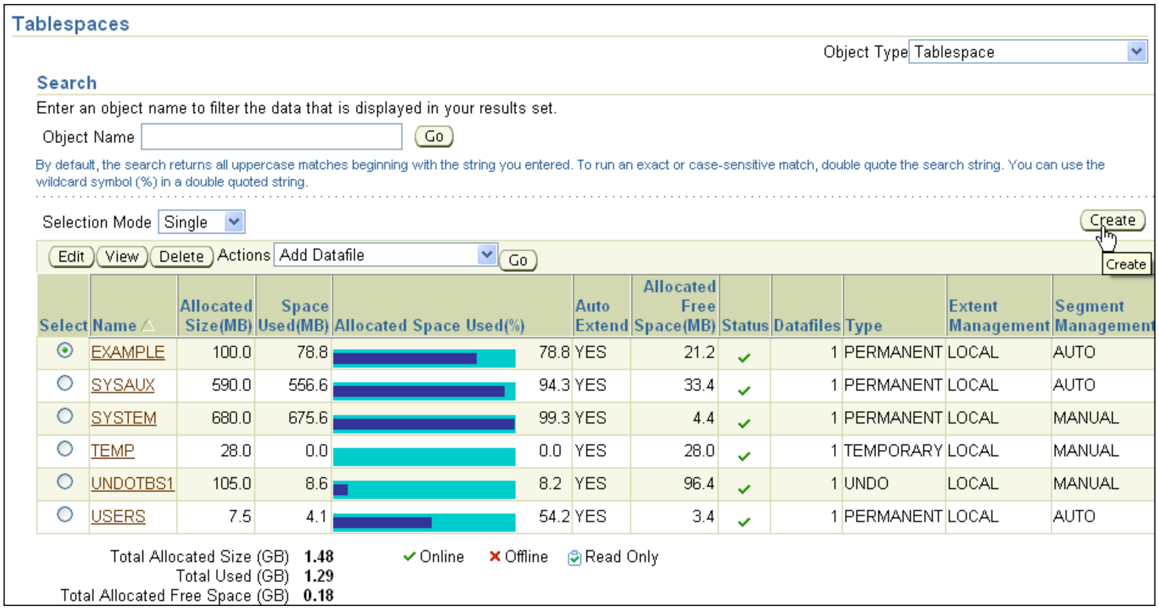
2) From your SQL*Plus session, check the databasewide threshold values for the Tablespace Space Usage metric using the following command:

```
SQL> SELECT warning_value,critical_value
FROM   dba_thresholds
WHERE  metrics_name='Tablespace Space Usage'
AND    object_name IS NULL;

WARNING_VALUE
--------------------------------------------------------------
CRITICAL_VALUE
--------------------------------------------------------------
85
97
```

3) Create a new tablespace called `TBSALERT` with a 120 MB file called `tbsalert.dbf`. Make sure that this tablespace is locally managed and uses Automatic Segment Space Management. Do *not* make it autoextensible, and do *not* specify any thresholds for this tablespace. Use Enterprise Manager Database Control to create it. If this tablespace already exists in your database, drop it first, including its files.

   a) Logged into Enterprise Manager as the `SYS` user, navigate to Server > Tablespaces.

Practice 18-1: Managing Storage (continued)



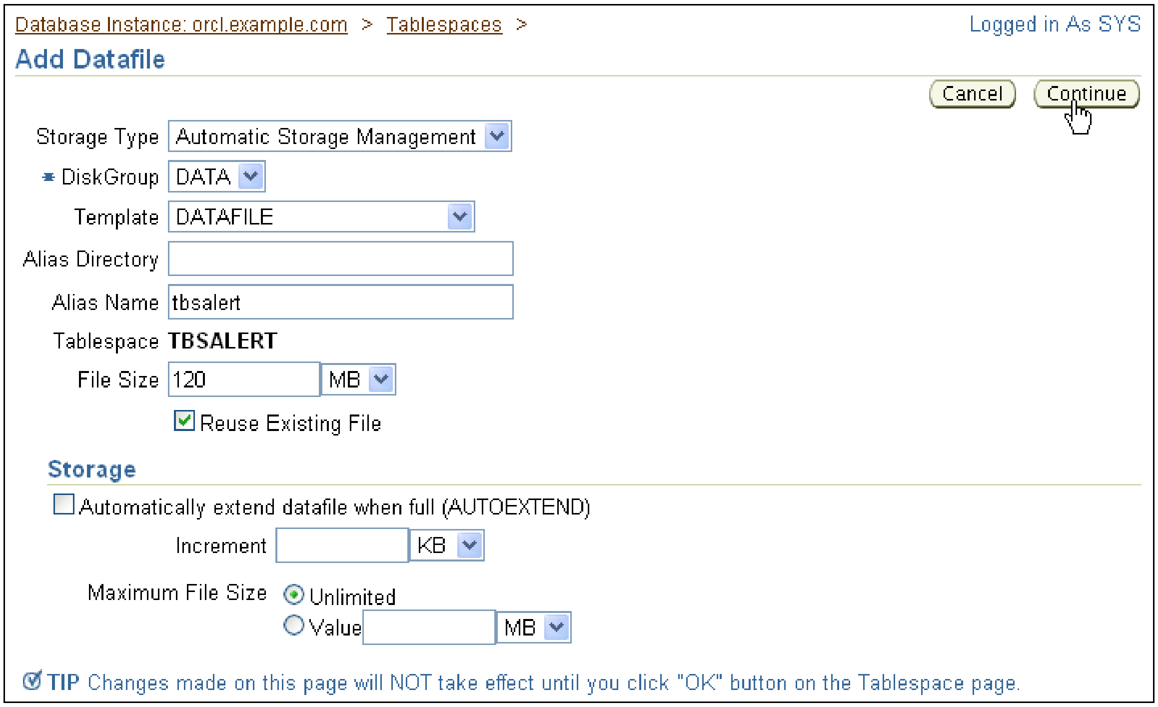
b). Click the Create button.

c) Enter `TBSALERT` as the name, and click the Add button in the Datafiles region.

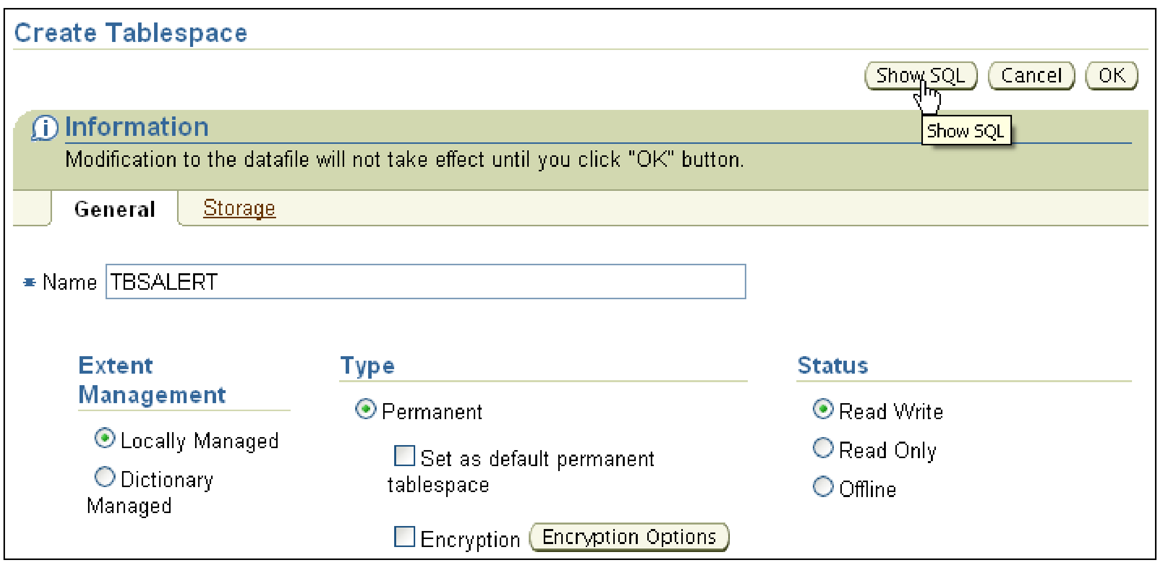
d) Enter or confirm the following values, then click Continue

Name	**Value**
Storage Type	`Automatic Storage Management`
Disk Group	`DATA`
Template	`DATAFILE`
Alias name	`tbsalert`
File Size	`120 MB`
Reuse Existing File	`TRUE`
AUTOEXTEND	`FALSE`

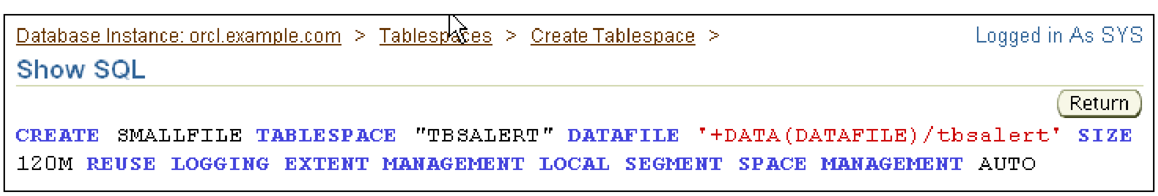
Practice 18-1: Managing Storage (continued)



e) Click Continue



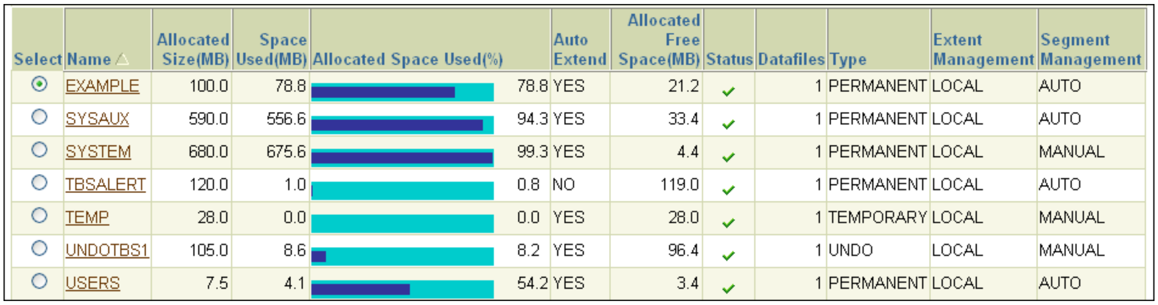
f) On the Crate tablespace page, click Show SQL.



g) Review the SQL and then click Return.

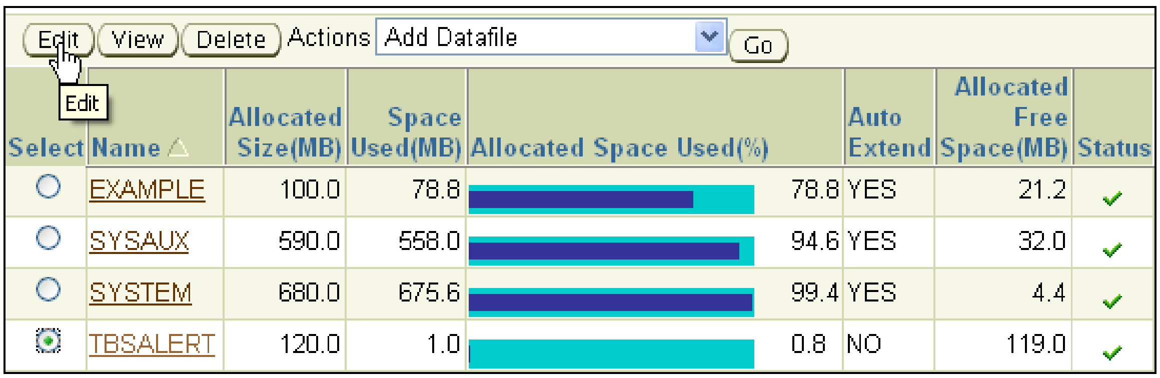
Practice 18-1: Managing Storage (continued)

h) Click OK to create the tablespace. You should receive a success message.



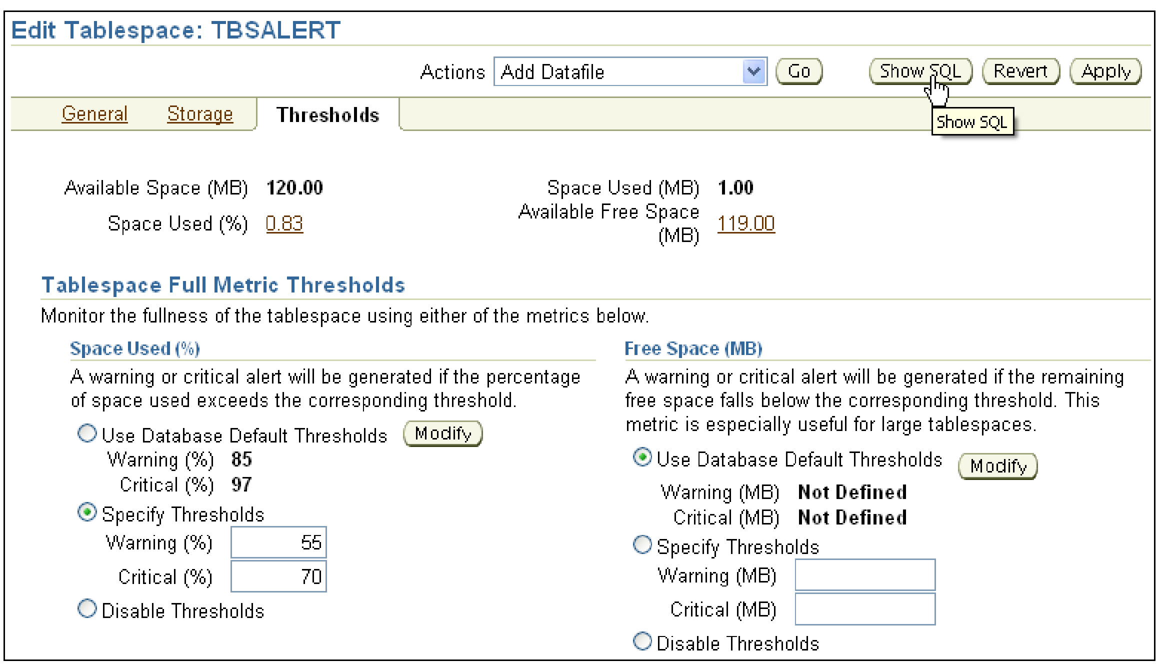
Select	Name △	Allocated Size(MB)	Space Used(MB)	Allocated Space Used(%)	Auto Extend	Allocated Free Space(MB)	Status	Datafiles	Type	Extent Management	Segment Management
◉	EXAMPLE	100.0	78.8	78.8	YES	21.2	✓	1	PERMANENT	LOCAL	AUTO
○	SYSAUX	590.0	556.6	94.3	YES	33.4	✓	1	PERMANENT	LOCAL	AUTO
○	SYSTEM	680.0	675.6	99.3	YES	4.4	✓	1	PERMANENT	LOCAL	MANUAL
○	TBSALERT	120.0	1.0	0.8	NO	119.0	✓	1	PERMANENT	LOCAL	AUTO
○	TEMP	28.0	0.0	0.0	YES	28.0	✓	1	TEMPORARY	LOCAL	MANUAL
○	UNDOTBS1	105.0	8.6	8.2	YES	96.4	✓	1	UNDO	LOCAL	MANUAL
○	USERS	7.5	4.1	54.2	YES	3.4	✓	1	PERMANENT	LOCAL	AUTO

4) In Enterprise Manager, change the Tablespace Space Usage thresholds of the TBSALERT tablespace. Set its warning level to 55 percent and its critical level to 70 percent.

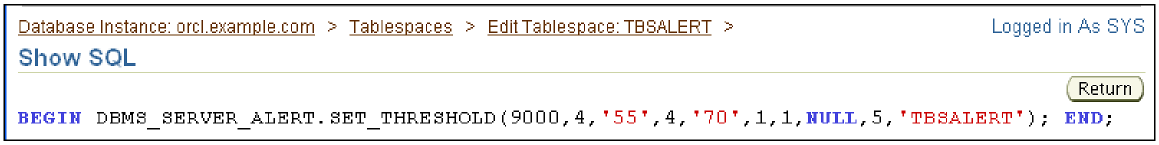


Select	Name △	Allocated Size(MB)	Space Used(MB)	Allocated Space Used(%)	Auto Extend	Allocated Free Space(MB)	Status
○	EXAMPLE	100.0	78.8	78.8	YES	21.2	✓
○	SYSAUX	590.0	558.0	94.6	YES	32.0	✓
○	SYSTEM	680.0	675.6	99.4	YES	4.4	✓
◉	TBSALERT	120.0	1.0	0.8	NO	119.0	✓

a) On the Tablespaces page, select TBSALERT, click Edit, and then click Thresholds.



b) Select Specify Thresholds, and enter `55` as Warning (%) and `70` as Critical (%) under the Space Used section. Then click Show SQL.



c) Review the statement and click Return.

d) On the Edit Tablespace: TBSALERT, click Apply to modify the threshold values.

You should receive a success message.

5) Return to your SQL*Plus session and check the new threshold values for the `TBSALERT` tablespace. In your SQL*Plus session, enter:

```
SQL> select warning_value,critical_value
from dba_thresholds
where metrics_name='Tablespace Space Usage' and
object_name='TBSALERT';

WARNING_VALUE
--------------------------------------------------------------
CRITICAL_VALUE
--------------------------------------------------------------
55
70
```

6) In your SQL*Plus session, query the `reason` and `resolution` columns from `DBA_ALERT_HISTORY` for the `TBSALERT` tablespace.

```
SQL> select reason,resolution
from dba_alert_history
where object_name='TBSALERT';
```

The result should be (if you are repeating this practice, look at the last row):

```
REASON
RESOLUT
------------------------------------------------------------  ----
---
Threshold is updated on metrics "Tablespace Space Usage"
cleared
SQL> exit
$
```

Practice 18-1: Managing Storage (continued)

7) From the `labs` directory, review and execute the `seg_advsr_setup.sh` script that creates and populates new tables in the `TBSALERT` tablespace.

```
$ cd ~/labs
$ cat seg_advsr_setup.sh

#!/bin/sh
# For training only, execute as oracle OS user

sqlplus /nolog <<EOF
connect / as sysdba
alter system set disk_asynch_io = FALSE scope = spfile;
shutdown immediate;
startup
set echo on
create table employees1 tablespace tbsalert as select * from
hr.employees;
create table employees2 tablespace tbsalert as select * from
hr.employees;
create table employees3 tablespace tbsalert as select * from
hr.employees;
create table employees4 tablespace tbsalert as select * from
hr.employees;
create table employees5 tablespace tbsalert as select * from
hr.employees;

alter table employees1 enable row movement;
alter table employees2 enable row movement;
alter table employees3 enable row movement;
alter table employees4 enable row movement;
alter table employees5 enable row movement;

BEGIN
FOR i in 1..10 LOOP
   insert into employees1 select * from employees1;
   insert into employees2 select * from employees2;
   insert into employees3 select * from employees3;
   insert into employees4 select * from employees4;
   insert into employees5 select * from employees5;
   commit;
 END LOOP;
END;
/
insert into employees1 select * from employees1;
insert into employees2 select * from employees2;
insert into employees3 select * from employees3;
commit;
exit
EOF
$
```

```
$ ./seg_advsr_setup.sh

SQL> Connected.
SQL>
System altered.
```

Practice 18-1: Managing Storage (continued)

```
SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> ORACLE instance started.

Total System Global Area  481259520 bytes
Fixed Size                  1337324 bytes
Variable Size             385878036 bytes
Database Buffers           88080384 bytes
Redo Buffers                5963776 bytes
Database mounted.
Database opened.
SQL> SQL>
Table created.
SQL>
Table created.
SQL>
Table created.
SQL>
Table created.
SQL>
Table created.
SQL> SQL>
Table altered.
SQL>
Table altered.
SQL>
Table altered.
SQL>
Table altered.
SQL>
Table altered.
SQL> SQL>   2    3    4    5    6    7    8    9   10   11
PL/SQL procedure successfully completed.
SQL>
109568 rows created.
SQL>
109568 rows created.
SQL>
109568 rows created.
SQL>
Commit complete.
SQL>
```

8) Check the fullness level of the TBSALERT tablespace by using Database Control or SQL*Plus. The current level should be around 60%. Wait a few minutes and check that the warning level is reached for the TBSALERT tablespace. (*If you are too fast and receive erros, just use your browser's Refresh button, or select your destintion again.)*

   a) Logged into SQL*Plus as the SYS user, enter:

```
$ sqlplus / as sysdba
```

Practice 18-1: Managing Storage (continued)

```
SQL> select sum(bytes) *100 /125829120
        from dba_extents
        where tablespace_name='TBSALERT';

SUM(BYTES)*100/125829120
------------------------
                      60
SQL>
```

b) Enter the following query. Your results should be similar to the following:

```
SQL> select reason
from dba_outstanding_alerts
where object_name='TBSALERT';

REASON
--------------------------------------------------------------
Tablespace [TBSALERT] is [60 percent] full
SQL>
```

Note: If your result is: no rows selected, wait a little longer and repeat the query.

c) In Enterprise Manager on the Tablespaces page, see Used (%).

Select	Name △	Allocated Size(MB)	Space Used(MB)	Allocated Space Used(%)	Auto Extend	Allocated Free Space(MB)	Status
◉	EXAMPLE	100.0	78.8	78.8	YES	21.2	✓
○	SYSAUX	590.0	556.6	94.3	YES	33.4	✓
○	SYSTEM	680.0	675.6	99.3	YES	4.4	✓
○	TBSALERT	120.0	73.0	60.8	NO	47.0	✓

d) Navigate to the Database home page. You should see the new alert in the Space Summary section. It might take several minutes for the alert to appear.

Space Summary

Database Size (GB)	1.465
Problem Tablespaces	⚠ 1
Segment Advisor Recommendations	0
Policy Violations	✓ 0
Dump Area Used (%)	87

9) In your SQL*Plus session, execute the inserts below to add more data to TBSALERT. Wait a few moments and view the critical level in both the database and Database Control. Verify that TBSALERT fullness is around 75%.

a) Execute the following commands:

```
SQL> insert into employees4 select * from employees4;
109568 rows created.
SQL> commit;
SQL> insert into employees5 select * from employees5;
109568 rows created.
SQL> commit;
SQL>
```

b) Wait a few minutes and view the critical level in both the database and Database Control. Verify that TBSALERT fullness is around 75%. In SQL*Plus, enter:

```
SQL> select sum(bytes) *100 /125829120
from dba_extents
where tablespace_name='TBSALERT';

SUM(BYTES)*100/125829120
------------------------
                      75
```

c) Check the outstanding alrets. You may need to wait a few minutes.

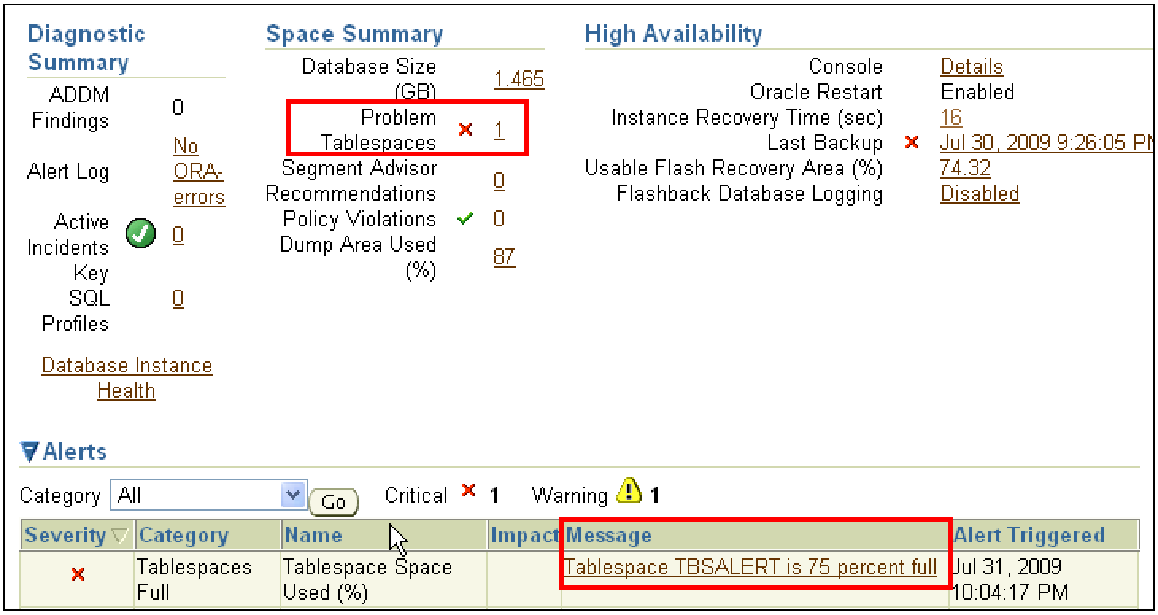
```
SQL> select reason, message_level
from dba_outstanding_alerts
where object_name='TBSALERT';

REASON                                        MESSAGE_LEVEL
----------------------------------------      -------------

Tablespace [TBSALERT] is [75 percent] full    1
```

d) In Enterprise Manager,navigate to Server > Tablespaces page, amd review Used (%).

Select	Name △	Allocated Size(MB)	Space Used(MB)	Allocated Space Used(%)	Auto Extend	Allocated Free Space(MB)	Status
◉	EXAMPLE	100.0	78.8	78.8	YES	21.2	✓
○	SYSAUX	590.0	558.4	94.6	YES	31.6	✓
○	SYSTEM	680.0	675.6	99.4	YES	4.4	✓
○	TBSALERT	120.0	91.0	75.8	NO	29.0	✓

e) Navigate to the Database home page. You should see the new alert in the Space Summary region. It will take several minutes for the change in status to take effect. Note the red flag instead of the yellow one.

Practice 18-1: Managing Storage (continued)



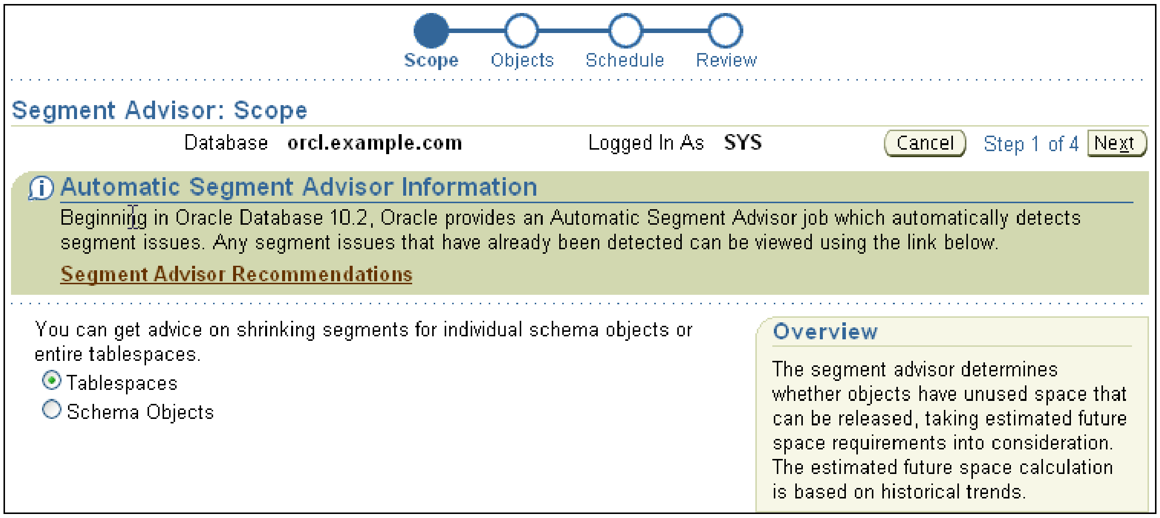
10) In your SQL*Plus session, execute the following delete statements to delete rows from tables in TBSALERT. These statements will take several minutes to complete. Then exit your SQL*Plus session.

```
SQL> delete employees1;
219136 rows deleted.
SQL> commit;
Commit complete.
SQL> delete employees2;
219136 rows deleted.
SQL> commit;
Commit complete.
SQL> delete employees3;
219136 rows deleted
SQL> commit;
Commit complete.
SQL> exit
$
```

11) Now, run the Segment Advisor for the TBSALERT tablespace in Enterprise Manager. Make sure that you run the Advisor in Comprehensive mode without time limitation. Accept and implement its recommendations. After the recommendations have been implemented, check whether the fullness level of TBSALERT is below 55%.

a) From the Database home page, select Advisor Central under Related Links and then click Segment Advisor.

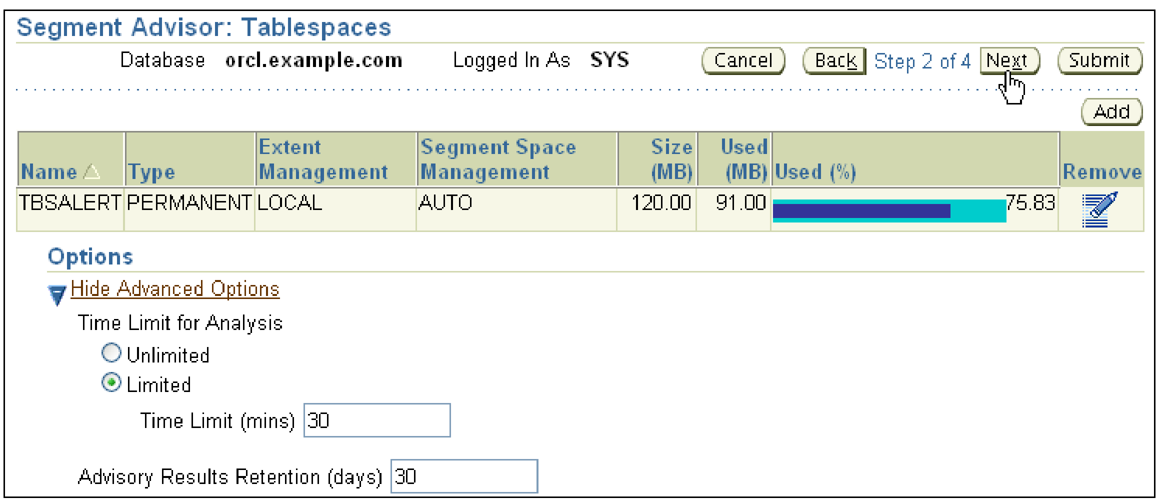
Practice 18-1: Managing Storage (continued)



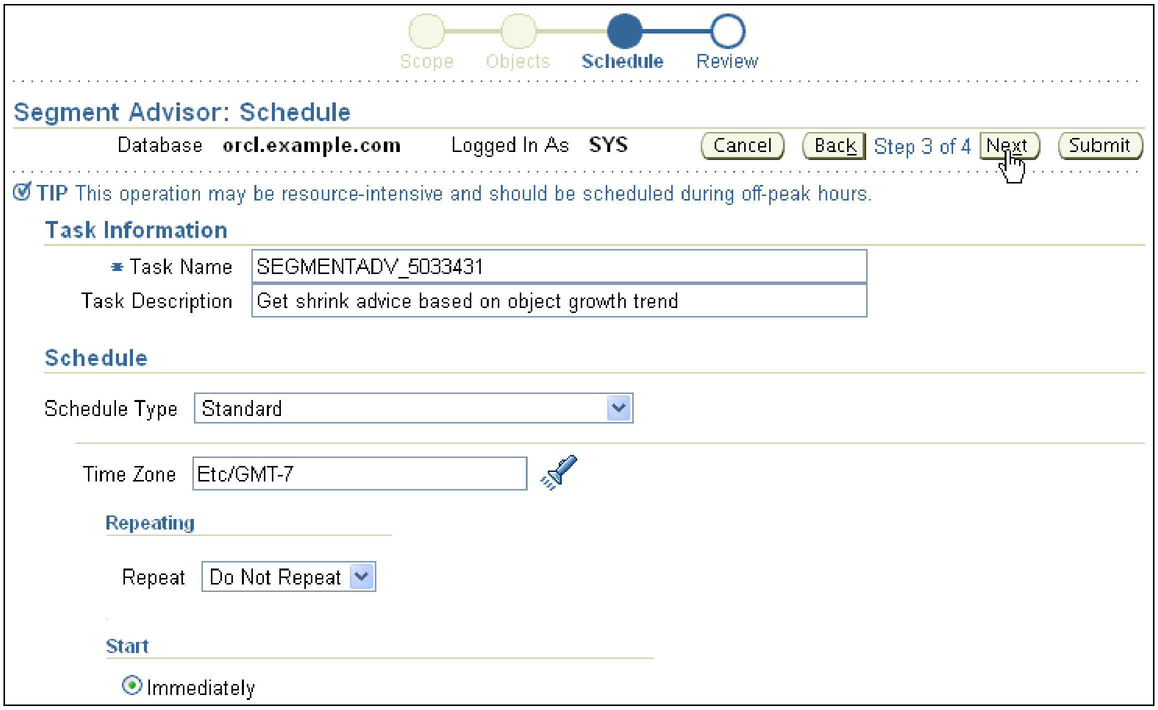
b) On the Segment Advisor: Scope page, select Tablespaces and click Next.

c) On the Segment Advisor: Objects page, click Add, select TBSALERT. Click OK and then click Show Advanced Options.

d) In the Options section, click Limited and enter **`30`** for Time Limit (mins)



e) Then click Next.



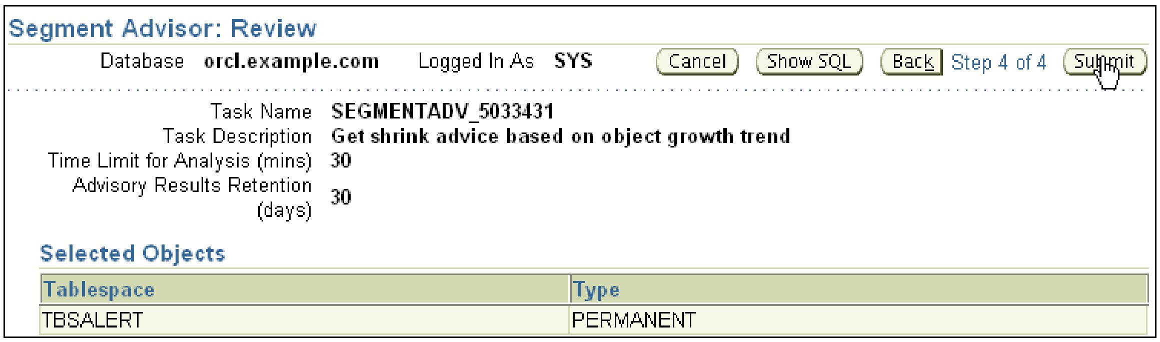
f) On the Segment Advisor: Schedule page, make sure Immediately is selected. Select your proper timezone and click Next.

g) On the Segment Advisor: Review page, click Show SQL.

```
Create task and objects script
DECLARE

taskname varchar2(100);
taskdesc varchar2(128);
task_id number;
object_id number;
timeLimit varchar2(25);
numDaysToRetain varchar2(25);
objectName varchar2(100);
objectType varchar2(100);

BEGIN
taskname := 'SEGMENTADV_5033431';
taskdesc :='Get shrink advice based on object growth trend';
numDaysToRetain :='30';
dbms_advisor.create_task('Segment Advisor',?,taskname,taskdesc
,NULL);
dbms_advisor.create_object(taskname, 'TABLESPACE', 'TBSALERT',
' ', ' ', NULL, object_id);
dbms_advisor.set_task_parameter(taskname, 'RECOMMEND_ALL',
'TRUE');
timeLimit :='1800';
dbms_advisor.set_task_parameter(taskname, 'TIME_LIMIT',
timeLimit);
```

```
dbms_advisor.set_task_parameter(taskname, 'DAYS_TO_EXPIRE',
numDaysToRetain);
END;

Execute task script
DECLARE
taskname varchar2(100);
BEGIN
taskname := 'SEGMENTADV_5033431';
dbms_advisor.reset_task(taskname);
dbms_advisor.execute_task(taskname);
END;
```

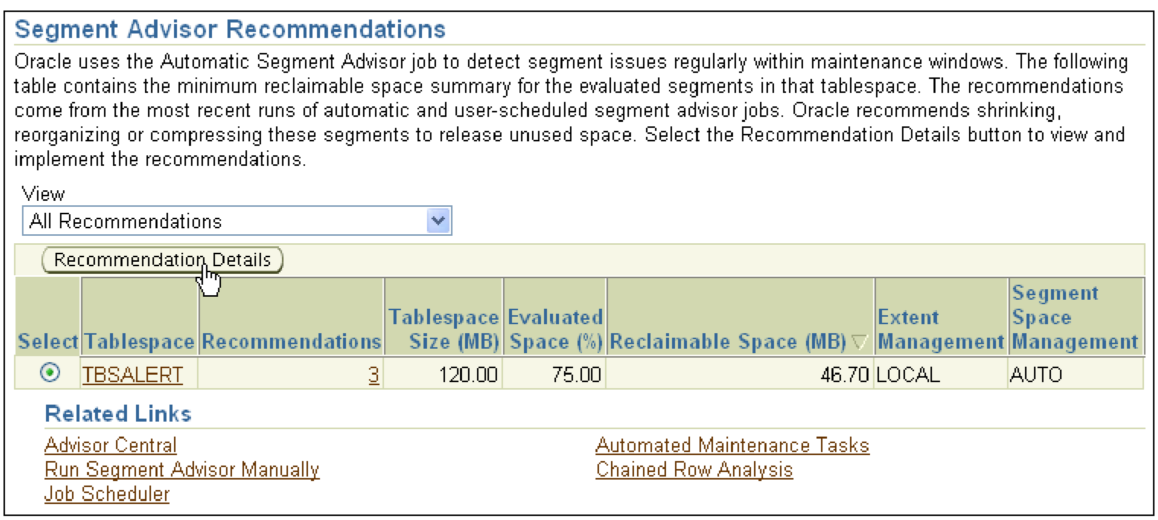
h) Review the statements and click Return.



i) Back on the Segment Advisor: review page, the Submit button.

j) This takes you back to the Advisor Central page. Click the `SEGMENTADV_`*`xxxxx`* link in the Name column.

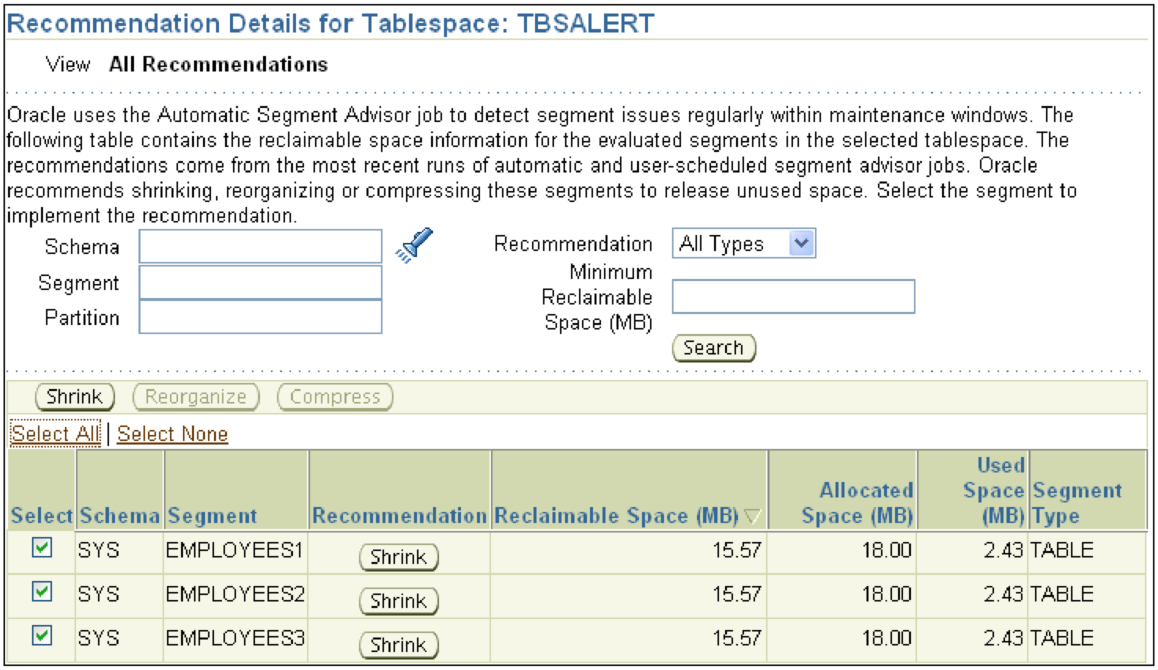
k) On the Segment Advisor Task page, click the Recommendation Details button .



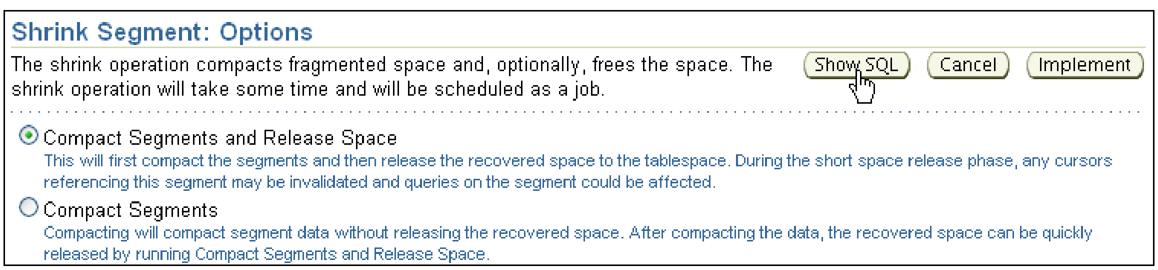
Practice 18-1: Managing Storage (continued)

l) If needed, click your browser's Refresh button until you see recommendations for the TBSALERT tablespace.

m) Click Recommendation Details.



n) Click the Select All link and then click the Shrink button.

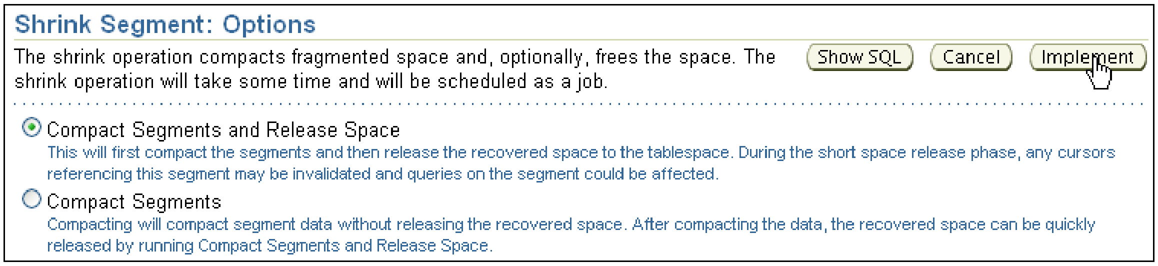


o) On the Shrink Segment: Options page, make sure that you click the "Compact Segments and Release Space" option button. Click Show SQL.

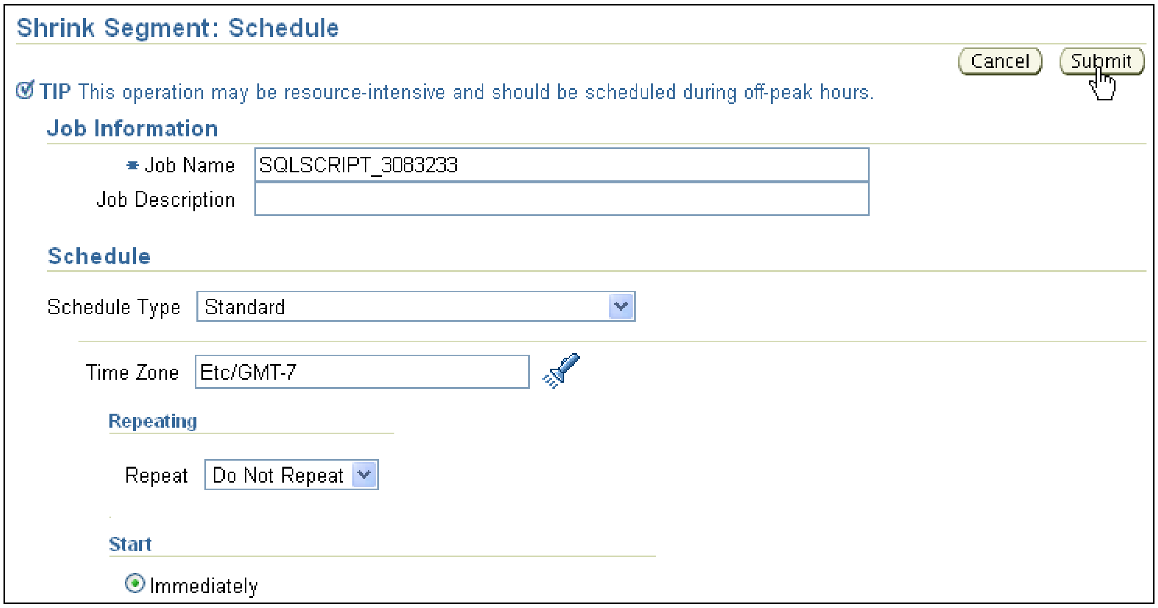
```
alter table "SYS"."EMPLOYEES1" shrink space
alter table "SYS"."EMPLOYEES2" shrink space
alter table "SYS"."EMPLOYEES3" shrink space
```

p) Review the statements and click Return.

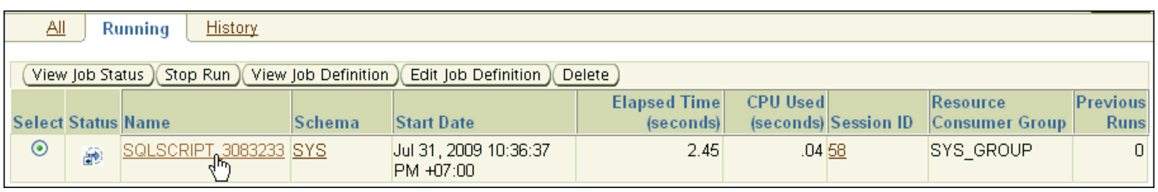
Practice 18-1: Managing Storage (continued)



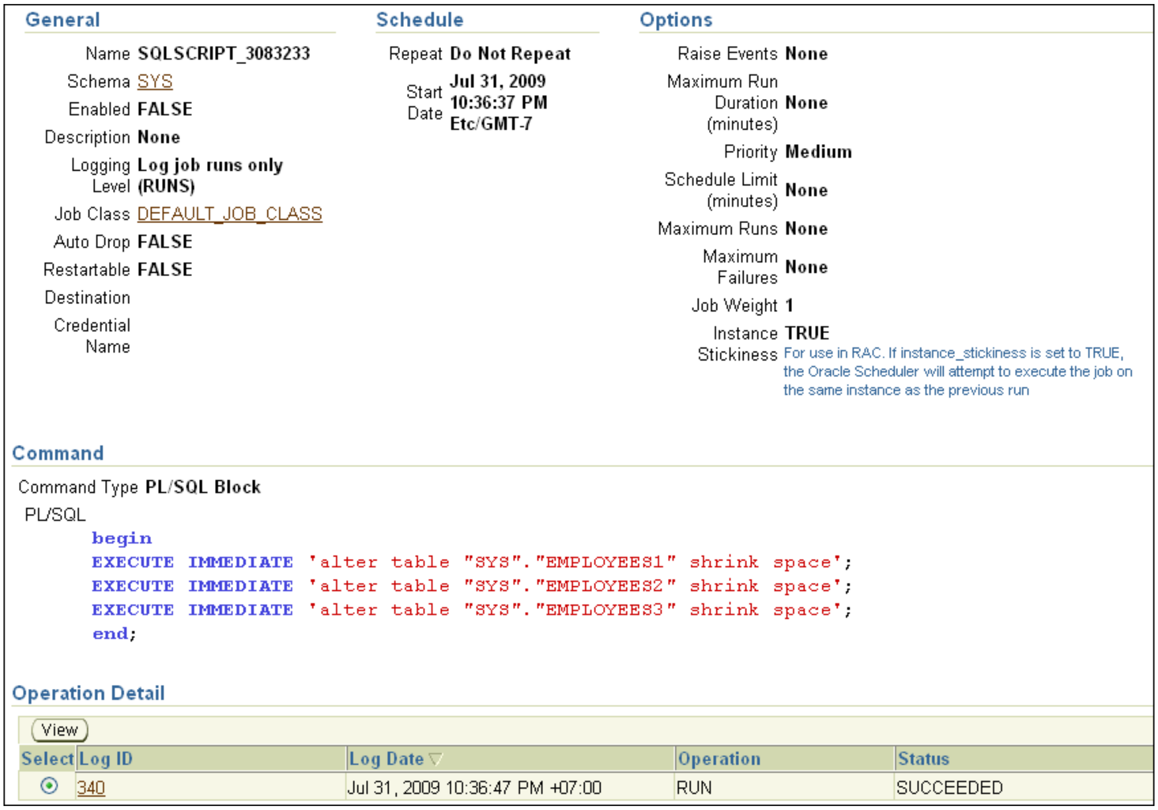
q) On the Shrink Segment: Options page, click Implement.



r) On the Shrink Segment: Schedule page, click the Submit button.

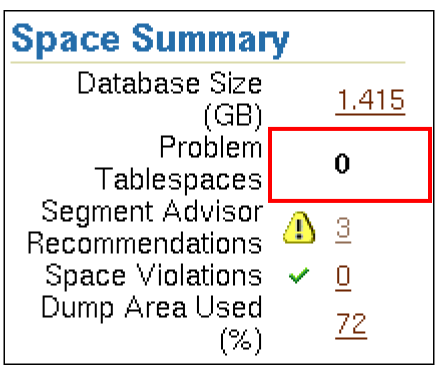


s) On the Scheduler Jobs page, click the *SQLSCRIPT_nnn* link.



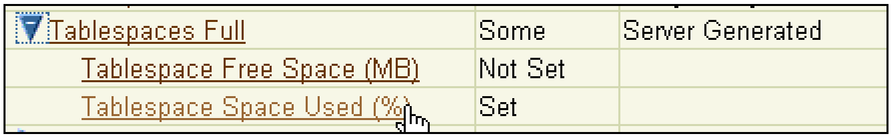
t) On the View Job page, scroll to the bottom of the page. Under Operation Detail, you should see that the job succeeded. (If it's still running, use your browser's Refresh button). Then click OK.

12) Wait a few minutes and check that there are no longer any outstanding alerts for the TBSALERT tablespace. Then navigate to the Database home page. You should see Problem Tablespaces: 0.

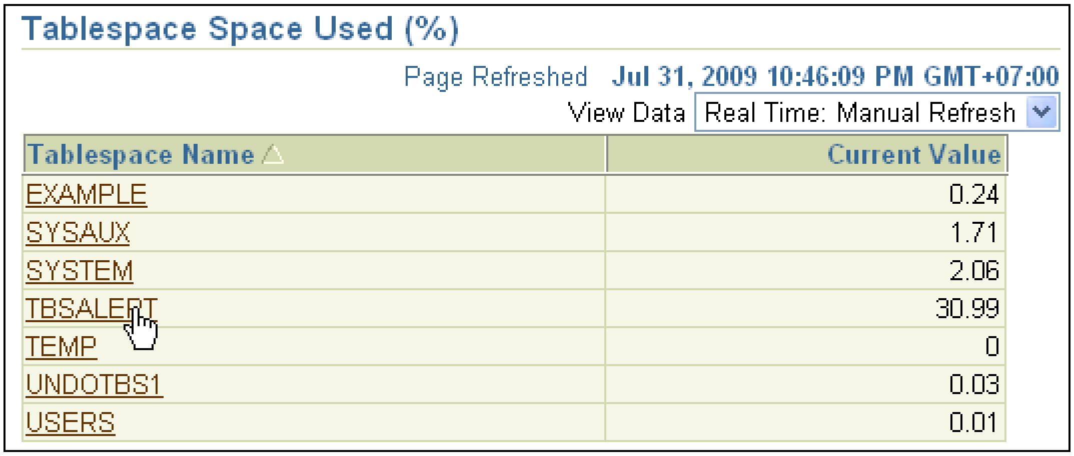


13) Retrieve the history of the TBSALERT Tablespace Space Usage metric for the last 24 hours.

a) On the Database home page, select All Metrics in the Related Links region.

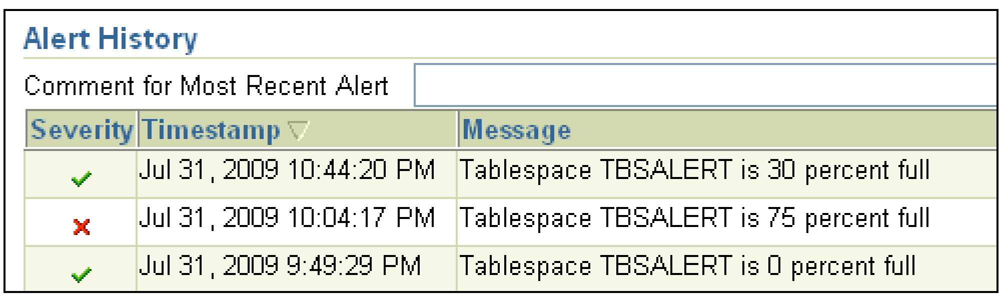


b) Expand the Tablespaces Full category, and click the Tablespace Space Used (%) link.



c) Make sure that you select Real Time: Manual Refresh from the View Data drop-down list. Then click the TBSALERT link.

d) This takes you to the Tablespace Space Used (%): Tablespace Name TBSALERT page. Select "Last 24 hours" from the View Data drop-down list.



e) View entries in the Alert History.

14) Reset the databasewide default thresholds from the Tablespace Space Usage metric for

a) On the Tablespace Space Used (%): Tablespace Name TBSALERT page, click the Edit Tablespace link in the related Links section.

b) This opens the Edit Tablespace: TBSALERT page. Click the Thresholds tab.

c) Click Use Database Default Thresholds option in the Space Used (%) section. Then click Show SQL.

Show SQL

Return

```
BEGIN DBMS_SERVER_ALERT.SET_THRESHOLD(9000,NULL,NULL,NULL,NULL,1,1,NULL,5,'TBSALERT');
END;
```

d) Review the statement and click return.

e) On the Edit Tablespace: TBSALERT, Thresholds page, click the Apply button. You should receive a success message.

15) **Note: This is your mandatory cleanup step**. Because you have finished with your test case, view and execute the `seg_advsr_cleanup.sh` script from the `labs` directory to drop your `TBSALERT` tablespace.

```
$ cat seg_advsr_cleanup.sh
#!/bin/sh
# For training only, execute as oracle OS user

sqlplus /nolog <<EOF
connect / as sysdba
alter system set disk_asynch_io = TRUE scope = spfile;
shutdown immediate;
startup
drop tablespace tbsalert including contents and datafiles;
exit
EOF
$
```

```
$ ./seg_advsr_cleanup.sh
SQL> Connected.
SQL>
System altered.

SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> ORACLE instance started.

Total System Global Area  481259520 bytes
Fixed Size                  1337324 bytes
Variable Size             385878036 bytes
Database Buffers           88080384 bytes
Redo Buffers                5963776 bytes
Database mounted.
Database opened.
SQL>
Tablespace dropped.

SQL>
$
```

Practices for Lesson 19

Physical disks with 4 KB-sectors are now available. Although this largely concerns only the operating system, the Oracle server is aware of them and uses them automatically when you create new databases.

However, as a DBA you might have databases, which were created on 512-bytes disks. In this practice, you learn about how to perform an offline migration from 512-bytes disks to 4 KB-sector disks.

Practice 19-1: Managing Space for the Database

In this practice, you view a demonstration of using 4 KB-sector disks. The focus is on performing an offline migration of redo log groups form 512-bytes to 4 KB-sector disks.

1) Click the `oracle's Home` icon on your desktop.
2) Navigate to the `/home/oracle/demos/4kb_disks` directory.
3) Double-click the `4kb_disks_viewlet_swf.html` file.
4) In the Run or Display window, click Display and view the presentation.
5) Use the controls at the bottom of the viewlet window to start, pause and stop the presentation, as suits your personal learning style.
6) Uninterrupted viewing of the demos takes about ten minutes. When you have finished viewing the presentation, close your Web browser window.

Practices for Lesson 20

Background: You are responsible for an active database that cannot be shut down. It is running in `ARCHIVELOG` mode. Now you are requested to duplicate this database, for testing purposes.

To setup a working environment for your duplicated database, you:

- Add two disks to your DATA disk group.
- Ensure that the orcl source database is in `ARCHIVELOG` mode with a 7 GB fast recovery area.
- Set up `dbtest` as the net service name for your planned `dbtest` database.
- Use Oracle Net Manager to configure the `LISTENER` with the `dbtest` and `orcl` database services.
- Clone a database by using RMAN. Start the duplication process in Enterprise Manager.
- Test access

Practice 20-1: Duplicating a Database

In this practice, you clone a database and use additional utilities to setup a working environment. To simulate this environment, assume that your active database is `orcl` (which is stored in ASM).

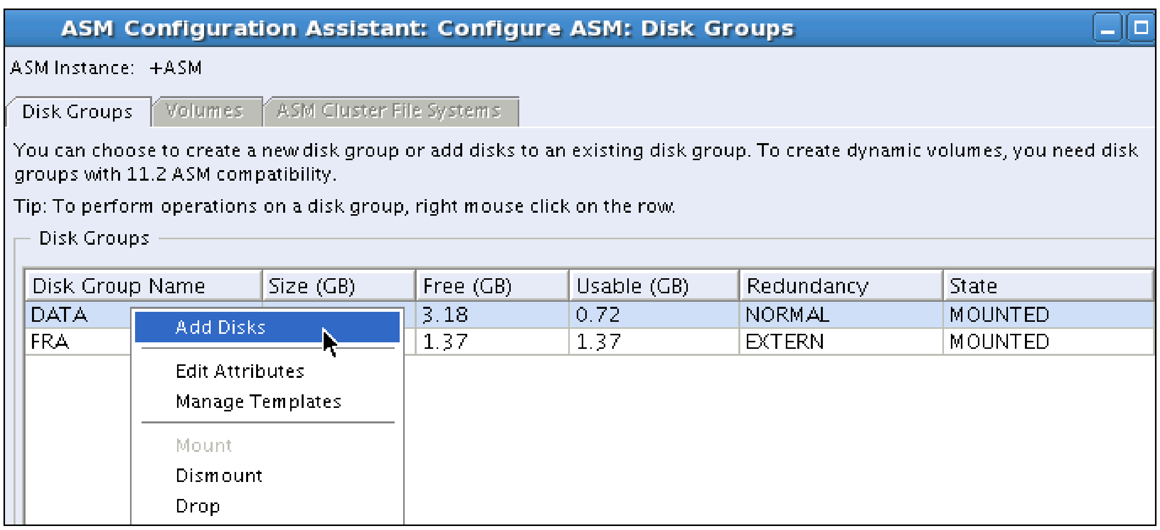
1) Before you start cloning your database, add the last two ASM disks to the DATA disk group.

   a) From a graphical terminal window, connected as user oracle, set up your environment to use the +ASM instance, and execute asmca.

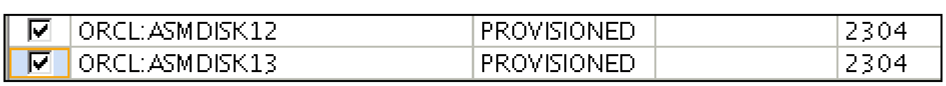
```
$ . oraenv
ORACLE_SID = [orcl] ? +ASM
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is
/u01/app/oracle
$
$ asmca
```

   b) On the Configure ASM: Disk Groups subpage, select DATA disk group.

   c) Right click.



   d) Select Add disks.



   e) On the Add Disks page, select both ORCl:ASMDISK12 and ORCL:ASMDISK13.

   f) Click OK.

Practice 20-1: Duplicating a Database (continued)

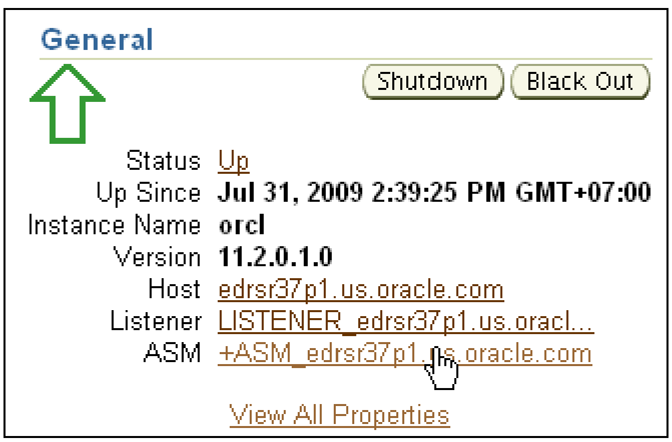
g) On the information window that appears, click OK.

h) Back to the Configure ASM: Disk Groups subpage, click Exit.

i) On the ASM Configuration Assistant window, click Yes.

2) Check the ASM activities and wait until the rebalance operation is finished:

a) Log in to Enterprise Manager as user SYS.



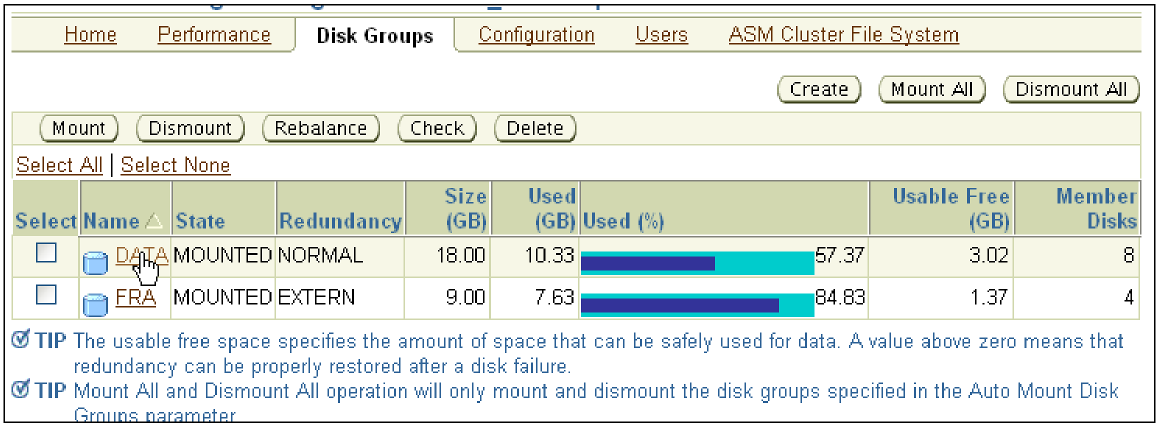
b) On the Home page, click the +ASM link in the General section.

c) On the ASM Home page, click the Disk Groups tab.

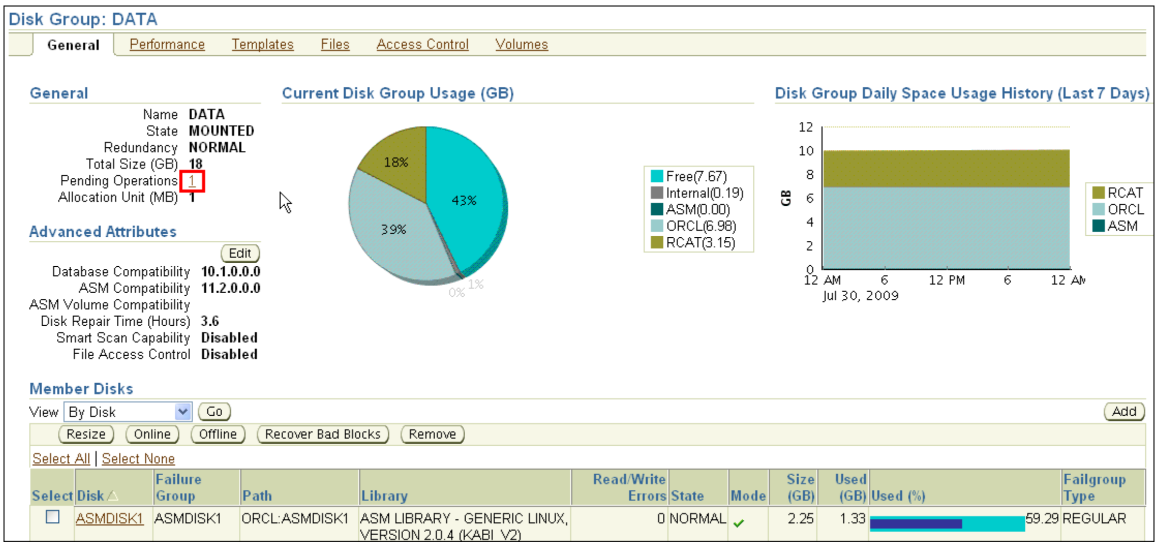
d) You may have to log in to the ASM instance if you have not previously done it and saved credentials:

- On the Automatic Storage Management Login page, enter SYS in the Username field, `oracle_4U` in the Password field, and SYSASM in the Connect As field.
- Select Save as Preferred Credential. Then click Login.

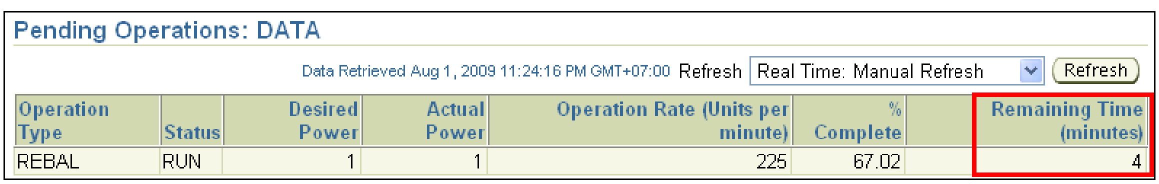
Practice 20-1: Duplicating a Database (continued)



e) On the Disk Groups sub page, click the DATA link.



f) On the Disk Group: DATA page, click at the Pending Operations field in the General section.



g) Click Refresh and **Wait until the rebalance operation is finished.**

h) Then click the database tab.

3) To ensure that you are pointing to the `orcl` database and that this database is in `ARCHIVELOG` mode with a 7 GB fast recovery area, execute the `rman_archivelog.sh` script from a terminal window in your working directory.

```
$ . oraenv
ORACLE_SID = [+ASM] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1 is
/u01/app/oracle
```

Practice 20-1: Duplicating a Database (continued)

```
$ ./rman_archivelog.sh
*******************************************
For demo purposes ONLY:
  * Enable ARCHIVELOG mode for database

The script may appear to hang at the SQL prompt
when the database is shutting down and being
opened. Wait a few minutes and it should progress.
*******************************************


SQL> SQL>
System altered.

SQL> SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> ORACLE instance started.

Total System Global Area  481259520 bytes
Fixed Size                  1337324 bytes
Variable Size             394266644 bytes
Database Buffers           79691776 bytes
Redo Buffers                5963776 bytes
Database mounted.
SQL>
Database altered.

SQL>
Database altered.

SQL> Database log mode          Archive Mode
Automatic archival              Enabled
Archive destination             USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence      15
Next log sequence to archive    17
Current log sequence            17
SQL>
$
```

4) Set up `dbtest` as the net service name for your planned `dbtest` database.

   a) In a graphical terminal window as the `oracle` user, set up your environment to point to your `orcl` instance and invoke the `netca` utility.

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
$ netca
```

   The Oracle Net Configuration Assistant (NETCA) opens a window.

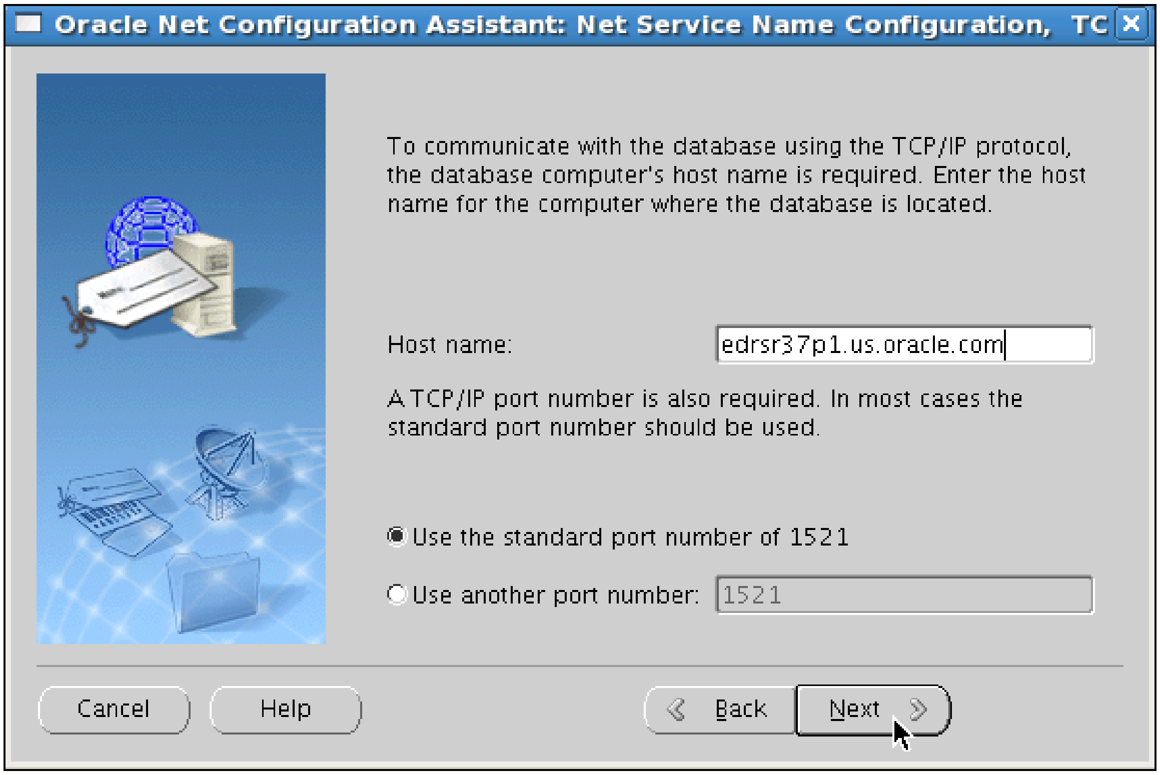
   b) On the Welcome page, select “Local Net Service Name configuration,” and click Next.

   c) On the Net Service Name Configuration page, select Add and click Next.

Practice 20-1: Duplicating a Database (continued)

d) In the Service Name field, enter `dbtest` and click Next.

e) On the Net Service Name Configuration, Select Protocols page, select TCP and click Next.



f) On the Net Service Name Configuration, TCP/IP Protocol page, enter **your** host name, for example, `edrsr37p1.us.oracle.com`, select "Use the standard port number of 1521," and then click Next.

g) On the Net Service Name Configuration, Test page, select "No, do not test" (because your `dbtest` database does not yet exist) and click Next.

h) On the Net Service Name Configuration, Net Service Name page, enter `dbtest` as Net Service Name, and then click Next.

i) Click No in answer to the question "Would you like to configure another net service name?" and then click Next.

j) When you see the completion message, click Next again.

k) Finally, click Finish.

5) Use Oracle Net Manager to configure the `LISTENER` with the `dbtest` and `orcl` database services.

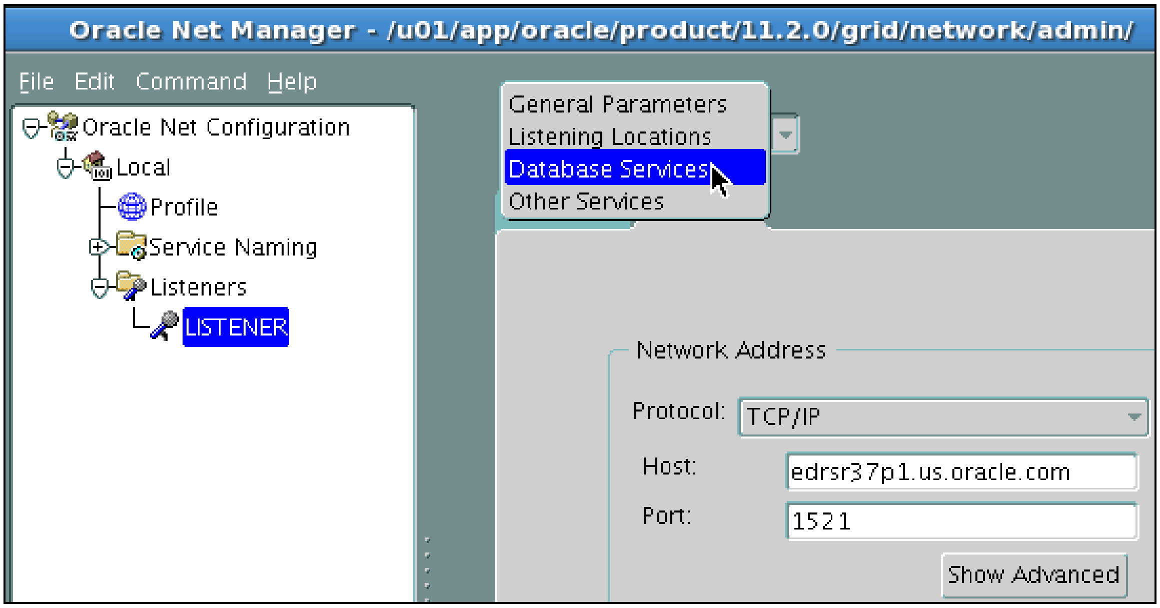
a) In a graphical terminal window as the `oracle` user, set up your environment to point to your `+ASM` instance and invoke the `netmgr` utility.

```
$ . oraenv
ORACLE_SID = [orcl] ? +ASM
```

Practice 20-1: Duplicating a Database (continued)

```
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is
/u01/app/oracle
$ netmgr
```

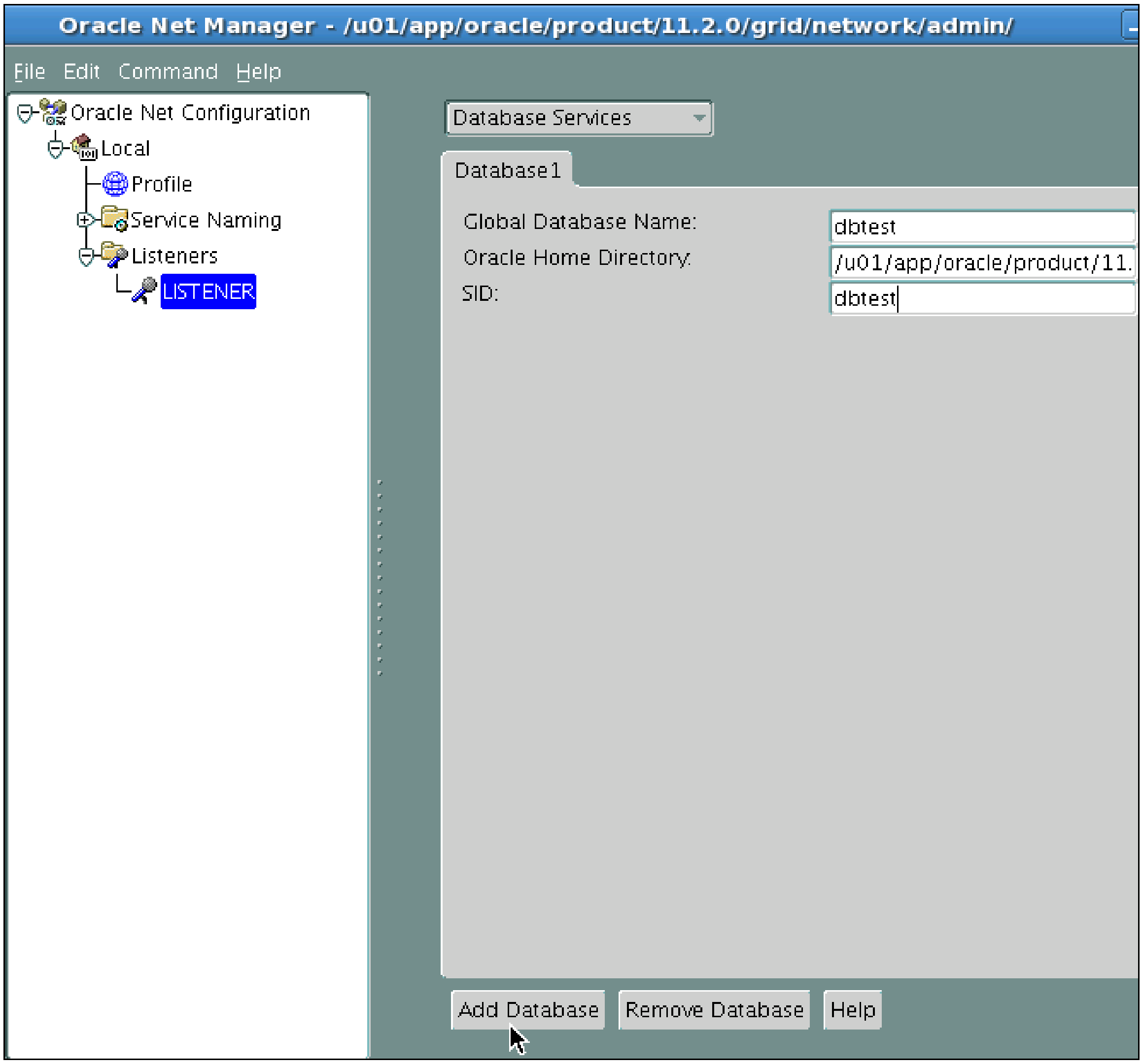
b) The Oracle Net Manager opens a window. Click the "+" icon, right before the word "Local," then click the "+" icon, right before the word "Listeners" to expand the nodes in the navigation tree until you see the listener, called "LISTENER."



c) First select LISTENER in the left part of the window then, select Database Services from the drop-down in the right part of the window.

d) Click the Add Database button.

Practice 20-1: Duplicating a Database (continued)

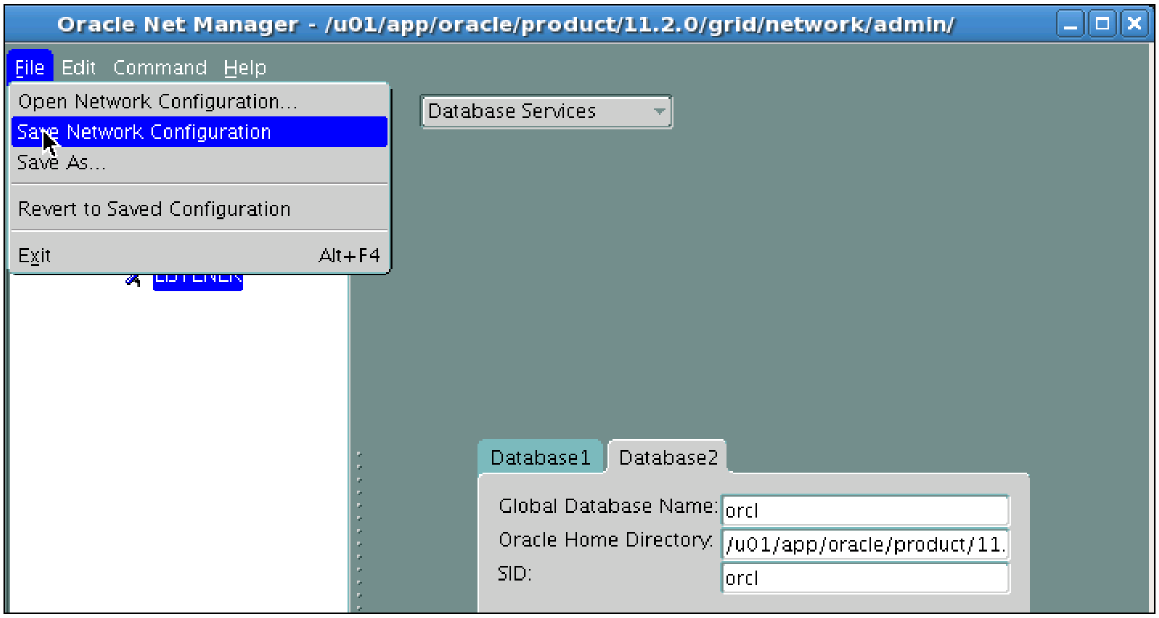


e) Enter or confirm the following values:
Global Database Name: `dbtest`
Oracle Home Directory: `/u01/app/oracle/product/11.2.0/grid/`
SID: `dbtest`

f) Click the Add Database button again.

g) Enter the following values:
Global Database Name: `orcl`
Oracle Home Directory: `/u01/app/oracle/product/11.2.0/grid/`
SID: `orcl`

Practice 20-1: Duplicating a Database (continued)

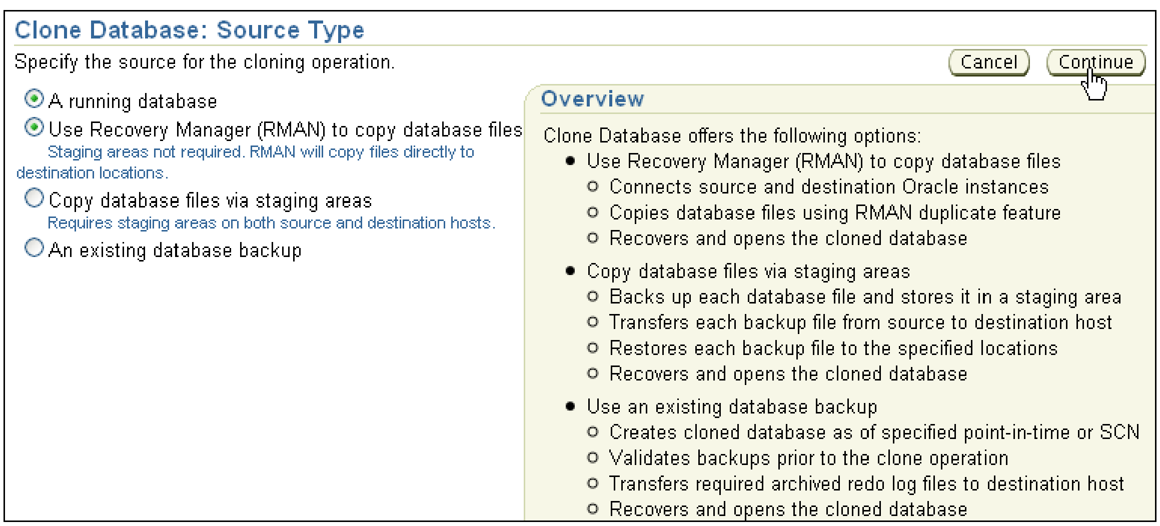


h) From the Oracle Net Manager menu bar, select File > Save Network Configuration, then File > Exit.

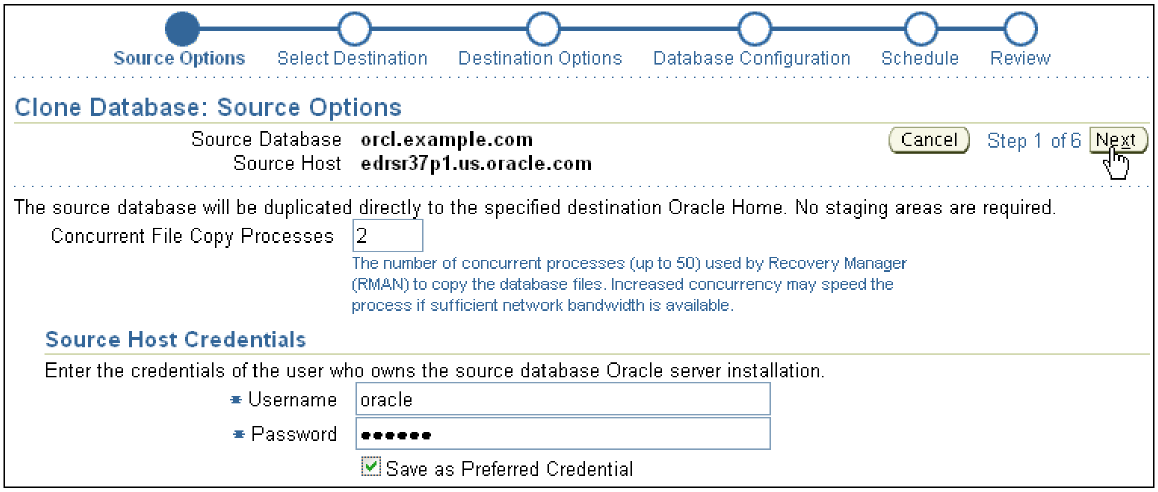
6) Clone a database by using RMAN. Start the duplication process in Enterprise Manager.

a) Log in to Enterprise Manager as the SYS user with the `oracle_4U` password and connect as SYSDBA.

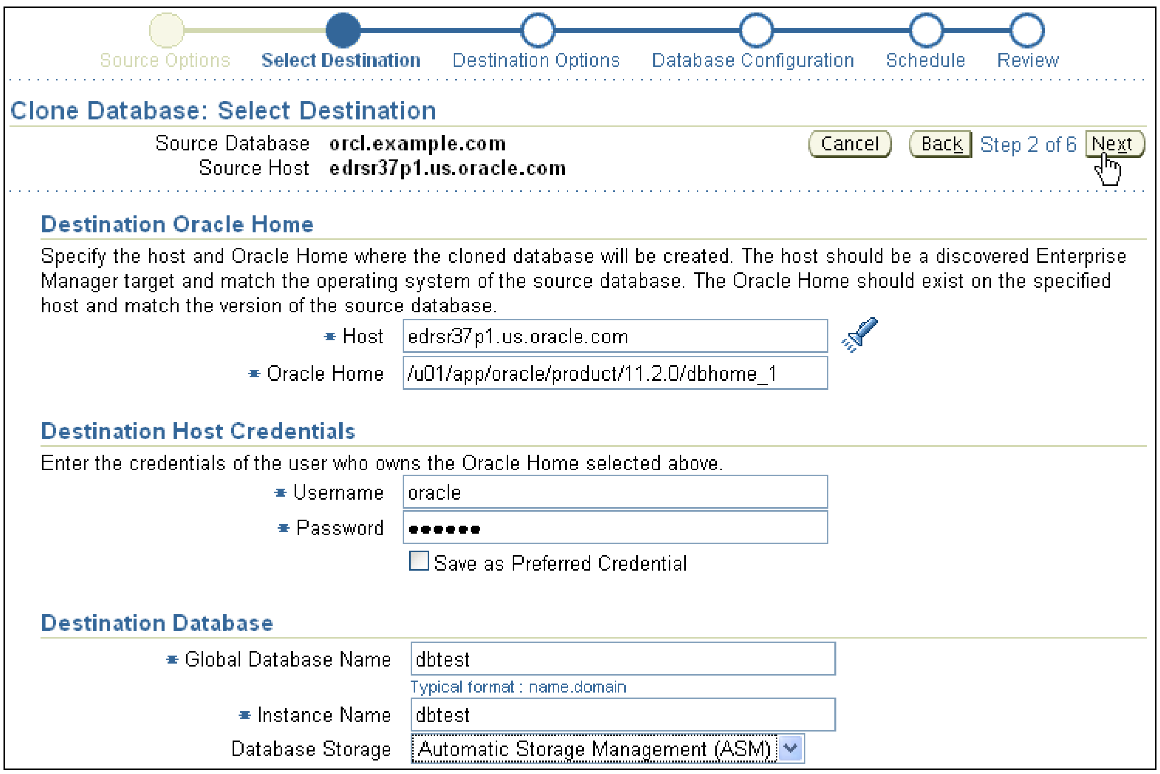
b) Navigate to Data Movement > Clone Database (in the Move Database Files section).



c) On the Clone Database: Source Type page, select “A running database” and “Use Recovery Manager (RMAN) to copy database files” and then click Continue.



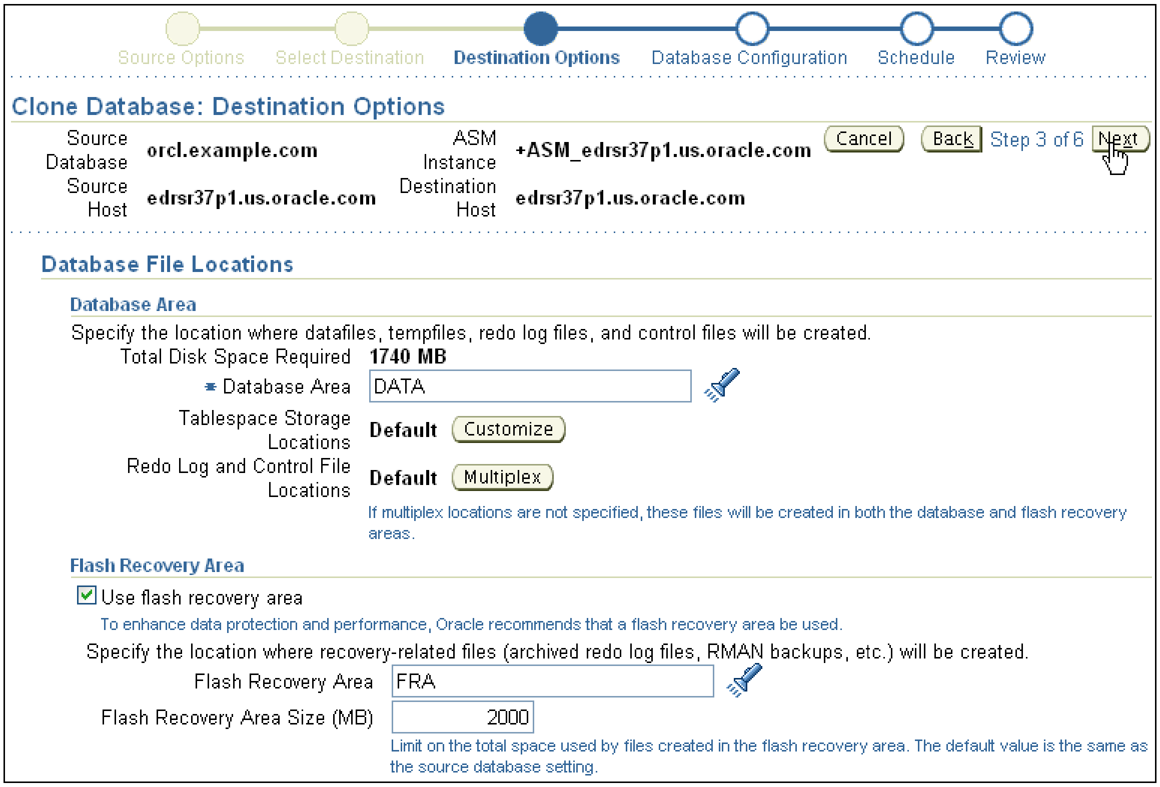
d) On the Clone Database: Source Options page, enter or confirm `oracle` as the username and password, click "Save as Preferred Credential," and then click Next.



e) On the Clone Database: Select Destination page, enter `dbtest` both as Global Database Name and as Instance Name, and select Automatic Storage Management (ASM) from the Database Storage drop down list. Then click Next.

f) If the Clone Database: ASM Instance Login page appears, enter `oracle_4U` as SYS password, and click Login.

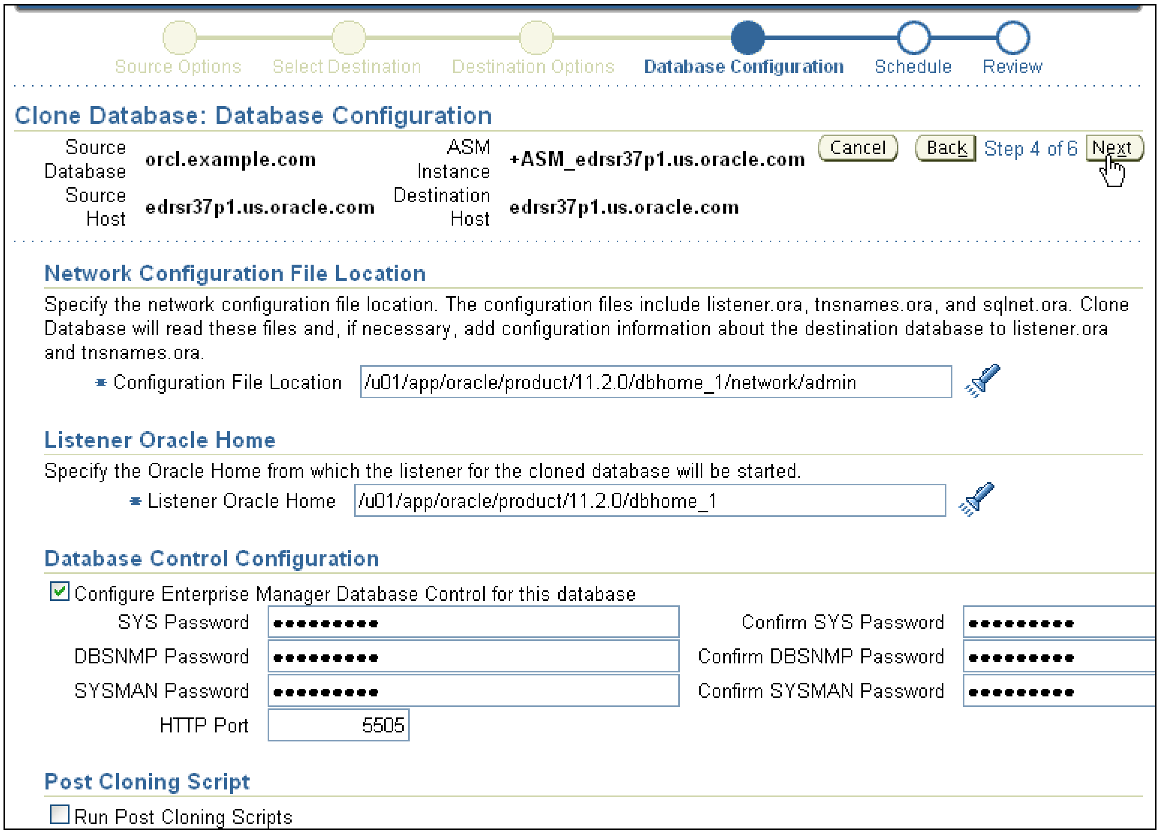
Practice 20-1: Duplicating a Database (continued)



g) On the Clone Database: Destination Options page, enter or confirm `DATA` as Database Area, `FRA` as the Flash Recovery Area, and enter `2000` as the Flash Recovery Area Size. Then click Next.

h) You receive a warming that the FRA is smaller then twice the database size. Because you plan to rarely use the FRA of this test database, click Yes to continue.

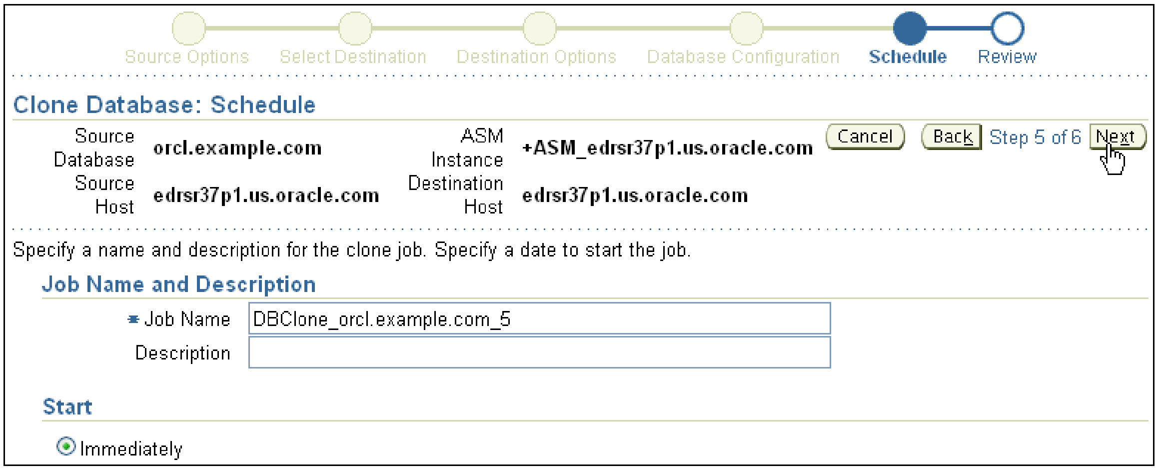
Practice 20-1: Duplicating a Database (continued)



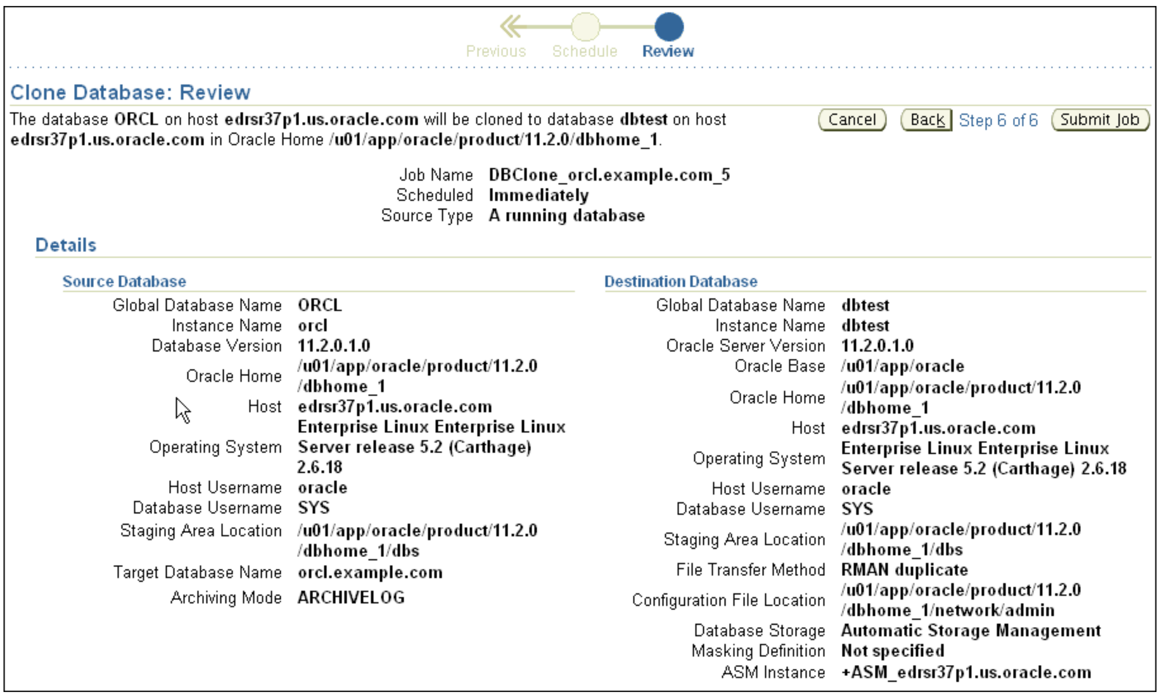
i) On the Clone Database: Database Configuration page:

- Select “Configure Enterprise Manager Database Control for this database”,
- Confirm `/u01/app/oracle/product/11.2.0/dbhome_1/network/admin` as the Configuration File Location.
- Enter or confirm `/u01/app/oracle/product/11.2.0/dbhome_1` in the Listener Oracle Home field.
- Enter `oracle_4U` six times in all password fields, and enter `5505` as HTTP port.
- Click Next.

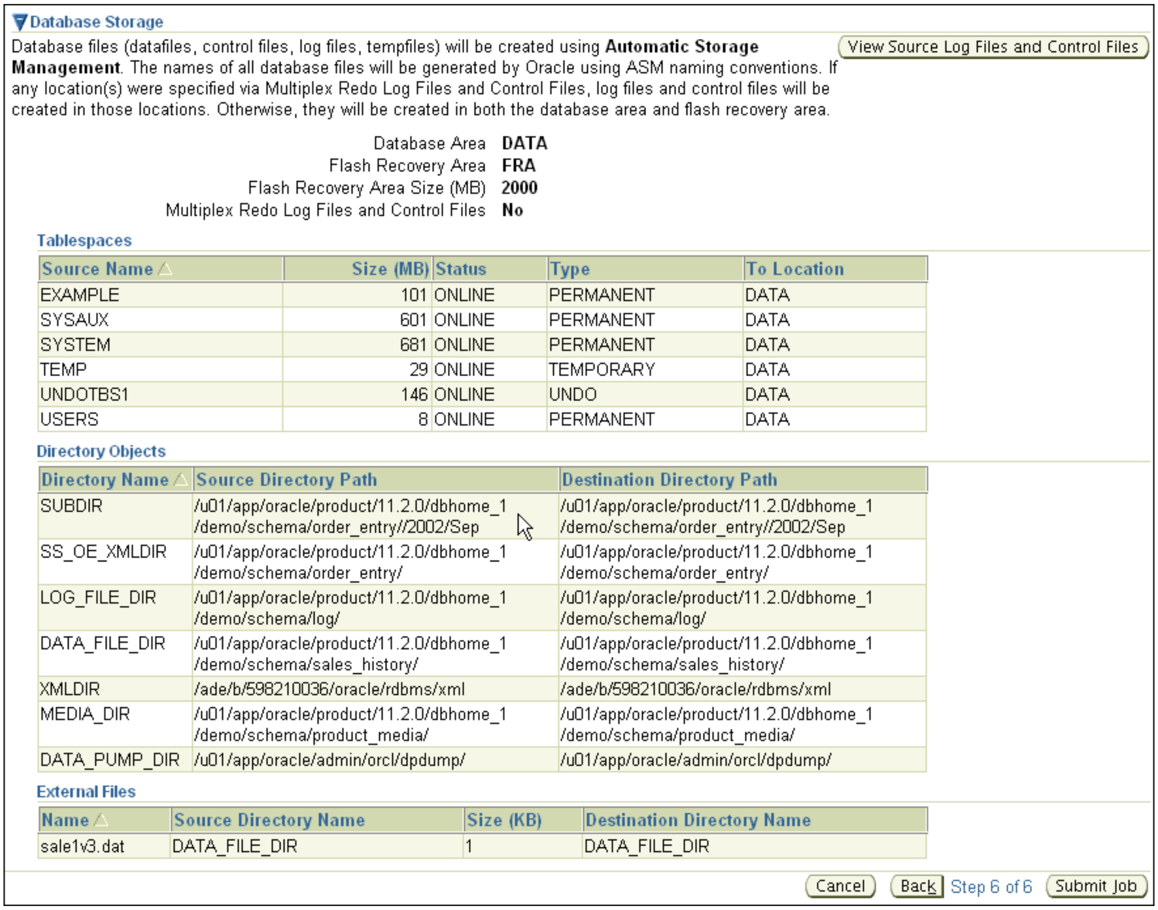
j) If you receive a warning, that the `sqlnet.ora` file or the `listener.ora` file do not exist, accept the warning by clicking Yes.



k) On the Clone Database: Schedule page, ensure that the job starts immediately and click Next.



Practice 20-1: Duplicating a Database (continued)



Database Storage

Database files (datafiles, control files, log files, tempfiles) will be created using **Automatic Storage Management**. The names of all database files will be generated by Oracle using ASM naming conventions. If any location(s) were specified via Multiplex Redo Log Files and Control Files, log files and control files will be created in those locations. Otherwise, they will be created in both the database area and flash recovery area.

View Source Log Files and Control Files

Database Area	**DATA**
Flash Recovery Area	**FRA**
Flash Recovery Area Size (MB)	**2000**
Multiplex Redo Log Files and Control Files	**No**

Tablespaces

Source Name	Size (MB)	Status	Type	To Location
EXAMPLE	101	ONLINE	PERMANENT	DATA
SYSAUX	601	ONLINE	PERMANENT	DATA
SYSTEM	681	ONLINE	PERMANENT	DATA
TEMP	29	ONLINE	TEMPORARY	DATA
UNDOTBS1	146	ONLINE	UNDO	DATA
USERS	8	ONLINE	PERMANENT	DATA

Directory Objects

Directory Name	Source Directory Path	Destination Directory Path
SUBDIR	/u01/app/oracle/product/11.2.0/dbhome_1 /demo/schema/order_entry//2002/Sep	/u01/app/oracle/product/11.2.0/dbhome_1 /demo/schema/order_entry//2002/Sep
SS_OE_XMLDIR	/u01/app/oracle/product/11.2.0/dbhome_1 /demo/schema/order_entry/	/u01/app/oracle/product/11.2.0/dbhome_1 /demo/schema/order_entry/
LOG_FILE_DIR	/u01/app/oracle/product/11.2.0/dbhome_1 /demo/schema/log/	/u01/app/oracle/product/11.2.0/dbhome_1 /demo/schema/log/
DATA_FILE_DIR	/u01/app/oracle/product/11.2.0/dbhome_1 /demo/schema/sales_history/	/u01/app/oracle/product/11.2.0/dbhome_1 /demo/schema/sales_history/
XMLDIR	/ade/b/598210036/oracle/rdbms/xml	/ade/b/598210036/oracle/rdbms/xml
MEDIA_DIR	/u01/app/oracle/product/11.2.0/dbhome_1 /demo/schema/product_media/	/u01/app/oracle/product/11.2.0/dbhome_1 /demo/schema/product_media/
DATA_PUMP_DIR	/u01/app/oracle/admin/orcl/dpdump/	/u01/app/oracle/admin/orcl/dpdump/

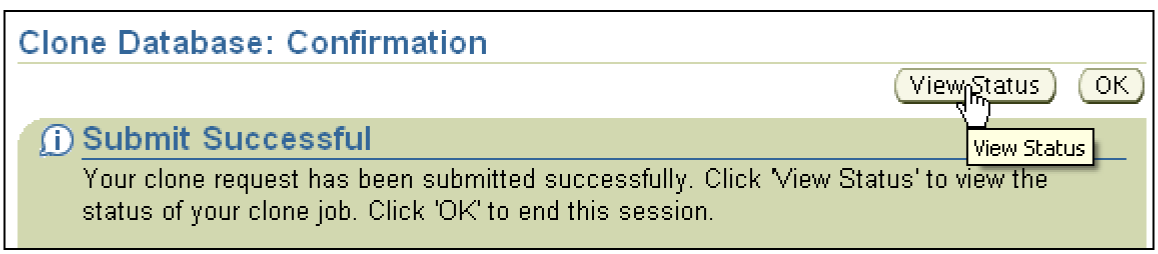
External Files

Name	Source Directory Name	Size (KB)	Destination Directory Name
sale1v3.dat	DATA_FILE_DIR	1	DATA_FILE_DIR

Cancel Back Step 6 of 6 Submit Job

l) On the Clone Database: Review page, review the Details, including the Database Storage and click Submit Job.

m) Wait on the “Clone Database job is being submitted” page.



n) The Clone Database: Confirmation page should display a success message. Click the View Status to transfer to the Job Activity page.

o) This takes you to the Execution page. Occasionally, click your browser’s reload button, to display the job progress until you receive a success message. (The execution time for your job depends on your hardware and available system resources.)

Practice 20-1: Duplicating a Database (continued)

Summary

Status	**Succeeded**
Scheduled	**Aug 2, 2009 6:53:22 PM (UTC+07:00)**
Started	**Aug 2, 2009 6:53:22 PM (UTC+07:00)**
Ended	**Aug 2, 2009 7:52:29 PM (UTC+07:00)**
Elapsed Time	**3546 seconds**
Notification	**No**

p) Scroll to the bottom of the page to see the executed steps. Some of the executed steps have output logs, for example:

- Source Preparation shows the initialization parameters of the source database.
- Destination Preparation shows `dbtest` listener and service information.
- Duplicate database contains RMAN operations.
- Recover Database shows restarting of the `dbtest` database.
- Add Temporary Files also shows output from Enterprise Manager configuration.
- Check Database and Mask data each log into the new duplicated database.

Name	Targets	Status	Started	Ended	Elapsed Time (seconds)
Execution: edrsr37p1.us.oracle.com	edrsr37p1.us.oracle.com	Succeeded	Aug 2, 2009 6:53:22 PM (UTC+07:00)	Aug 2, 2009 7:52:29 PM (UTC+07:00)	3546
Previous					
Step: Source Preparation	edrsr37p1.us.oracle.com	Succeeded	Aug 2, 2009 6:53:35 PM (UTC+07:00)	Aug 2, 2009 6:53:38 PM (UTC+07:00)	3
Step: Create Control File	edrsr37p1.us.oracle.com	Succeeded	Aug 2, 2009 6:53:45 PM (UTC+07:00)	Aug 2, 2009 6:53:46 PM (UTC+07:00)	1
Step: Destination Directories Creation	edrsr37p1.us.oracle.com	Succeeded	Aug 2, 2009 6:53:56 PM (UTC+07:00)	Aug 2, 2009 6:53:56 PM (UTC+07:00)	0
Step: Copy Initialization and Password Files	edrsr37p1.us.oracle.com	Succeeded	Aug 2, 2009 6:54:06 PM (UTC+07:00)	Aug 2, 2009 6:54:07 PM (UTC+07:00)	1
Step: Skip Copy or Transfer Controlfile	edrsr37p1.us.oracle.com	Succeeded	Aug 2, 2009 6:54:16 PM (UTC+07:00)	Aug 2, 2009 6:54:17 PM (UTC+07:00)	1
Step: Destination Preparation	edrsr37p1.us.oracle.com	Succeeded	Aug 2, 2009 6:54:26 PM (UTC+07:00)	Aug 2, 2009 6:54:34 PM (UTC+07:00)	8
Step: Duplicate Database	edrsr37p1.us.oracle.com	Succeeded	Aug 2, 2009 6:54:36 PM (UTC+07:00)	Aug 2, 2009 7:15:47 PM (UTC+07:00)	1270
Step: Skip Creating Standby Control File	edrsr37p1.us.oracle.com	Succeeded	Aug 2, 2009 7:15:56 PM (UTC+07:00)	Aug 2, 2009 7:15:58 PM (UTC+07:00)	2
Step: Skip Switching Clone Type	edrsr37p1.us.oracle.com	Succeeded	Aug 2, 2009 7:16:07 PM (UTC+07:00)	Aug 2, 2009 7:16:09 PM (UTC+07:00)	2
Step: Recover Database	edrsr37p1.us.oracle.com	Succeeded	Aug 2, 2009 7:16:17 PM (UTC+07:00)	Aug 2, 2009 7:16:45 PM (UTC+07:00)	28
Step: Add Temporary Files	edrsr37p1.us.oracle.com	Succeeded	Aug 2, 2009 7:16:48 PM (UTC+07:00)	Aug 2, 2009 7:51:31 PM (UTC+07:00)	2083
Step: Check Database and Run Post Cloning Scripts	edrsr37p1.us.oracle.com	Succeeded	Aug 2, 2009 7:51:43 PM (UTC+07:00)	Aug 2, 2009 7:51:58 PM (UTC+07:00)	14
Step: Mask data	edrsr37p1.us.oracle.com	Succeeded	Aug 2, 2009 7:52:07 PM (UTC+07:00)	Aug 2, 2009 7:52:09 PM (UTC+07:00)	2
Step: Add EM Target	edrsr37p1.us.oracle.com	Succeeded	Aug 2, 2009 7:52:17 PM (UTC+07:00)	Aug 2, 2009 7:52:19 PM (UTC+07:00)	2
Step: Cleanup Source Temporary Directory	edrsr37p1.us.oracle.com	Succeeded	Aug 2, 2009 7:52:27 PM (UTC+07:00)	Aug 2, 2009 7:52:29 PM (UTC+07:00)	2

7) Test the access to your cloned databases in SQL*Plus.

Practice 20-1: Duplicating a Database (continued)

a) Connected as the `oracle` user in a graphical terminal session, ensure that you are pointing to the `orcl` database.

```
$ . oraenv
ORACLE_SID = [+ASM] ? orcl
$
```

b) Connect as the `SYS` user to your `orcl` database and execute the following query:

`select dbid, name, created, open_mode`
`from v$database;`

```
$ sqlplus / as sysdba

SQL> select dbid, name, created, open_mode from v$database;

      DBID NAME      CREATED             OPEN_MODE
---------- --------- ------------------- --------------------
1221383234 ORCL      2009-07-30:19:21:38 READ WRITE
SQL>
```

c) Now connect as the `SYSTEM` user with the `oracle_4U` password to your `dbtest` database and execute the preceding query. Exit from SQL*Plus.

```
SQL> connect system@dbtest
Enter password: oracle_4U <<< not displayed
Connected.
SQL> select dbid, name, created, open_mode from v$database;

      DBID NAME      CREATED             OPEN_MODE
---------- --------- ------------------- --------------------
1086723993 DBTEST    2009-08-02:19:12:25 READ WRITE
SQL> exit
$
```

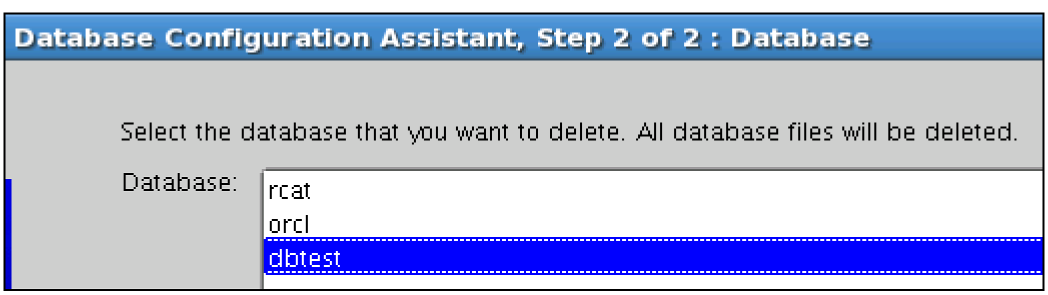
8) With the `dbca` utility, delete the `DBTEST` database because it is no longer needed

a. In a terminal window,sStart `dbca`.

```
$ dbca
```

b. Click Next on the Welcome page.

c. Choose Delete a Database on the Operations page.



d. Select `dbtest` from the list of databases to delete. Then click Finish.

Practice 20-1: Duplicating a Database (continued)

e. Confirm the delete operation by clicking Yes.

f. After the delete operation is finished, click No when asked if you want to perform another `dbca` operation.

B

Performing Tablespace Point-in-Time Recovery

Objectives

After completing this lesson, you should be able to:

- List what operations occur when you perform tablespace point-in-time recovery (TSPITR)
- Define the terminology used with TSPITR
- Identify the circumstances where TSPITR is a good solution
- Determine the correct target time for the point-in-time recovery
- Identify those situations where TSPITR cannot be used, and how to work around them
- Perform automated TSPITR

Tablespace Point-in-Time Recovery (TSPITR): Concepts

- TSPITR enables you to quickly recover one or more tablespaces to an earlier time.
- TSPITR does not affect the state of other tablespaces or objects in the database.

Tablespace Point-in-Time Recovery (TSPITR): Concepts

RMAN automatic tablespace point-in-time recovery (TSPITR) enables you to quickly recover one or more tablespaces in an Oracle database to an earlier time, without affecting the state of the other tablespaces and objects in the database.

Tablespace Point-in-Time Recovery (TSPITR): Terminology

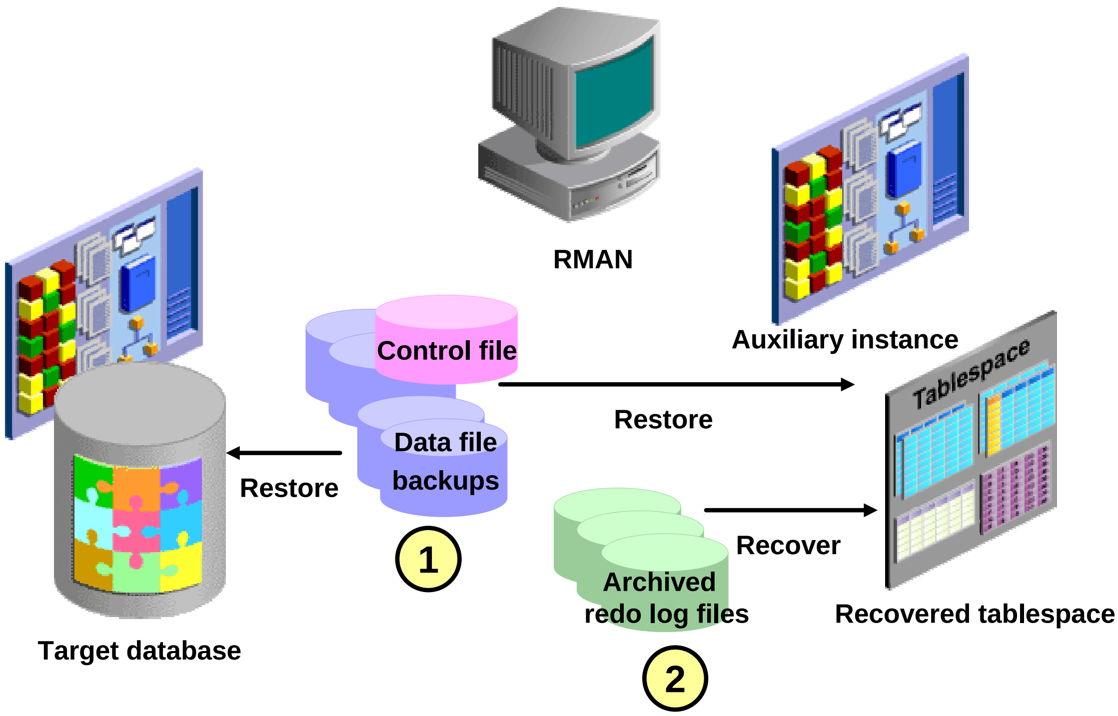
- Target time: The point in time or SCN that the tablespace will be recovered to
- Recovery set: Data files that compose the tablespaces to be recovered
- Auxiliary set: Data files required for the TSPITR of the recovery set that are not part of the recovery set. It typically includes:
  - `SYSTEM` tablespace
  - Undo segment tablespaces
  - Temporary tablespace
- Auxiliary destination: Disk location to store files

Tablespace Point-in-Time Recovery (TSPITR): Terminology

The following terminology is used when discussing TSPITR:

- **Target time:** The point in time or system change number (SCN) that the tablespace will be recovered to during TSPITR
- **Recovery set:** Data files composing the tablespaces to be recovered
- **Auxiliary set:** Data files required for TSPITR of the recovery set that are not themselves part of the recovery set. The auxiliary set typically includes:
  - A copy of the `SYSTEM` tablespace
  - Data files containing undo segments from the target instance
  - In some cases, a temporary tablespace, used during the export of database objects from the auxiliary instance
- **Auxiliary destination:** A location on disk that can be used to store any of the auxiliary set data files, control files, and online logs of the auxiliary instance during TSPITR. Files stored in the auxiliary destination can be deleted after TSPITR is complete.

Tablespace Point-in-Time Recovery: Architecture



Tablespace Point-in-Time Recovery: Architecture

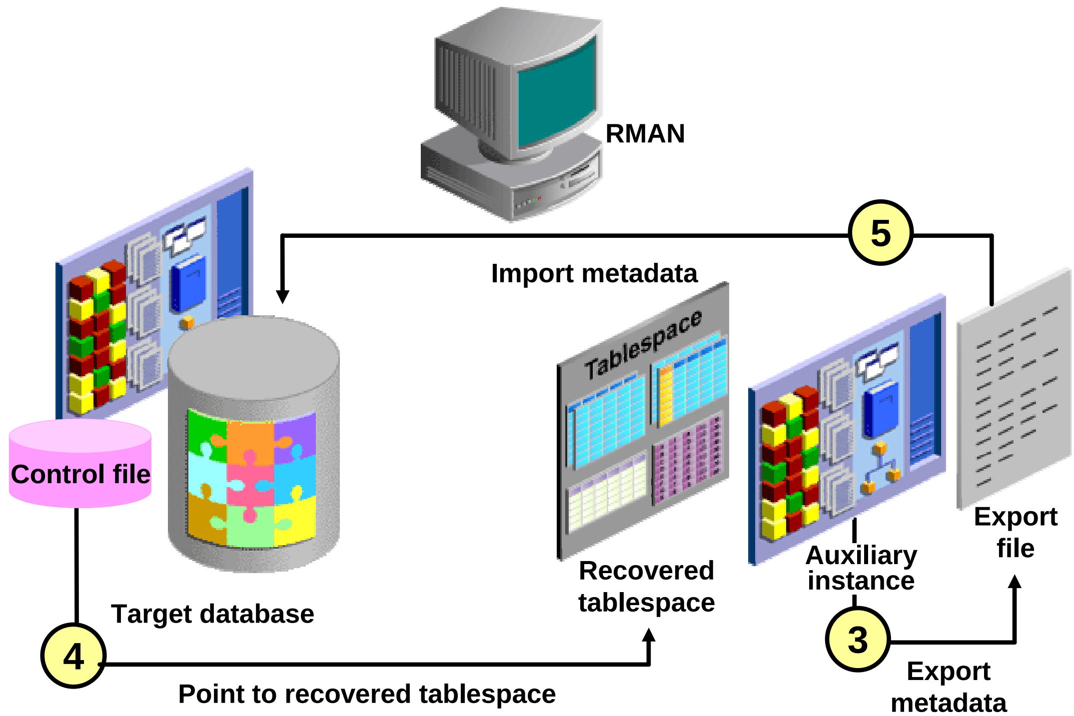
In the diagram, the following TSPITR entities are shown:

- **Target database:** Contains the tablespace to be recovered
- **Control file:** Provides backup information to RMAN
- **Backup sets:** Come from the target database and are the source of the reconstructed tablespace
- **Archived redo logs:** Come from the target database and are the source of the reconstructed tablespace
- **Auxiliary instance:** Is the Oracle database instance used during the recovery process to perform the recovery

RMAN performs the following steps during tablespace point-in-time recovery:

1. Restores a backup control file from a point in time before the target time to the auxiliary instance. It restores the data files for the *recovery set* to the target database and the data files for the *auxiliary set* to the auxiliary instance.
2. Recovers the restored data files to the specified point in time

Tablespace Point-in-Time Recovery: Architecture



Tablespace Point-in-Time Recovery Architecture (continued)

3. Exports the dictionary metadata about objects in the recovered tablespace to the target database
4. Issues `SWITCH` commands on the target database so that the target database control file points to the data files in the recovery set that were recovered on the auxiliary instance
5. Imports the dictionary metadata from the auxiliary instance to the target instance, allowing the recovered objects to be accessed

When to Use TSPITR

- TSPITR can be used in the following situations:
  - To recover data lost after an erroneous `TRUNCATE TABLE` statement
  - To recover from logical corruption of a table
  - To undo the effects of a batch job or DML statements that have affected only a part of the database
  - To recover a logical schema to a different point from the rest of the physical database
- TSPITR uses transportable tablespaces and Data Pump, providing the following new capabilities and features:
  - TSPITR can be used to recover a dropped tablespace.
  - TSPITR can be performed repeatedly to points-in-time before the tablespace was brought online without requiring a recovery catalog.

When to Use TSPITR

RMAN TSPITR can be used to:

- Recover data lost after an erroneous `TRUNCATE TABLE` statement
- Recover from logical corruption of a table
- Undo the effects of an incorrect batch job or another data manipulation language (DML) statement that has affected only a subset of the database
- Recover a logical schema to a different point in time than other parts of the physical database

Prior to Oracle Database 11*g* Release2, TSPITR used export and import for processing. TSPITR now uses transportable tablespaces and Data Pump. Because of this change to the underlying technology, TSPITR can be used to recover a dropped tablespace. In addition, TSPITR can be performed repeatedly to different points in time without the need for a recovery catalog.

Preparing for TSPITR

To prepare for TSPITR, perform the following steps:

- Determine the correct target time.
- Determine what is needed in the recovery set.
- Identify and preserve objects that will be lost after TSPITR.

Preparing for TSPITR

Before performing TSPITR, you need to determine the correct target time for your recovery. You need to determine whether you need additional tablespaces in your recovery set. You should evaluate what objects will be lost as a result of the TSPITR operation and determine how you want to preserve those objects.

Each of these steps is discussed in more detail in this lesson.

Determining the Correct Target Time

- You cannot perform TSPITR a second time unless you are using a recovery catalog.
- After you perform TSPITR and bring the tablespace online, you cannot use a backup from an earlier time.
- Use the following methods to determine the correct target time:
  - Flashback Query
  - Flashback Transaction Query
  - Flashback Version Query
- Simple alternative to TSPITR: Flash back data (if still available as undo).

ORACLE

Determining the Correct Target Time

It is extremely important that you choose the right target time or SCN for TSPITR. After you perform TSPITR and bring a tablespace online, you cannot use any backup from a time earlier than the moment you brought the tablespace online. In practice, this means that you cannot make a second attempt at TSPITR if you choose the wrong target time the first time, unless you are using a recovery catalog. However, if you have a recovery catalog, you can perform repeated TSPITR operations to different target times.

The current control file does not contain a record of an older incarnation of the recovered tablespace if you do not use a recovery catalog. Recovery with a current control file that involves the tablespace cannot use a backup taken prior to the time when you brought the tablespace online. However, you can perform incomplete recovery of the whole database to any time prior to or equal to the time when you brought the tablespace online if you can restore a backup control file from before that time.

You can use Oracle Flashback Query, Oracle Flashback Transaction Query, and Oracle Flashback Version Query to investigate changes to your database and to help determine the correct target time for TSPITR.

Note: With the Flashback tools and the data still available as undo data, it is usually much simpler to use the Flashback tools for undoing unwanted changes (rather than TSPITR).

Determining the Tablespaces for the Recovery Set

- Use the `TS_PITR_CHECK` view to identify relationships that span recovery set boundaries.
- If objects in the tablespace that you are recovering have relationships with objects in other tablespaces, you can:
  - Add the tablespace that contains the related objects to the recovery set
  - Suspend the relationship for the duration of TSPITR
  - Remove the relationship
- Use the `DBMS_TTS.TRANSPORT_SET_CHECK` procedure to determine whether the tablespaces in the recovery set are self-contained.

Determining the Tablespaces for the Recovery Set

If you are unsure whether you have accounted for all objects that have relationships with the objects in the tablespaces you are performing the TSPITR operation for, you can use the `TS_PITR_CHECK` view to identify any additional objects. When you query this view, information about any objects that will prevent you from proceeding with TSPITR is displayed. The reason why tablespace point-in-time recovery cannot proceed is displayed in the `REASON` column of the `TS_PITR_CHECK` view.

As an example, if you are planning to perform TSPITR for the `USERS` and `EXAMPLE` tablespaces, execute the following query to determine whether there are any relationships with objects in other tablespaces that are not accounted for:

```
SELECT * FROM SYS.TS_PITR_CHECK
WHERE (TS1_NAME IN ('USERS','EXAMPLE')
   AND TS2_NAME NOT IN ('USERS','EXAMPLE'))
OR (TS1_NAME NOT IN ('USERS','EXAMPLE')
AND TS2_NAME IN ('USERS','EXAMPLE'));
```

Refer to the *Oracle Database Backup and Recovery User's Guide* for additional examples using the `TS_PITR_CHECK` view.

Identifying Relationships That Span Recovery Set Boundaries

```
SELECT *
FROM SYS.TS_PITR_CHECK
WHERE (
   TS1_NAME IN ('USERS','EXAMPLE')
   AND TS2_NAME NOT IN ('USERS','EXAMPLE'))
OR (
   TS1_NAME NOT IN ('USERS','EXAMPLE')
   AND TS2_NAME IN ('USERS','EXAMPLE'));
```

Use `DBMS_TTS.TRANSPORT_SET_CHECK` to ensure that TSPITR will be successful:

```
DBMS_TTS.TRANSPORT_SET_CHECK ('USERS', 'EXAMPLE');
SELECT *  FROM TRANSPORT_SET_VIOLATIONS;
```


Identifying Relationships Between Objects That Span the Recovery Set Boundaries

Before performing TSPITR, you must determine the recovery set. If objects in the tablespaces you need to recover have relationships with objects in other tablespaces, you need to make provisions for those objects.

Prior to Oracle Database 11*g* Release 2, you used the `SYS.TS_PITR_CHECK` view to identify relationships between objects that span the recovery set boundaries. Now you should use the `DBMS_TTS.TRANSPORT_SET_CHECK` procedure and query the `TRANSPORT_SET_VIOLATIONS` view.

Note: RMAN TSPITR automatically executes the `DBMS_TTS.TRANSPORT_SET_CHECK` procedure for the recovery set tablespaces and verifies that the query against `TRANSPORT_SET_VIOLATIONS` returns no rows. If the query returns rows, RMAN stops TSPITR processing and any tablespace containment violations must be resolved before TSPITR can proceed. You can execute the procedure and query the view as described above as a precautionary measure.

Identifying Objects That Will Be Lost

- Objects created in the tablespace after the target recovery time are lost.
- Query `TS_PITR_OBJECTS_TO_BE_DROPPED` to determine which objects will be lost following TSPITR.
- Use Export prior to TSPITR and Import following TSPTIR to preserve and re-create the lost objects.

Identifying Objects That Will be Lost

Query the `TS_PITR_OBJECTS_TO_BE_DROPPED` view to determine whether there are any objects that will be lost as a result of performing tablespace point-in-time recovery.

As an example, you are performing TSPITR for the `USERS` and `EXAMPLE` tablespaces to the target time of April 3, 2006 at 8:30:00 AM. Issue the following query to determine whether there are any objects that will be lost following your TSPITR:

```
SELECT OWNER, NAME, TABLESPACE_NAME,
TO_CHAR(CREATION_TIME, 'YYYY-MM-DD:HH24:MI:SS')
FROM TS_PITR_OBJECTS_TO_BE_DROPPED
WHERE TABLESPACE_NAME IN ('USERS','EXAMPLE')
AND CREATION_TIME >
    TO_DATE('2006-APR-03:08:30:00','YY-MON-DD:HH24:MI:SS')
ORDER BY TABLESPACE_NAME, CREATION_TIME;
```

Performing Basic RMAN TSPITR

- Fully automated TSPITR
  - Specify an auxiliary destination.
  - RMAN manages all aspects of TSPITR.
  - This is the recommended method.
- Customized TSPITR with an automatic auxiliary instance
  - This is based on fully automated TSPITR.
  - Customize the location of files.
  - Specify initialization parameters.
  - Specify channel configurations.
- TSPITR using your own auxiliary instance
  - Configure and manage the auxiliary instance.

ORACLE

Performing Basic RMAN TSPITR

You have the following options when performing TSPITR:

- **Fully automated TSPITR:** Specify an auxiliary destination, and RMAN manages all aspects of the TSPITR operation. This is the simplest way to perform TSPITR, and is recommended unless you specifically need more control over the location of recovery set files after TSPITR or auxiliary set files during TSPITR, or control over the channel configurations or some other aspect of your auxiliary instance.
- **Customized TSPITR with an automatic auxiliary instance:** TSPITR is based on the behavior of fully automated TSPITR, possibly still using an auxiliary destination. You can customize one or more aspects of the behavior, such as the location of auxiliary set or recovery set files. You can specify initialization parameters or channel configurations for the auxiliary instance created and managed by RMAN.
- **TSPITR with your own auxiliary instance:** Set up, start, stop, and clean up the auxiliary instance used in TSPITR. In addition, you can manage the TSPITR process by using some of the methods available in customized TSPITR with an automatic auxiliary instance.

Performing Fully Automated TSPITR

1. Configure channels required for TSPITR on the target instance.
2. Specify the auxiliary destination using the `AUXILIARY DESTINATION` option.

```
RMAN> CONNECT TARGET
RMAN> RECOVER TABLESPACE users, example
    > UNTIL TIME '2007-06-29:08:00:00'
    > AUXILIARY DESTINATION
    > '/u01/app/oracle/oradata/aux';
```

3. Back up the recovered tablespaces and bring them online.

Performing Fully Automated TSPITR

In addition to the preparation requirements discussed earlier in the lesson, when you perform fully automated TSPITR, you must:

- Configure any channels required for TSPITR on the target instance
- Specify a destination for RMAN to use for the auxiliary set of data files and other auxiliary instance files

After TSPITR has completed, back up the recovered tablespaces and bring them online. You cannot use backups of any tablespaces that participate in TSPITR taken prior to TSPITR after you perform TSPITR.

Note: This time format assumes that `NLS_DATE_FORMAT` is set to `'yyyy-mm-dd:hh24:mi:ss'` and `NLS_LANG` is set to `AMERICAN_AMERICA.WE8MSWIN1252`.

Using Image Copies for Faster TSPITR Performance

```
CONFIGURE AUXNAME FOR DATAFILE
    '$ORACLE_BASE/oradata/orcl/users01.dbf'
TO '/backup/users01.dbf';

RECOVER TABLESPACE users UNTIL SEQUENCE 1300 THREAD 1;
```

`CONFIGURE AUXNAME` is replaced by `SET NEWNAME` for recovery set data files:

```
RUN
{
SET NEWNAME FOR DATAFILE
    '$ORACLE_BASE/oradata/orcl/users01.dbf'
TO '/backup/users01.dbf';

RECOVER TABLESPACE users UNTIL SEQUENCE 1300 THREAD 1;
}
```


Using Image Copies for Faster RMAN TSPITR Performance

You can improve TSPITR performance by directing RMAN to use the existing image copies of the recovery set and auxiliary set data files. This technique enables RMAN to skip restoring the data files from a backup.

Prior to Oracle Database 11*g* Release 2, you used the following techniques to specify the existence of an image copy of a data file:

- `CONFIGURE AUXNAME` command for image copies of recovery set data files or auxiliary set data files
- `SET NEWNAME` command for image copies of auxiliary set data files

Now you should use the `CONFIGURE AUXNAME` command only with image copies of auxiliary set data files. You should use the `SET NEWNAME` command for image copies of recovery set data files or auxiliary set data files.

Using Enterprise Manager to Perform TSPITR

Using Enterprise Manager to Perform TSPITR

You can also use Enterprise Manager to perform TSPITR. Navigate to Availability > Perform Recovery. In the User Directed Recovery section, select Tablespaces from the Recovery Scope drop-down menu.

There are three operations you can perform, for tablespaces:

- **Recover to current time or a previous point in time:** Restores the data files for the tablespace, if needed. This operation then uses redo to recover to the time you specify: either the current time or a time in the past. This is the combination of the following two operations.
- **Restore tablespaces:** Only restore the data files for the tablespace. No recovery is performed.
- **Recover from previously restored tablespaces:** Perform recovery (redo application) only, of the tablespace's data files.

RMAN TSPITR Processing

RMAN performs the following steps:

1. Creates the auxiliary instance, starts it, and connects to it
2. Takes the tablespaces that will be recovered offline
3. Restores a backup control file from a point in time before the target time to the auxiliary instance
4. Restores the data files from the recovery set and the auxiliary set to the auxiliary instance
5. Recovers the restored data files to the specified time
6. Opens the auxiliary database with the `RESETLOGS` option

RMAN TSPITR Processing

The steps that RMAN carries out to perform TSPITR are listed in this slide and the next.

RMAN TSPITR Processing

7. Exports the dictionary metadata about objects in the recovered tablespaces to the target database
8. Shuts down the auxiliary instance
9. Imports the dictionary metadata from the auxiliary instance to the target instance
10. Deletes all auxiliary set files

RMAN TSPITR Processing (continued)

After RMAN completes the last step, the TSPITR process is complete. The recovery set data files are returned to the state they were in at the specified target time.

Performing RMAN TSPITR with an RMAN-Managed Auxiliary Instance

- Rename or relocate your recovery set data files.
- Specify a location other than the auxiliary destination for some or all of the auxiliary set data files.
- Create image copy backups of your data files in advance of TSPITR.
- Use a different channel configuration for the auxiliary instance.
- Specify different initialization parameters for your RMAN-managed auxiliary instance.

ORACLE

Performing RMAN TSPITR with an RMAN-Managed Auxiliary Instance

If you want to customize RMAN TSPITR, you can use an RMAN-managed auxiliary instance and make the following changes:

- Rename the recovery set data files using `SET NEWNAME` so that they are not restored and recovered in their original locations.
- Control the location of your auxiliary set data files by specifying new names for individual files with `SET NEWNAME` and using `DB_FILE_NAME_CONVERT` to provide rules for converting data file names in the target database to data file names for the auxiliary database.
- Use existing image copies of the recovery set and auxiliary set data files on disk rather than restoring them from backup for faster RMAN TSPITR performance.

Note: Refer to the *Oracle Database Backup and Recovery User's Guide* for additional information.

Performing RMAN TSPITR Using Your Own Auxiliary Instance

- Not recommended, but supported
- Perform the following steps:
  1. Create an Oracle password file for the auxiliary instance.
  2. Create an initialization parameter file for the auxiliary instance.
  3. Verify Oracle Net connectivity to the auxiliary instance.
  4. Start the auxiliary instance in `NOMOUNT` mode.
  5. Connect the RMAN client to the target and auxiliary instances.
  6. Execute the `RECOVER TABLESPACE` command.

Performing RMAN TSPITR Using Your Own Auxiliary Instance

Oracle recommends that you allow RMAN to manage the creation and destruction of the auxiliary instance used during RMAN TSPITR. However, creating and using your own auxiliary instance is supported.

To create an Oracle instance suitable for use as an auxiliary instance, perform the following steps:

1. Create an Oracle password file for the auxiliary instance by using the `orapwd` utility.
2. Create a text initialization parameter file for the auxiliary instance.
3. Verify Oracle Net connectivity to the auxiliary instance using a valid net service name.

To perform TSPITR, complete the following steps:

4. Start the auxiliary instance in `NOMOUNT` mode.
5. Connect the RMAN client to target and auxiliary instances.
6. Execute the `RECOVER TABLESPACE` command.

Refer to the *Oracle Database Backup and Recovery User's Guide* for a detailed example.

Troubleshooting RMAN TSPITR

- File name conflicts: Ensure that there are no name conflicts when using `SET NEWNAME`, `CONFIGURE AUXNAME`, and `DB_FILE_NAME_CONVERT`.
- RMAN cannot identify tablespaces with undo segments: Use the `UNDO TABLESPACE` clause.
- Restarting a manual auxiliary instance after TSPITR failure: Shut down and restart in `NOMOUNT` mode.

Troubleshooting RMAN TSPITR

File name conflicts: If your use of `SET NEWNAME`, `CONFIGURE AUXNAME`, and `DB_FILE_NAME_CONVERT` cause multiple files in the auxiliary or recovery sets to have the same name, you receive an error during TSPITR. To correct the problem, specify different values for these parameters to eliminate the duplicate name.

RMAN cannot identify tablespaces with undo segments: During TSPITR, RMAN needs information about which tablespaces had undo segments at the TSPITR target time. This information is usually available in the recovery catalog, if one is used. If there is no recovery catalog, or if the information is not found in the recovery catalog, RMAN proceeds assuming that the set of tablespaces with undo segments at the target time is the same as the set of tablespaces with undo segments at the present time. If this assumption is not correct, the TSPITR operation fails and an error is reported. To prevent this from happening, provide a list of tablespaces with undo segments at the target time in the `UNDO TABLESPACE` clause.

Restarting manual auxiliary instance after TSPITR failure: If you are managing your own auxiliary instance and there is a failure in TSPITR, then before you can retry TSPITR, you must shut down the auxiliary instance, correct the problem, and put the auxiliary instance back in `NOMOUNT` mode.

Summary

In this lesson, you should have learned how to:

- List what operations occur when you perform tablespace point-in-time recovery (TSPITR)
- Define the terminology used with TSPITR
- Identify the circumstances where TSPITR is a good solution
- Determine the correct target time for the point-in-time recovery
- Identify those situations where TSPITR cannot be used, and how to work around them
- Perform automated TSPITR

ORACLE

Performing User-Managed Backup and Recovery

Objectives

After completing this lesson, you should be able to:

- Describe the difference between user-managed and server-managed backup and recovery
- Perform user-managed complete database recovery
- Perform user-managed incomplete database recovery

Types of Backup and Recovery Practices

Types of database backup and recovery are:

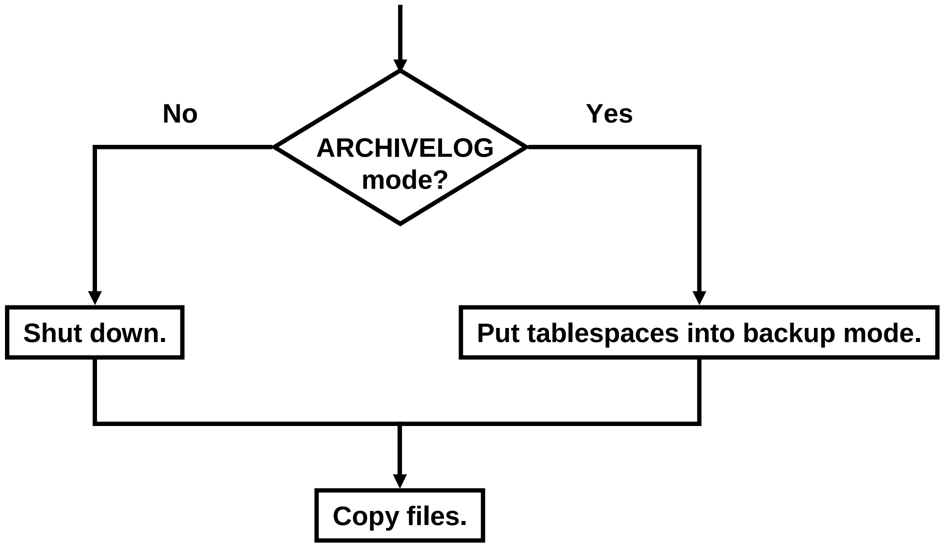
- User-managed: Does not use RMAN.
  - Uses OS commands to move files around
  - Requires DBA to manually maintain backup activity records
- Server-managed: Uses RMAN

Types of Backup and Recovery Practices

There are two methods you can use to recover your database. You can use RMAN, and take advantage of its automatic recovery capabilities. It can restore the appropriate files and bring the database back to a current state by using very few commands. You can also recover manually. This is called *user-managed recovery*. User-managed recovery entails moving the files around using OS commands, and then issuing the recovery commands in SQL*Plus.

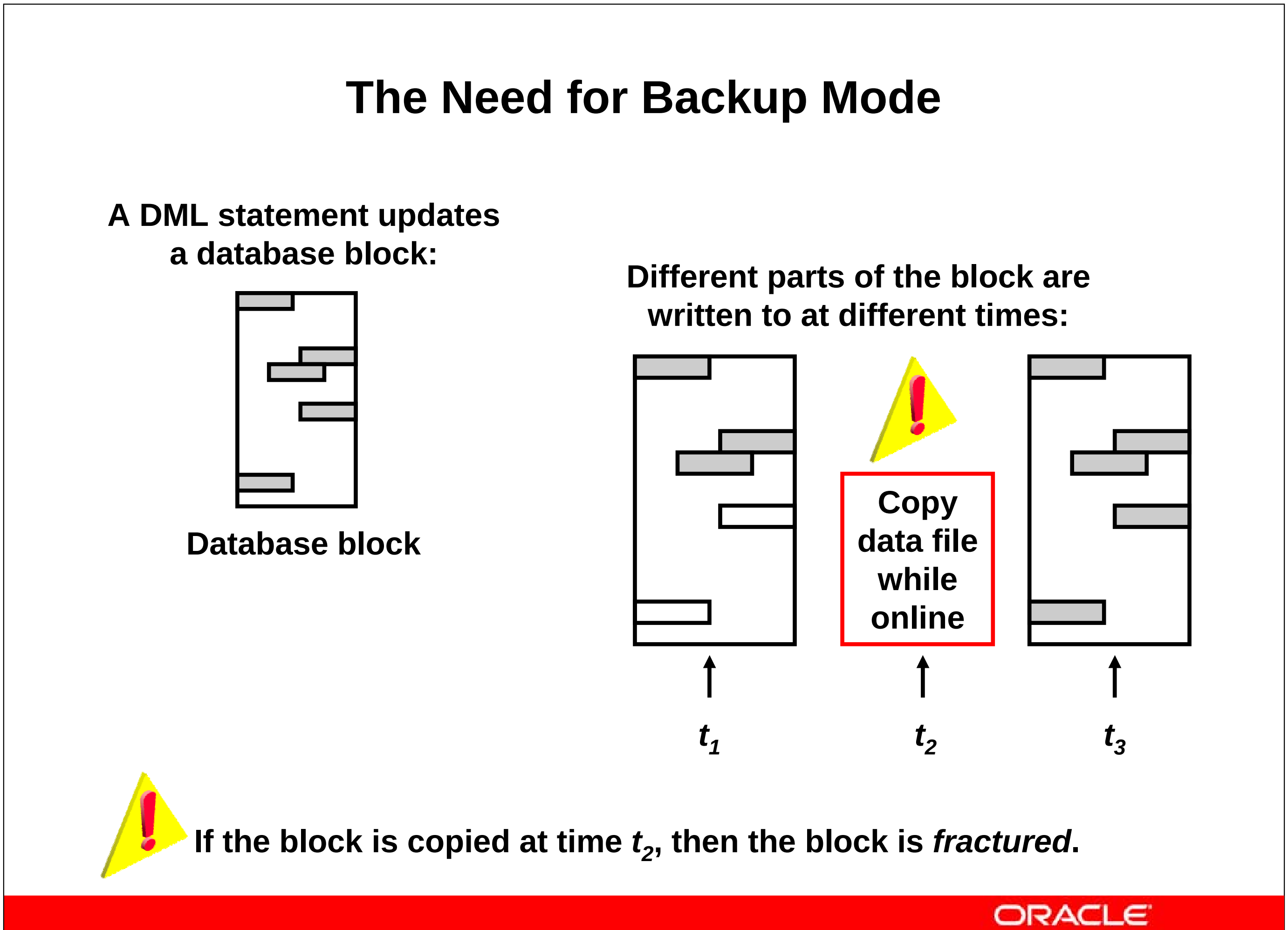
Both of these methods use restore and recovery processes.

Performing a User-Managed Backup of the Database



Performing a User-Managed Backup of the Database

You can back up the database by using OS commands to make copies of the data files. The course of action depends on whether the database is in `ARCHIVELOG` mode or not. If it is, then you can keep the database open and available by putting each tablespace into backup mode before copying its data files. Otherwise, you have to shut down the database before copying the data files.



The Need for Backup Mode

When a block is being written to as part of the execution of a data manipulation language (DML) statement, there could be several parts of the block affected. Not all of the modifications to the block happen at the same time, so there is the possibility of inconsistency in the block at certain times. Suppose time t_2 represents the time between when different parts of the block are written to. If, at time t_2, the block is copied as part of the execution of an OS copy command, then the block is considered fractured. Also, the OS copy command does not necessarily copy the file header first, so it must be frozen for the duration of the copy execution.

RMAN has the means to deal with this problem. If a fractured block is read, it keeps rereading it until it is consistent.

However, if an OS command such as the Linux `cp` command is copying the data file, the fractured block is not recognized as such, and the copy of the block is not consistent. In order to remedy this, put the tablespace, or even the entire database, into what is called *backup mode*. The effect of doing this is that additional redo is generated. An image of each block, before it is modified, is written to the redo log. Then, during recovery of blocks in that data file, the before image of a fractured block can be used for the basis of recovery and the additional redo data is applied to it. In order to reduce the overhead associated with maintaining extra redo data, Oracle recommends putting one tablespace at a time into backup mode, while its data files are being copied.

Identifying Files to Manually Backup

```
SQL> select name from v$datafile;

NAME
------------------------------------------------------------------------
/u01/app/oracle/oradata/ORCL/datafile/o1_mf_system_36mky81f_.dbf
/u01/app/oracle/oradata/ORCL/datafile/o1_mf_sysaux_36mky81p_.dbf
/u01/app/oracle/oradata/ORCL/datafile/o1_mf_undotbs1_36mky857_.dbf
/u01/app/oracle/oradata/ORCL/datafile/o1_mf_users_36mky876_.dbf
/u01/app/oracle/oradata/ORCL/datafile/o1_mf_example_36ml2cmh_.dbf
/u01/app/oracle/oradata/ORCL/datafile/survey01.dbf
```

```
SQL> select name from v$controlfile;

NAME
------------------------------------------------------------------------
/u01/app/oracle/oradata/ORCL/controlfile/o1_mf_36ml1f8x_.ctl
/u01/app/oracle/flash_recovery_area/ORCL/controlfile/o1_mf_36ml1fkk_.ctl
```

ORACLE

Identifying Files to Manually Backup

User-managed backups require you to know the data file names and locations on disk, so you know what files need to be copied. Identify the data files to be backed up by querying the `V$DATAFILE` view. Identify the control file location by querying the `V$CONTROLFILE` view. Only one of the multiplexed control files needs to be backed up, because they are identical.

Manually Backing Up a NOARCHIVELOG Database

- Shut down the database instance:

```
SQL> SHUTDOWN IMMEDIATE
```

- Copy the data files to the backup location:

```
$ cp $ORACLE_BASE/ORCL/datafile/*.dbf \
> /u02/backup/datafile
```

- Copy the control files to the backup location:

```
$ cp $ORACLE_BASE/ORCL/controlfile/*.ctl \
> /u02/backup/controlfile
```

- Start the instance and open the database:

```
SQL> STARTUP
```


Manually Backing Up a NOARCHIVELOG Database

You can make a consistent, whole database backup of a NOARCHIVELOG database by shutting down the database and copying all the data files and the control file to a backup directory. Because the action of copying the files is done using OS commands (in this case, the Linux cp command), the database must be shut down first. This puts it in a consistent state. If your database is running in NOARCHIVELOG mode, this is your only option. Otherwise, in ARCHIVELOG mode, you can make inconsistent backups, which allows you to leave the database running while you take the backup.

Manually Backing Up an ARCHIVELOG Database

- Identify tablespaces and their data files:

```
SQL> select file_name, tablespace_name from dba_data_files;
FILE_NAME                                     TABLESPACE_NAME
--------------------------------------------- ---------------
/u01/app/oracle/oradata/orcl/users01.dbf      USERS
/u01/app/oracle/oradata/orcl/users02.dbf      USERS
/u01/app/oracle/oradata/orcl/undotbs1.dbf     UNDOTBS1
/u01/app/oracle/oradata/orcl/sysaux01.dbf     SYSAUX
/u01/app/oracle/oradata/orcl/system01.dbf     SYSTEM
/u01/app/oracle/oradata/orcl/example01.dbf    EXAMPLE
```

For each tablespace:

- Put the tablespace into backup mode:

```
SQL> ALTER TABLESPACE users BEGIN BACKUP;
```

- Copy the data files for that tablespace to the backup location:

```
$ cp $ORACLE_HOME/oradata/orcl/users*.dbf /u02/backup/datafile
```

- Bring the tablespace out of backup mode:

```
SQL> ALTER TABLESPACE users END BACKUP;
```

ORACLE

Manually Backing Up an ARCHIVELOG Database

If the database is in ARCHIVELOG mode, then you do not necessarily have to shut it down before copying the files. You end up with an inconsistent backup, but the application of redo data recovers it to a consistent state.

Starting Backup Mode: You do have to put each of the data files into backup mode before copying them, though. Do this using the BEGIN BACKUP clause of the ALTER TABLESPACE and ALTER DATABASE commands. Here is the syntax for each:

```
ALTER TABLESPACE <tablespace> BEGIN BACKUP;
ALTER DATABASE BEGIN BACKUP;
```

The ALTER TABLESPACE command affects only those data files that belong to that tablespace. ALTER DATABASE affects all data files in the database.

Ending Backup Mode: It is important to bring the data files out of backup mode. You cannot have any data files in backup mode at the time the database is shut down. If you attempt to shut down the database in that state, you will receive an error. Also, because backup mode causes additional redo to be generated, there is extra load on the system. There is no reason to have any data files in backup mode if you are not actively backing them up.

Note: In addition, you need to archive out the current redo log files, and back them up safely as well.

Backing Up the Control File

Back up the control file:

- As an image copy, to a specifically named file:

```
SQL> ALTER DATABASE BACKUP CONTROLFILE TO
  2> '/u01/backup/controlfile.bak';

Database altered.
```

- By generating a script that re-creates it, in a trace file:

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

Database altered.
```


Backing Up the Control File

You should back up the control file every time you make a structural change to the database. Use one of the commands shown in the slide to do this. The first command creates a binary copy of the file. You can optionally supply the REUSE keyword if the backup file already exists and you want to overwrite it.

The second command makes a plain text version of the control file, which is actually a script that creates the control file when run. The resulting script is written to the diagnostics trace directory, such as:

```
$ORACLE_BASE/diag/rdbms/orcl/orcl/trace
```

You can also specify a name for the trace file by using the AS 'filename' clause.

Performing User-Managed Complete Database Recovery: Overview

User-managed complete database recovery:

- Recovers the database to the most recent SCN
- Can be done with the entire database at once, or a data file or tablespace at a time
- Requires a current or backup control file
- Requires backups of all files to be recovered
- Requires all archive logs up to the present

Performing Complete Database Recovery: Overview

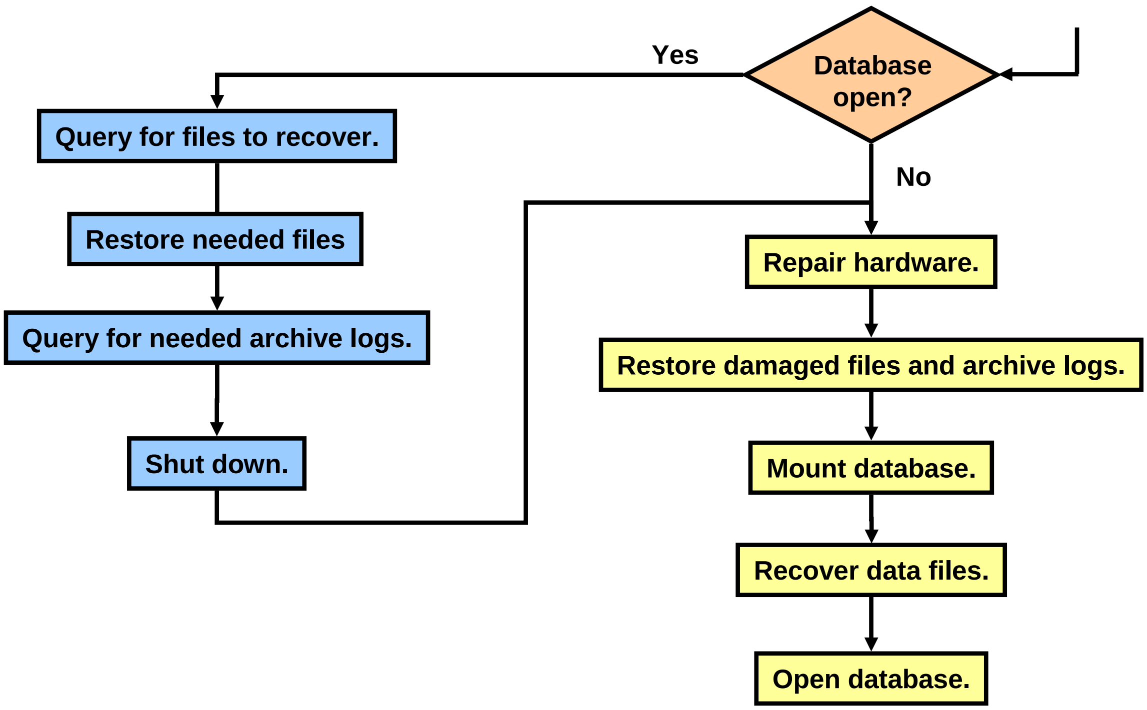
Complete database recovery brings the database back to its most current state. You can recover the entire database, or a single tablespace or data file at a time. You must have a current or backup control file in order to perform complete database recovery. You must also have backups available for all files in need of media recovery or you must have all archived redo log files that were generated since the data file was added to the database. Refer to the *Oracle Database Backup and Recovery User's Guide* for additional information about re-creating data files when backups are not available.

You must have all the archive logs available, from the point in time the backups were taken, to the present. If you do not have all of them, you can recover only to the last point in time when redo is available. If no archive logs are required, then only online redo logs are applied.

Query the following views:

- **`V$RECOVER_FILE`:** To see which files need media recovery
- **`V$RECOVERY_LOG`:** To see which archive logs are required to perform recovery

Performing Complete Closed Database Recovery: Overview



Performing Complete Closed Database Recovery: Overview

Under certain circumstances, such as damage to a file belonging to the `SYSTEM` tablespace, the instance shuts down automatically. Even if the instance keeps running, and there are problems with other data files, you may decide there is no value is keeping the database running; too many database objects are affected. In that case, shut down the database to perform the recovery.

If the database is still open, you can query the `V$RECOVER_FILE` view to see which data files are in need of recovery, and after you restored them, query `V$RECOVERY_LOG` to see which archive logs are required. That will tell you which files need to be restored from backup, if any.

Then shut down the database. Look into the media failure to determine the cause of the problem. Repair the problem so that you can restore the files from backup. For example, you may need to replace a disk drive.

Now you can perform the recovery using the `RECOVER` command. Limit the scope of the recovery to only what is needed, such as data file or tablespace. If needed, recover the entire database. Then open the database.

Identifying Recovery-Related Files

- Identify data files that need to be recovered:

```
SQL> SELECT file#, error FROM v$recover_file;
```

- Identify archive log files that are required to complete recovery:

```
SQL> SELECT archive_name FROM v$recovery_log;
```


Identifying Recovery-Related Files

If the database is still open, query the files as described below. Otherwise, attempt to start the instance and mount the database to issue the queries.

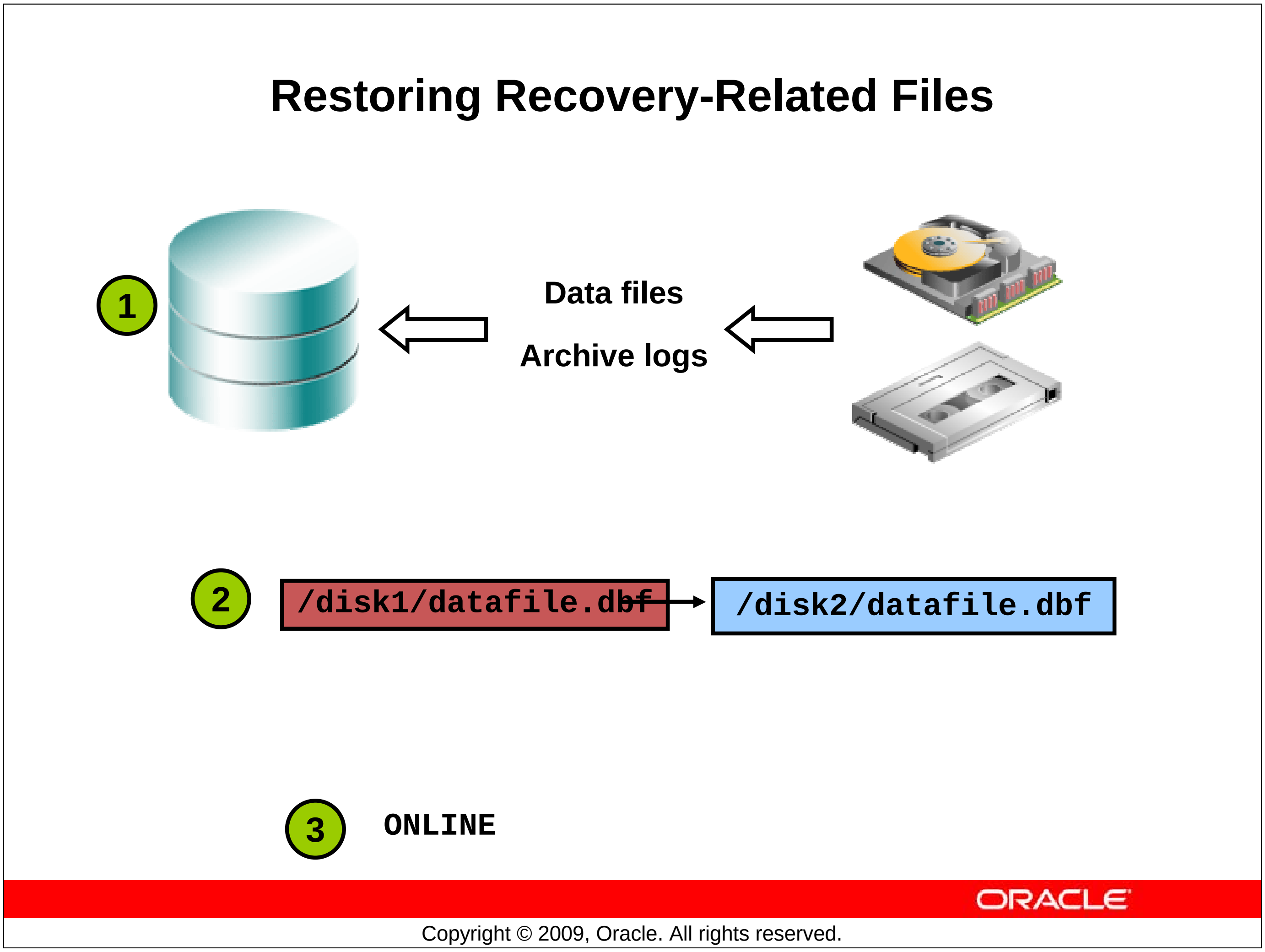
In order to determine which data files require recovery, query the `V$RECOVER_FILE` view. The `ERROR` column indicates why the file requires recovery. If this column has any value other than `OFFLINE NORMAL`, then it needs recovery. To see the whole picture of which data files and tablespaces are affected, join `V$DATAFILE` and `V$TABLESPACE` in this query. Here is an example:

```
SELECT r.FILE#, d.NAME df_name, t.NAME tbsp_name,
       d.STATUS, r.ERROR, r.CHANGE#, r.TIME
FROM   V$RECOVER_FILE r, V$DATAFILE d, V$TABLESPACE t
WHERE  t.TS# = d.TS#
AND    d.FILE# = r.FILE#;
```

This tells you the extent of the damage, helping you decide what the objects of the `RECOVER` command should be.

The `V$RECOVERY_LOG` view shows which archive log files are needed to perform the recovery. If the list shows files that have since been moved off the default archive log location, then you have to restore them to some location before performing recovery.

After recording the results of these queries, shut down the database.



Restoring Recovery-Related Files

After determining what data files and archive log files are required, restore them to appropriate disk locations. Restore a data file by copying it from the backup location, as in this example:

```
$ cp /disk2/backup/datafile/survey01.dbf \
> $ORACLE_BASE/oradata/ORCL/datafile/survey01.dbf
```

If any archive logs are needed for recovery, check whether they are still in the default disk location for archive logs. They may not be, if they have been moved to tape or another disk drive, for example. If they have been moved, they need to be restored, either to the default archive log location or to a temporary location. If there is enough space available in the default location (which is specified by the `LOG_ARCHIVE_DEST_1` initialization parameter), then restore them there. Otherwise, you can put them on some other disk location. When it is time to restore, you can specify that alternate location to find archive log files.

If you had to move a data file, that fact has to be recorded in the control file. That is done by executing the `ALTER DATABASE RENAME FILE` command, as shown in the following example:

```
SQL> ALTER DATABASE RENAME FILE
  2> '/u01/app/oracle/oradata/ORCL/datafile/survey01.dbf' TO
  3> '/newdisk/ORCL/datafile/survey01.dbf';
```

Note: You must start the instance and mount the database before executing the `ALTER DATABASE RENAME FILE` command.

Restoring Recovery-Related Files (continued)

If you have not yet done so, mount the database and bring all the data files online. You can check the status of each data file by querying the `V$DATAFILE` view. Bring the data files online by using a command like the following:

```
SQL> ALTER DATABASE DATAFILE \
2  > '/newdisk/ORCL/datafile/survey01.dbf' ONLINE;
```

Applying Redo Data

1. Apply redo data using the `RECOVER` command:

```
SQL> RECOVER AUTOMATIC FROM '/u01/arch_temp' DATABASE;
```

Apply each redo log without prompting.

Alternate location for restored archive log files

Could be `DATABASE`, `TABLESPACE`, or `DATAFILE`

2. Open the database:

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

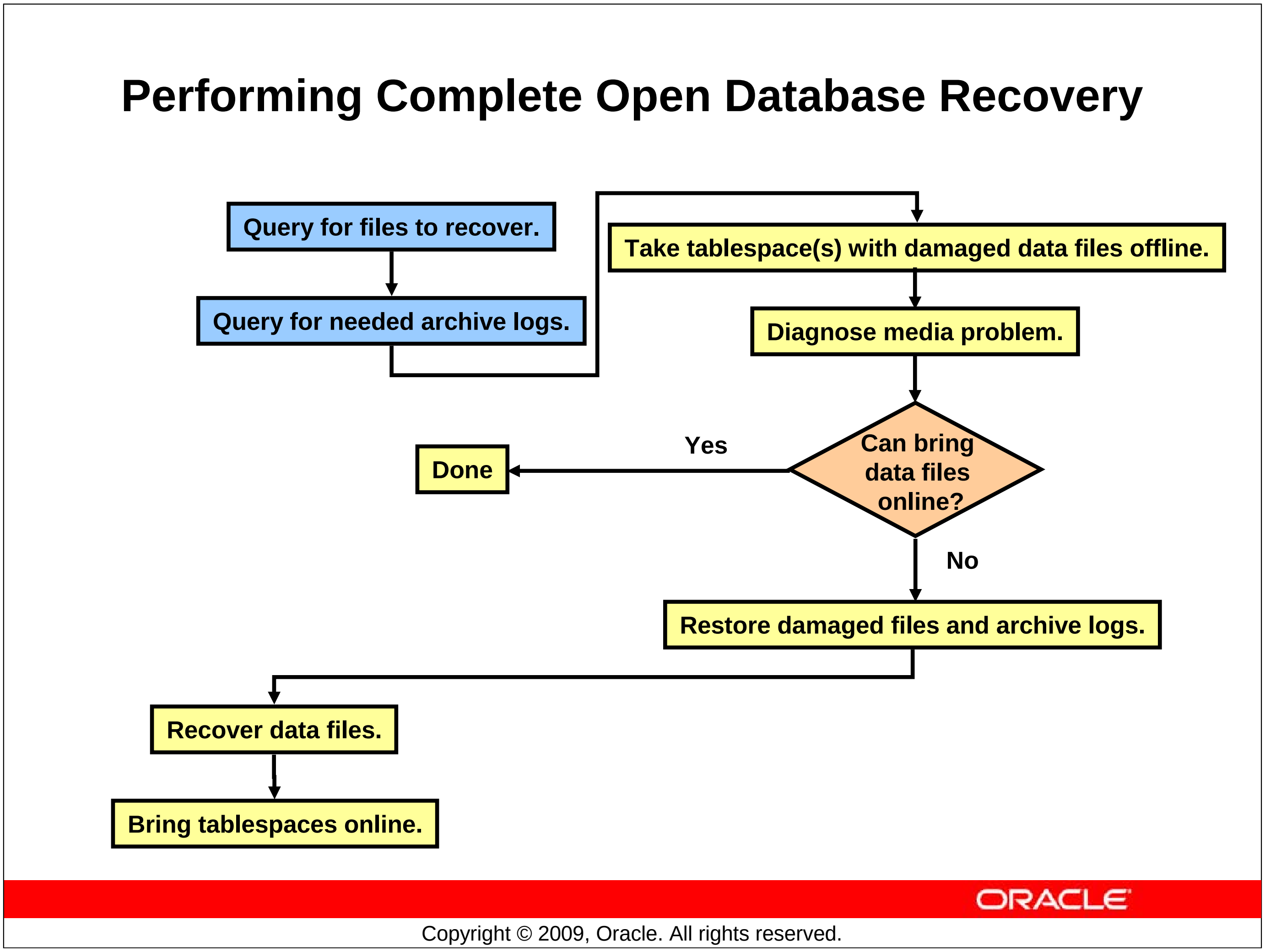

Applying Redo Data

Now the data files have been restored to some point in the past. The archive log files have also been restored: either to their default location or to some other location, for the purpose of this recovery only. You are ready to perform the actual recovery step, which means the redo is applied and the data files are brought up to the latest SCN. Do that using the SQL*Plus `RECOVER` command.

If you do not specify the `AUTOMATIC` option, then you are prompted for each redo log file that is about to be applied. That gives you more control over the recovery process. Typically, `AUTOMATIC` is used for full recovery.

If the archive log files have been restored to some disk location other than the default for the database, then you must specify the `FROM` clause. Supply the directory where the files are stored, and the recovery process will look there for the files.

Finally, open the database. It is now fully recovered.



Performing Complete Open Database Recovery

If media failure occurs while the database is open, the database continues to function. When an attempt is made to write to one of these data files, the data file is taken offline automatically. A query against one of these data files does not cause it to go offline, but it does result in an error being returned to the user that issued the query.

As with the closed database recovery, you first need to query for the files and archive logs that need to be recovered. Then, take all tablespaces that contain damaged data files offline. Use a command such as the following to do this:

```
SQL> ALTER TABLESPACE survey OFFLINE TEMPORARY;
```

Using the `TEMPORARY` option causes Oracle to perform a checkpoint on any online data files belonging to the tablespace. Checkpointed data files do not require recovery when they are brought back online, because they are up-to-date for the latest SCN of any transactions that would have affected them. This is the more desirable option, although the data files must be available at the time this command is run. It is possible that the problem was temporary, and you are able to bring the tablespaces online with no errors.

Performing Complete Open Database Recovery (continued)

Inspect the media to determine the cause of the problem. You can use the `DBVERIFY` utility for this. If the files are permanently damaged, then proceed to restore and recover as described for the closed database recovery earlier in this lesson. After the restore and recovery steps are complete, bring all the tablespaces online again.

Note: For more information about the `DBVERIFY` utility, see the *Backup and Recovery User's Guide*.

Performing User-Managed Incomplete Recovery: Overview

Recover the database to a past point in time in the following situations:

- You want the database to be in the state that existed before a user error or an administrative error occurred.
- The database contains corrupt blocks after you tried block media recovery.
- You are unable to perform complete database recovery because some of the redo log files are missing.
- You want to create a test database that is in the state at some time in the past.
- One or more unarchived redo log files and a data file are lost.

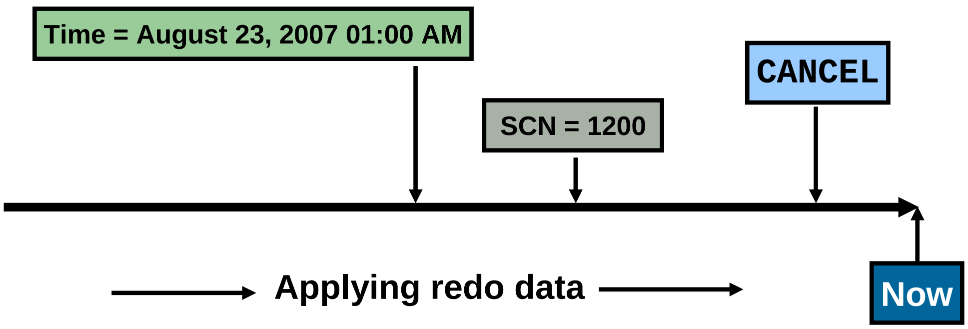
Performing User-Managed Incomplete Recovery: Overview

An incomplete recovery is one that does not bring the database back to the most recent SCN that was transacted. For some reason, as listed in the slide, you need to recover that database only up to a point in the past, not to the present. The processing that occurs when performing incomplete recovery differs from the processing for complete recovery, basically, in the amount of redo that is applied.

Choosing an Incomplete Recovery Method

Indicate when to stop applying redo data by:

- Specifying a time at which to stop
- Specifying an SCN at which to stop
- Issuing a `CANCEL` command while the recovery is executing



Choosing an Incomplete Recovery Method

As you plan your incomplete recovery, decide which method you want to use for specifying when to stop applying redo data. You stop the recovery process by specifying one of the following:

- **A time:** The time in the logs at which recovery should stop. This can be automated so that the process does not prompt you for each file name.
- **An SCN:** The system change number at which recovery should stop. This can be automated so that the process does not prompt you for each file name.
- **`CANCEL`:** Specify the `CANCEL` keyword when the recovery process prompts for the next redo log file name. You cannot automate this process because you must specify `CANCEL` to terminate the recovery operation.

Performing User-Managed Incomplete Recovery

- Recover a database until time:

```
SQL> RECOVER DATABASE UNTIL
  2  TIME '2005-12-14:12:10:03';
```

- Recover a database until cancel:

```
SQL> RECOVER DATABASE UNTIL CANCEL;
```


Performing User-Managed Incomplete Recovery

The following command is used to perform incomplete recovery:

```
RECOVER [AUTOMATIC] DATABASE option
```

Following are the meanings of the options:

- **`AUTOMATIC`:** Automatically applies archived and redo log files
- ***`option`*:** `UNTIL TIME 'YYYY-MM-DD:HH24:MI:SS'`
  `UNTIL CANCEL`
  `UNTIL CHANGE <integer>`
  `USING BACKUP CONTROLFILE`

Cancel-Based Incomplete Recovery

Cancel-based incomplete recovery is very much like closed database complete recovery. The difference is how you execute the `RECOVER` command; specify the `UNTIL CANCEL` clause. This clause causes the recovery process to prompt you with the suggested name for each redo log file to be applied. So, as the recovery proceeds, you are prompted with an archived or online redo log file name, and, for each one, you can either accept it or change it. When you reach the point where you want the recovery to stop, enter `CANCEL` instead of accepting the file name. This stops the recovery.

After this is done, you have to open the database with the `RESETLOGS` option. The database is in another instantiation now, so the redo log sequence numbers need to be reset.

Performing User-Managed Incomplete Recovery (continued)

After opening the database, check the alert log for messages. This is how you find out if the recovery was successful.

Time- and Change-Based Incomplete Recovery

Both time- and change-based incomplete recovery are like the cancel-based recovery, except that different criteria are used to specify when to stop the recovery. Time-based recovery uses a time specified on the command line of the `RECOVER` command, to know when to stop. Change-based recovery uses an SCN, specified on the command line.

As with all incomplete recoveries, the database must then be opened using the `RESETLOGS` option.

Note: To apply redo log files automatically during recovery, you can use the SQL*Plus `SET AUTORECOVERY ON` command, enter `AUTO` at the recovery prompt, or use the `RECOVER AUTOMATIC` command.

Performing User-Managed Incomplete Recovery: Steps

To perform user-managed incomplete recovery, follow these steps:

1. Shut down the database.
2. Restore data files.
3. Mount the database.
4. Recover the database.
5. Open the database with the `RESETLOGS` option.

ORACLE

Performing User-Managed Incomplete Recovery: Steps

1. If the database is open, shut it down by using the `NORMAL`, `IMMEDIATE`, or `TRANSACTIONAL` option.
2. Restore all data files from backup. You must use a backup taken before the time you plan to recover to. You may also need to restore archived logs. If there is enough space available, restore to the `LOG_ARCHIVE_DEST` location or use the `ALTER SYSTEM ARCHIVE LOG START TO <LOCATION>` command or the `SET LOGSOURCE <LOCATION>` command to change the location. If you perform incomplete recovery to a point when the database structure is different than the current, you also need to restore the control file.
3. Mount the database.
4. Recover the database by using the `RECOVER DATABASE` command.
5. To synchronize data files with control files and redo logs, open the database by using the `RESETLOGS` option.

User-Managed Time-Based Recovery: Example

This is the scenario:

- A job ran in error, and its effects must be undone.
- This happened 15 minutes ago, and there has been little database activity since then.
- You decide to perform incomplete recovery to restore the database back to its state as of 15 minutes ago.

```
SQL> SHUTDOWN IMMEDIATE
$ cp /BACKUP/*.dbf /u01/db01/ORADATA
SQL> STARTUP MOUNT
SQL> RECOVER DATABASE UNTIL TIME '2005-11-28:11:44:00';
SQL> ALTER DATABASE OPEN RESETLOGS;
```


User-Managed Time-Based Recovery: Example

The following is a typical scenario employing `UNTIL TIME` recovery. Assume the following facts:

- The current time is 12:00 PM on November 28, 2005.
- A job was run incorrectly, and many tables in several schemas were affected.
- This happened at approximately 11:45 AM.
- Database activity is minimal because most staff are currently in a meeting. The state of the database before the job ran must be restored.

Because the approximate time of the error is known and the database structure has not changed since 11:44 AM, you can use the `UNTIL TIME` method:

1. If the database is open, shut it down by using the `NORMAL`, `IMMEDIATE`, or `TRANSACTIONAL` option.
2. Restore all data files from backup (the most recent if possible). You may also need to restore archived logs. If there is enough space available, restore to the `LOG_ARCHIVE_DEST` location or use the `ALTER SYSTEM ARCHIVE LOG START TO <LOCATION>` command or the `SET LOGSOURCE <LOCATION>` command to change the location.
3. Mount the database.

User-Managed Time-Based Recovery: Example (continued)

4. Recover the database:
```
SQL> recover database until time '2005-11-28:11:44:00'
ORA-00279: change 148448 … 11/27/05 17:04:20 needed for thread
...
Media recovery complete.
```
5. To synchronize data files with control files and redo logs, open the database by using the `RESETLOGS` option:
```
SQL> alter database open resetlogs;
SQL> archive log list
...
Oldest online log sequence 0
Next log sequence to archive 1
Current log sequence 1
```

When recovery is successful, notify users that the database is available for use, and any data entered after the recovery time (11:44 AM) will need to be reentered.

User-Managed Cancel-Based Recovery: Example

The scenario is the same as the one for the time-based example, except for these findings:

- Redo logs are not multiplexed.
- One of the online redo logs is missing.
- The missing redo log is not archived.
- The redo log contained information from 11:34 AM.
- Twenty-six minutes of data are lost.
- Users can reenter their data manually.

User-Managed Cancel-Based Recovery: Example

After searching through the directory for the redo log files, you notice that redo log `log2a.rdo` cannot be located and has not been archived. Therefore, you cannot recover past this point.

Querying `V$ARCHIVED_LOG` confirms the absence of archived log sequence 48 (`log2a.rdo`):

```
SQL> SELECT * FROM v$archived_log;
RECID    STAMP       ...   FIRST_CHANGE#    FIRST_TIME
-----    ---------   ...   -------------    ----------
   1     318531466   ...   88330            05-11-15:12:43
  47     319512880   ...   309067           05-11-28:11:26
```

User-Managed Cancel-Based Recovery: Example

Recover the database as follows:

- Shut down the database.
- Restore all data files from the most recent backup.
- Mount the database.
- Execute `RECOVER DATABASE UNTIL CANCEL`.
- Execute `ALTER DATABASE OPEN RESETLOGS` to open the database.

User-Managed Cancel-Based Recovery: Example (continued)

The steps for cancel-based recovery are the same as for time-based recovery, except for the `RECOVER DATABASE` step. When the `RECOVER DATABASE UNTIL CANCEL` command is executed, it recovers the database until it cannot find a log file. When you are prompted for the name of the missing archived redo log file, enter `CANCEL`; the recovery stops at that point in time.

Summary

In this lesson, you should have learned how to:

- Describe the difference between user-managed and server-managed backup and recovery
- Perform user-managed complete database recovery
- Perform user-managed incomplete database recovery

Managing the ASM Instance

Objectives

After completing this lesson, you should be able to:

- Describe the benefits of using ASM
- Manage the ASM instance
- Create and Drop ASM disk groups
- Extend ASM disk groups
- Retrieve ASM metadata using various utilities

Objectives

This lesson provides a more detailed look at the ASM instance and how to manage it with various utilities.

ASM Benefits for Administrators

ASM eliminates:
- I/O performance tuning
- Data file movements and reorganizations
- File name management
- Logical volume management
- File system management
- Cluster file system management
- Raw device management

ASM significantly reduces:
- LUN management
  - Fewer, larger LUNs
- Database administrator dependence on system administrator
- Likelihood of errors associated with manual maintenance tasks

ASM Benefits for Administrators

ASM eliminates the need for many tasks that are required in non-ASM storage environments. There include:

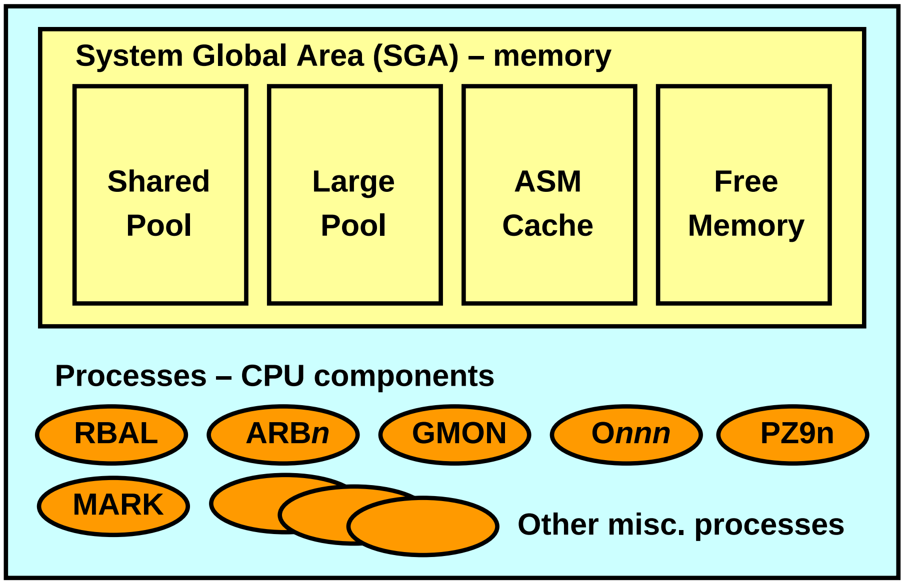
- I/O performance tuning - ASM's stripe and mirror everything policy coupled with automatic rebalancing operations means that I/O performance tuning aimed at balancing disk utilization and eliminating disk hot-spots is not required.
- Data file movements and reorganizations - Juggling the placement of data files to satisfy performance requirements and space constraints is not required.
- File name management - You no longer need to define and enforce a file naming policy.
- Logical volume, file system, cluster file system and raw device management - You no longer require these storage elements.

ASM delivers further benefits by reducing effort in these important areas:

- LUN management effort is reduced as ASM will typically require fewer, larger, LUNs.
- The dependence that often exists between a database administrator and a system administrator is greatly reduced. For example, system administrator involvement is not required to add a new data file or move disk resources from one disk group to another.
- The likelihood of errors associated with manual maintenance tasks is greatly reduced. For example, using a conventional file system a newly created data file may accidentally break a file naming convention that results in it not being backed up with the rest of the database.

ASM Instance

The ASM Instance is the process and memory components for ASM.



ASM Instance

Every time ASM or a database is started, a shared memory area called the system global area (SGA) is allocated and Oracle ASM or database background processes are started. The combination of the background processes and the SGA is called an Oracle ASM instance or an Oracle Database instance. The instance represents the CPU and RAM components of a running ASM environment.

The SGA in an ASM instance is different in memory allocation and usage than the SGA in a database instance. The SGA in the ASM instance is divided into four primary areas as follows:

- **Shared Pool:** Used for metadata information
- **Large Pool:** Used for parallel operations
- **ASM Cache:** Used for reading and writing blocks during rebalance operations
- **Free Memory:** Unallocated memory available

The minimum recommended amount of memory for an ASM instance is 256 MB. Automatic memory management is enabled by default on an ASM instance and will dynamically tune the sizes of the individual SGA memory components. The amount of memory that is needed for an ASM instance will depend on the amount of disk space being managed by ASM.

The second part of the ASM instance is the background processes. An ASM instance can have many background processes; not all are always present.

ASM Components: ASM Instance (continued)

The background processes specific to ASM functionality are covered in the next slide. There are required background processes and optional background processes. Some of these processes may include the following:

- **ARC*n*:** The archiver processes
- **CKPT:** The checkpoint process
- **DBW*n*:** The database writer processes
- **DIAG:** The diagnosability process
- **J*nnn*:** Job queue processes
- **LGWR:** The log writer process
- **PMON:** The process monitor process
- **PSP0:** The process spawner process
- **QMN*n*:** The queue monitor processes
- **RECO:** The recoverer process
- **SMON:** The system monitor process
- **VKTM:** The virtual keeper of time process
- **MMAN:** The memory manager process

The above list of processes in not complete. For the ASM instance, these processes will not always perform the same tasks as they would in a database instance. For example, the `LGWR` process in a database instance is responsible for copying change vectors from the log buffer section of the SGA to the online redo logs on disk. The ASM instance does not contain a log buffer in it's SGA, nor does it use online redo logs. The `LGWR` process in an ASM instance copies logging information to an ASM disk group.

If ASM is clustered, then additional processes related to cluster management will be running in the ASM instance. Some of these processes include the following:

- **LMON:** The global enqueue service monitor process
- **LMD*n*:** The global enqueue service daemons
- **LMS*n*:** The global cache service processes
- **LCK*n*:** The lock processes

ASM Components: ASM Instance—Primary Processes

The ASM instance primary processes are responsible for ASM-related activities.

Process	Description
RBAL	Opens all device files as part of discovery and coordinates the rebalance activity
ARB*n*	One or more slave processes that do the rebalance activity
GMON	Responsible for managing the disk-level activities such as drop or offline and advancing the ASM disk group compatibility
MARK	Marks ASM allocation units as stale when needed
Onnn	One or more ASM slave processes forming a pool of connections to the ASM instance for exchanging messages
PZ9n	One or more parallel slave processes used in fetching data on clustered ASM installation from GV$ views

ASM Components: ASM Instance—Primary Processes

The ASM instance uses dedicated background processes for much of its functionality. The RBAL process coordinates rebalance activity for disk groups in an Automatic Storage Management instance. It performs a global open on Automatic Storage Management disks. The ARB*n* processes perform the actual rebalance data extent movements in an Automatic Storage Management instance. There can be many of these at a time, called ARB0, ARB1, and so on. The GMON process maintains disk membership in ASM disk groups. The MARK process marks ASM allocation units as stale following a missed write to an offline disk. The Onnn processes represent the server side of a client/server connection. These processes will appear the moment the instance is started, and will disappear after that. They form a pool of connections to the ASM instance for exchanging messages and only appear when needed. The PZ9n processes represent one or more parallel slave processes that are used in fetching data when ASM is running in a clustered configuration on more than one machine concurrently.

ASM Instance Initialization Parameters

The ASM instance uses a small subset of the parameters that an Oracle Database instance uses.

```
INSTANCE_TYPE = ASM
ASM_POWER_LIMIT = 1
ASM_DISKSTRING = '/dev/sda1','/dev/sdb*'
ASM_DISKGROUPS = DATA2, FRA
ASM_PREFERRED_READ_FAILURE_GROUPS = DATA.FailGroup2
DIAGNOSTIC_DEST = /u01/app/oracle
LARGE_POOL_SIZE = 12M
REMOTE_LOGIN_PASSWORDFILE = EXLUSIVE
```

ORACLE

ASM Instance Initialization Parameters

An ASM instance is controlled by a parameter file in the same way as a regular database instance. Parameters commonly set there include:

- `INSTANCE_TYPE` should be set to ASM for ASM instances. This is the only parameter that must be defined. For database instances, this is set to the value `RDBMS`.
- `ASM_POWER_LIMIT` controls the speed for a rebalance operation. Values range from 1 through 11, with 11 being the fastest. If omitted, this value defaults to 1.
- `ASM_DISKSTRING` is an operating system–dependent value used by ASM to limit the set of disks considered for discovery. The default value is the null string, and this will be sufficient in most cases. A more restrictive value as shown above may reduce the time required for ASM to perform discovery, and thus improve disk group mount times.
- `ASM_DISKGROUPS` is the list of names of disk groups to be mounted by an ASM instance at startup, or when the `ALTER DISKGROUP ALL MOUNT` command is used. Oracle Restart can mount disk groups if they are listed as dependencies even if they are not listed with the `ASM_DISKGROUPS` parameter. This parameter has no default value.
- `ASM_PREFERRED_READ_FAILURE_GROUPS` specifies the failure groups that contain preferred read disk. This is useful in extended or stretched cluster databases that have mirrored copies of data with one of the copies in close proximity to the server.

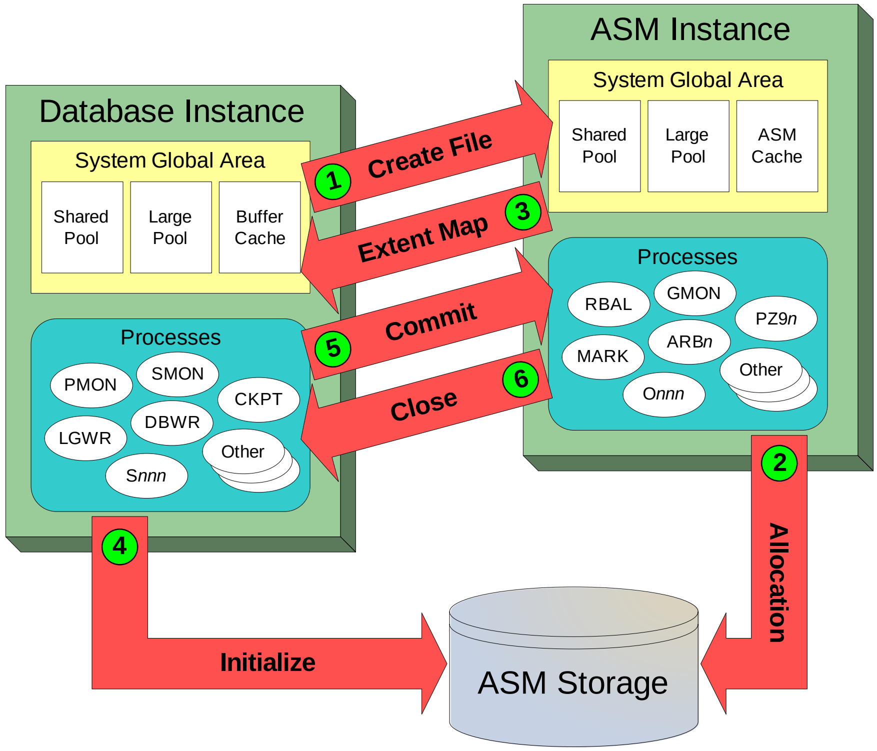
ASM Instance Initialization Parameters (continued)

- DIAGNOSTIC_DEST specifies the location of the Automatic Diagnostic Repository (ADR) home. Trace files, alert logs, core files, and incident files can be found under this directory. The default value of this parameter is derived from the value of ORACLE_BASE.
- LARGE_POOL_SIZE specifies (in bytes) the size of the large pool allocation heap. The large pool allocation heap is used in shared server systems for session memory, by parallel execution for message buffers, and by backup processes for disk I/O buffers. The ASM instance makes use of automatic memory management, so this parameter serves as a minimum size that the large pool can lowered to.
- REMOTE_LOGIN_PASSWORDFILE specifies whether the Oracle software checks for a password file. The default value is EXCLUSIVE.

The eight parameters listed above are the only non-default parameters created for an ASM instance. The ASM instance differs from a database instance in that not all database parameters are valid for an ASM instance. Approximately 74 of the 344 total database instance parameters can be used with an ASM instance. The remaining parameters not listed on this can be set as needed, although default values should be sufficient for most installations.

Note: Automatic memory management is enabled by default on ASM instances, even when the MEMORY_TARGET parameter is not explicitly set. This is the only parameter that you need to set for complete ASM memory management. Oracle Corporation strongly recommends that you use automatic memory management for ASM.

Interaction Between Database Instances and ASM



Interaction Between Database Instances and ASM

The file creation process provides a fine illustration of the interactions that take place between database instances and ASM. The file creation process occurs as follows:

1. The database requests file creation.
2. An ASM foreground process creates a Continuing Operation Directory (COD) entry and allocates space for the new file across the disk group.
3. The ASMB database process receives an extent map for the new file.
4. The file is now open and the database process initializes the file directly.
5. After initialization, the database process requests that the file creation is committed. This causes the ASM foreground process to clear the COD entry and mark the file as created.
6. Acknowledgement of the file commit implicitly closes the file. The database instance will need to reopen the file for future I/O.

This example reinforces two important points about the architecture of ASM:

- The Database Instance and ASM Instance work together in a coordinated fashion. A Database instance must interact with ASM to map database files to ASM extents. A Database instance also receives a constant stream of messages relating to ASM operations (such as disk group rebalancing) that may lock or move ASM extents.
- Database I/O is not channeled through the ASM instance. In fact, the database conducts I/O operations directly against ASM files, as illustrated in step 4 in the slide.

ASM Instance: Dynamic Performance Views

The ASM instance hosts memory-based metadata tables presented as dynamic performance views.

- Accessed by ASM utilities to retrieve metadata-only information using the SQL language
- Contains many dedicated ASM related views such as:

V$ASM_ALIAS	V$ASM_ATTRIBUTE	V$ASM_CLIENT
V$ASM_DISK	V$ASM_DISK_IOSTAT	V$ASM_DISK_STAT
V$ASM_DISKGROUP	V$ASM_DISKGROUP_STAT	V$ASM_FILE
V$ASM_OPERATION	V$ASM_TEMPLATE	

ASM Instance: Dynamic Performance Views

One of the major functions of any instance is the storage of memory-based metadata tables. These tables start with the prefix X$ and are generally not documented. A series of dynamic performance views that start with the prefix V$ are used to display a customized presentation of the data contained in the X$ memory tables. The information is presented in a read-only fashion available only to administrators with privileges. The information is retrieved from ASM using the SQL language. The above slide lists the most common dynamic performance views that contain ASM-related metadata. There are several hundred additional dynamic performance views, but most of these views will be empty because they require a database instance to mount a database control file. ASM instances do not mount database control files. For a complete list of the dynamic performance views, see the *Oracle Database Reference 11g Release 2 (11.2)* documentation manual.

ASM System Privileges

- An ASM instance does not have a data dictionary, so the only way to connect to ASM is by using these system privileges.

ASM Privilege	Privilege Group (Suggested)	Privilege
SYSASM	OSASM (asmadmin)	Full administrative privilege
SYSDBA	OSDBA (asmdba)	Access to data stored on ASM, and SYSASM in the current release
SYSOPER	OSOPER (asmoper)	Limited privileges to start and stop the ASM instance along with a set of nondestructive ALTER DISKGROUP commands

- SYS user is automatically created with the SYSASM privilege.

ASM System Privileges

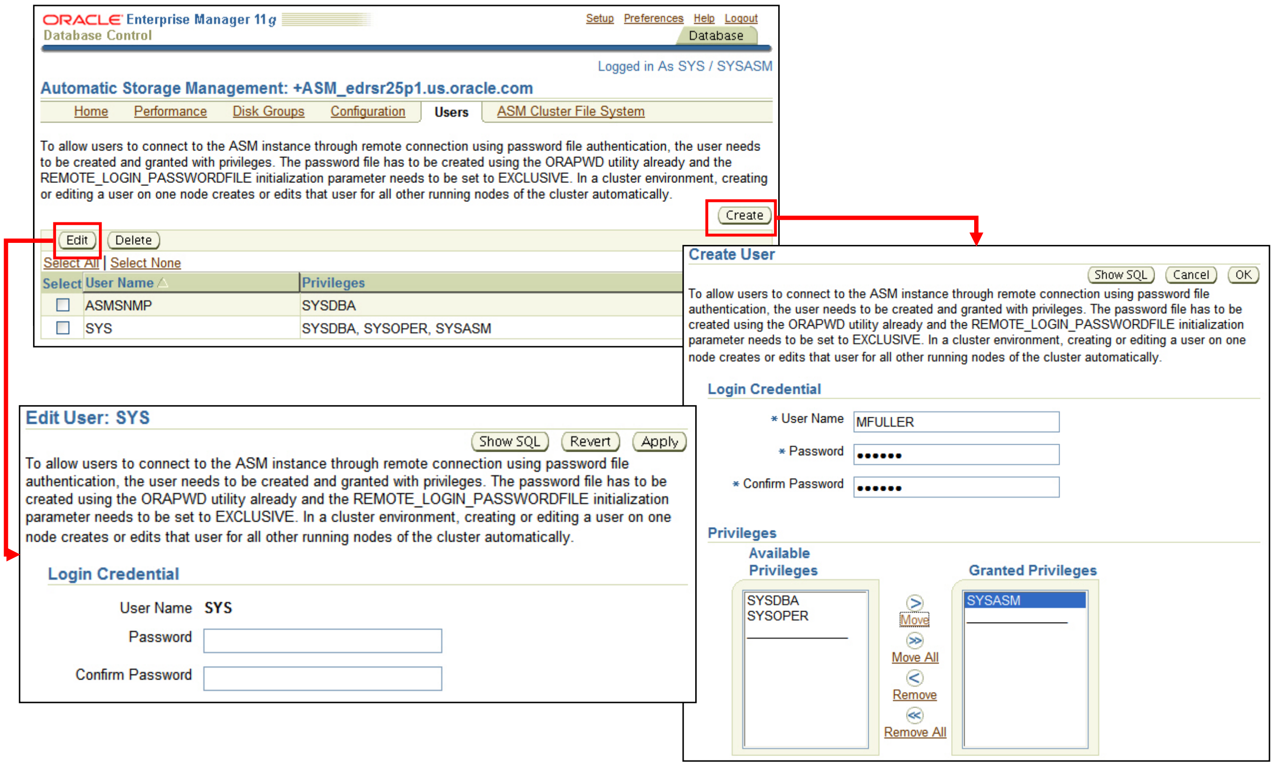
An ASM instance does not have a data dictionary, so the only way to connect to an ASM instance is by using one of three system privileges, SYSASM, SYSDBA, or SYSOPER. The following list introduces these ASM system privileges.

- SYSASM: This privilege provides full administrative privilege for the ASM instance.
- SYSDBA: This privilege grants access to data stored on ASM, and in the current release, grants the SYSASM administrative privileges.
- SYSOPER: This privilege grants the ability to start and stop ASM instances along with a set of nondestructive ALTER DISKGROUP commands. Other commands such as CREATE DISKGROUP are not allowed.

When ASM is installed, operating system groups are assigned the SYSASM, SYSDBA, and SYSOPER privileges. These groups are known as the OSASM, OSDBA, and OSOPER groups, respectively. Membership of these operating system groups automatically assigns the related system privileges to the operating system user in question

When an ASM instance is first created, sys and asmsnmp are the only ASM users that are defined.

Using Enterprise Manager to Manage ASM Users



Using Enterprise Manager to Manage ASM Users

Enterprise Manager allows you to manage the users who access the ASM instance through remote connection (using password file authentication). These users are reserved exclusively for the ASM instance.

You have this functionality only when you are connected as the SYSASM user. It is hidden if you connect as SYSDBA or SYSOPER users.

- When you click the Create button, the Create User page is displayed.
- When you click the Edit button the Edit User page is displayed.
- By clicking the Delete button, you can delete the created users.

Note: To login to ASM with the SYSASM role, click the Preferences link at the top of the page, followed by the Preferred Credentials link. You will then see ASM in the list of target types. Click the Set Credentials icon beside the ASM target type to define an account and password that has the SYSASM privilege. You may need to log out of Database Control and login before the change takes effect.

Starting and Stopping ASM Instances Using SQL*Plus

Starting and stopping ASM instances using SQL*Plus is similar to the way in which you start and stop database instances.

```
$ . oraenv
ORACLE_SID = [orcl] ? +ASM
The Oracle base for ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is
/u01/app/oracle
$ sqlplus / AS SYSASM
SQL*Plus: Release 11.2.0.1.0 - Production on Wed Jul 8 20:46:46 2009
Copyright (c) 1982, 2009, Oracle.  All rights reserved.
Connected to an idle instance.
SQL> startup
ASM instance started

Total System Global Area  284565504 bytes
Fixed Size                  1336028 bytes
Variable Size             258063652 bytes
ASM Cache                  25165824 bytes
ASM diskgroups mounted
ASM diskgroups volume enabled
SQL> shutdown abort
```

ORACLE

Starting and Stopping ASM Instances Using SQL*Plus

With SQL*Plus you start an ASM instance by using the `STARTUP` command similarly to the way in which you start an Oracle Database instance. When starting an ASM instance, note the following:

- To connect to an ASM instance with SQL*Plus, set the `ORACLE_SID` environment variable to the ASM SID. The default ASM SID for a single instance database is `+ASM`, and the default SID for ASM for an Oracle RAC node is `+ASM`*`node_number`* where *`node_number`* is the number of the node. The `oraenv` script will set the `ORACLE_BASE`, `ORACLE_SID`, `ORACLE_HOME` and `PATH` variables.
- The initialization parameter file must contain the following entry:

  `INSTANCE_TYPE = ASM`

  This parameter indicates that an ASM instance, not a database instance, is starting.
- When you run the `STARTUP` command, rather than trying to mount and open a database, this command attempts to mount the disk groups specified by the initialization parameter `ASM_DISKGROUPS`. If you have not entered a value for `ASM_DISKGROUPS`, you can later mount disk groups with the `ALTER DISKGROUP...MOUNT` command.

Starting and Stopping ASM Instances Using SQL*Plus (continued)

The following list describes the `STARTUP` command parameters relevant to ASM.

- `FORCE`: Issues a `SHUTDOWN ABORT` to the ASM instance before restarting it
- `MOUNT` or `OPEN`: Mounts the disk groups specified in the `ASM_DISKGROUPS` initialization parameter. This is the default if no command parameter is specified.
- `NOMOUNT`: Starts up the ASM instance without mounting any disk groups
- `RESTRICT`: Starts up an instance in restricted mode. The `RESTRICT` clause can be used in combination with the `MOUNT`, `NOMOUNT`, and `OPEN` clauses.

  In restricted mode, database instances cannot use the disk groups. That is, databases cannot open files that are in that disk group. Also, if a disk group is mounted by an instance in restricted mode, then that disk group cannot be mounted by any other instance in the cluster. Restricted mode enables you to perform maintenance tasks on a disk group without interference from clients. Rebalance operations that occur while a disk group is in restricted mode eliminate the lock and unlock extent map messaging that occurs between ASM instances in a clustered environment. This improves the overall rebalance throughput. At the end of a maintenance period, you must explicitly dismount the disk group and remount it in normal mode.

The ASM shutdown process is initiated when you run the `SHUTDOWN` command in SQL*Plus. Before you run this command, ensure that the `ORACLE_SID` and `ORACLE_HOME` environment variables are set so that you can connect to the ASM instance.

Oracle strongly recommends that you shut down all database instances that use the ASM instance before attempting to shut down the ASM instance.

The following list describes the `SHUTDOWN` command parameters relevant to ASM.

- `NORMAL`: ASM waits for any in-progress SQL to complete before dismounting all of the disk groups and shutting down the ASM instance. Before the instance is shut down, ASM waits for all of the currently connected users to disconnect from the instance. If any database instances are connected to the ASM instance, then the `SHUTDOWN` command returns an error and leaves the ASM instance running. `NORMAL` is the default shutdown mode.
- `IMMEDIATE` or `TRANSACTIONAL`: ASM waits for any in-progress SQL to complete before dismounting all of the disk groups and shutting down the ASM instance. ASM does not wait for users currently connected to the instance to disconnect. If any database instances are connected to the ASM instance, then the `SHUTDOWN` command returns an error and leaves the ASM instance running.
- `ABORT`: The ASM instance immediately shuts down without the orderly dismount of disk groups. This causes recovery to occur upon the next ASM startup. If any database instance is connected to the ASM instance, then the database instance aborts.

Note: The `NORMAL`, `IMMEDIATE`, and `TRANSACTIONAL` forms of shutdown do not apply when there are connected RDBMS instances. The following error will be returned:

```
ORA-15097: cannot SHUTDOWN ASM instance with connect RDBMS
instance
```

Starting and Stopping ASM Instances Using `srvctl`

- The Sever Control utility (`srvctl`) can be used to start and stop ASM instances.

```
$ . oraenv
ORACLE_SID = [orcl] ? +ASM
The Oracle base for
    ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is
    /u01/app/oracle
$ srvctl start asm -o mount
$ srvctl stop asm -f
```

- The Sever Control utility (`srvctl`) can be used to check the status of ASM instances.

```
$ srvctl status asm
ASM is running on edrsr25p1
```


Starting and Stopping ASM Instances Using `srvctl`

The Server Control utility (`srvctl`) can be used to start and stop ASM instances along with other resources managed by the Grid Infrastructure. The `srvctl` utility can be found under both the `ORACLE_HOME/bin` location for Grid infrastructure and the `ORACLE_HOME/bin` location of the database installation. You should use the srvctl utility located with the Grid Infrastructure installation when managing ASM, listeners, or Oracle restart. The `srvctl` utility can be used to control ASM in the following ways:

- Start an ASM instance.
  `srvctl start asm [-o <start_option>]`
  `<start_option>` is one the valid instance startup options (`FORCE`, `MOUNT`, `OPEN`, `NOMOUNT` or `RESTRICT`) (optional)
- Stop an ASM instance.
  `srvctl stop asm [-o <stop_option>] -f`
  `<stop_option>` is one the valid instance shutdown options (`NORMAL`, `IMMEDIATE`, `TRANSACTIONAL` or `ABORT`) (optional), and –f is to force
- Report the status of an ASM instance.
  `srvctl status asm`

Starting and Stopping ASM Instances Using asmcmd

The `asmcmd` utility provides a command line interface to ASM without using the SQL language.

```
$ . oraenv
ORACLE_SID = [orcl] ? +ASM
The Oracle base for ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is
/u01/app/oracle
$ asmcmd
Connected to an idle instance.
ASMCMD> startup
ASM instance started

Total System Global Area  284565504 bytes
Fixed Size                  1336028 bytes
Variable Size             258063652 bytes
ASM Cache                  25165824 bytes
ASM diskgroups mounted
ASM diskgroups volume enabled
ASMCMD> shutdown --abort
ASM instance shut down
Connected to an idle instance.
```


Starting and Stopping ASM Instances Using asmcmd

ASM metadata for administration is found inside the dynamic performance views that are contained within the ASM instance. These views are usually accessed with an ASM utility using the SQL language. The requirement for knowledge of the SQL language increases the learning curve for mastering ASM, and SQL language training is not a normal requirement in the learning path of a systems administrator or storage administrator. The `asmcmd` utility provides a pseudo-shell-like environment that accepts UNIX-style syntax for common ASM administration tasks. It can be used to manage Oracle ASM instances, disk groups, file access control for disk groups, files and directories within disk groups, templates for disk groups, and volumes.

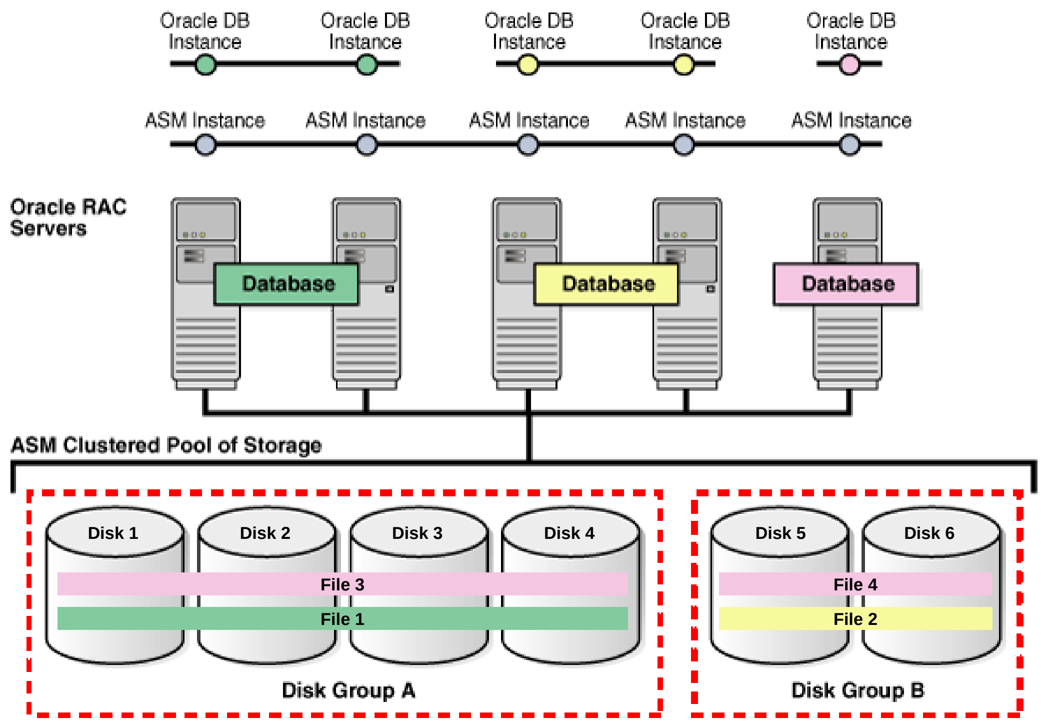
You can use the `asmcmd` utility to startup and shutdown your ASM instance. The supported startup options include:

- `--nomount` (To start the ASM instance without mounting disk groups)
- `--restrict` (To start the ASM instance and restrict database usage)
- `--pfile <pfile.ora>` (To start the ASM instance with a custom pfile)

The supported shutdown options include:

- `--immediate` (Performs shutdown immediate)
- `--abort` (Abort all existing operations)

Disk Group Overview



Disk Group Overview

A disk group is a logical grouping of one or more disks that ASM manages as a collection. Each disk group contains the metadata associated with itself. You can think of an ASM disk group as conceptually similar to a logical volume in a typical storage area network.

Files are allocated from the space inside a disk group. The content of files that are stored in a disk group are evenly distributed, or striped, across the disks in the disk group to eliminate hot spots and to provide uniform performance across the disks. Each ASM file is completely contained within a single disk group. However, a disk group can contain files belonging to several databases and a single database can use different files from multiple disk groups.

A key attribute of a disk group is it's redundancy setting. There are three possible disk group redundancy settings:

- External redundancy, where ASM does not provide any mirroring and the disks are assumed to highly reliable.
- Normal redundancy, where ASM supports 2-way mirroring by default to assure data integrity for less reliable storage.
- High redundancy, where ASM supports 3-way mirroring by default for even greater assurance of data integrity.

ASM supports the creation of up to 63 disk groups however for most installations you rarely need more than a few.

ASM Disks

ASM disks:

- Are the storage devices provisioned to ASM disk groups
- Are accessed through normal O/S interfaces
- Must be read and write accessible by the ASM owner
- Must be accessible by all nodes in a cluster
- May have different O/S names or paths on different nodes
- May be:
  - An entire physical disk or partition of a physical disk
  - A disk or partition from a storage array
  - Logical volumes (LV) or logical units (LUN)
  - Network-attached files (NFS)

ORACLE

ASM Disks

ASM disk groups contain one or more ASM disks.

ASM disks must be read and write accessible by the ASM owner from all nodes in the cluster.

An ASM disk can be a partition on a disk, but Oracle strongly advises against sharing the physical disks used by ASM with other applications. This is because the I/O performance of the ASM disks within a disk group should be similar and consistent for the ASM stripe and mirror policy to work best. Sharing devices between ASM and other applications makes it difficult to assure similar and consistent disk performance.

It is not necessary for the O/S device names of ASM disks on separate nodes in a cluster to be the same. ASM identifies members of a disk group by reading the headers on ASM disks.

Generally, ASM disks are raw LUNs from a storage array presented to ASM. In addition, ASM disks can also be files on a remote NFS filer.

Allocation Units

ASM disks are divided into allocation units (AU):

- AU size is configurable at disk group creation
- Default AU size is 1MB:
  – Small enough to be cached by database and large enough for efficient sequential access
- Allowable AU sizes:
  – 1, 2, 4, 8, 16, 32, or 64 MB.
  – Large AUs may be useful in very large database (VLDB) scenarios or when using specialized storage hardware

ORACLE

Allocation Units

Within an ASM disk, space is divided into allocation units (AU). The default AU size is one megabyte, which is small enough to not become a hot spot, but large enough to provide efficient sequential access. You can set the AU size when you create a disk group. You cannot change the AU size for a disk group. Larger AU sizes may be useful in very large database (VLDB) scenarios or when using specialized storage hardware. If an AU is constantly accessed, then the database kernel caches it for more efficient access.

ASM Files

ASM files:

- Are a collection of ASM extents composed of AUs
  – Variable sized extents support large files
- Appear as normal files to the database kernel
- Have file names that start with '+'
  – For example, `+DATA/orcl/datafile/system.256.689832921`
- May be associated with an optional alias file name
  – For example, `+DATA/dbfiles/mydb/system01.dbf`
- Are evenly distributed across disks in a disk group
- Are mirrored according to the policies defined in the disk group

ORACLE

ASM Files

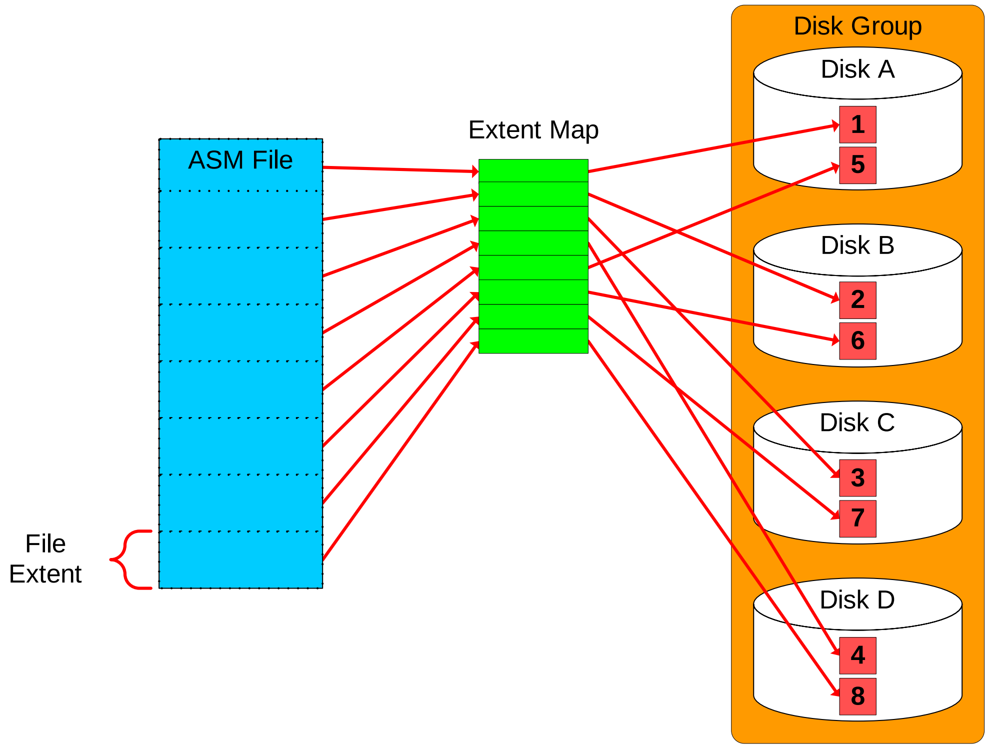
ASM exposes a set of files for use by clients of ASM. An ASM file is composed of a set of allocation units and appears as a normal file to the database kernel.

Each ASM File has a unique system-generated name. The slide shows and example of a fully-qualified ASM file name. The fully-qualified ASM file name represents a hierarchy beginning with a concatenation of a plus sign with the disk group name. Following the disk group name is the database name and file type. The final element is composed of a tag name, file number and incarnation number. An alias can optionally be created to give administrators a more user-friendly means of referring to an ASM file.

Files are evenly spread across the ASM disks in a disk group using the policy of stripe and mirror everything (SAME).

ASM natively supports most database related file types, such as data files, log files, control files, RMAN backups, and others. Prior to Oracle Database 11g Release 2, ASM only supported Oracle database related files and could not be used to store and manage ASCII trace files and alert logs, Oracle binaries, the Oracle cluster registry (OCR), and the cluster voting disk. Oracle Database 11g Release 2 removes this restriction by providing the means to run a general purpose file system over ASM.

Extent Maps



Extent Maps

ASM keeps track of a file’s layout with metadata called an extent map. An extent map is a table that maps data extents in a file to allocation units on disk.

The relationship between file extents and allocation units is as follows. An extent contains:

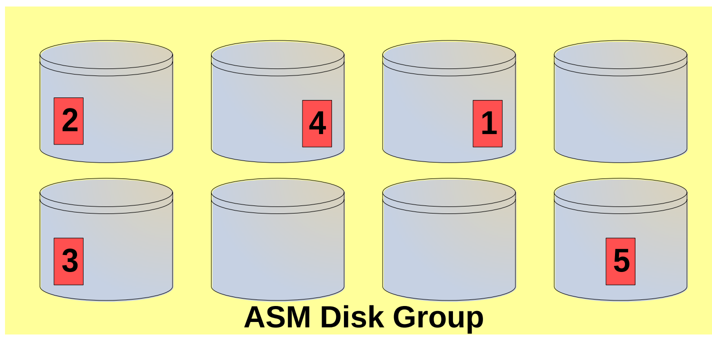
- One AU for the first 20,000 extents (0–19999)
- 4 AUs for the next 20,000 extents (20000–39999)
- 16 AUs for extents above 40,000

Variable size extents, coupled with large allocation units, can be used to accommodate very large ASM files.

Striping Granularity

ASM separates striping for load balance and striping for latency:

- Coarse grain striping concatenates allocation units for load balancing.
  - For example:



Striping Granularity

In ASM striping has two primary purposes:

- Load balance I/O across all the disks in a disk group.
- Improve I/O latency.

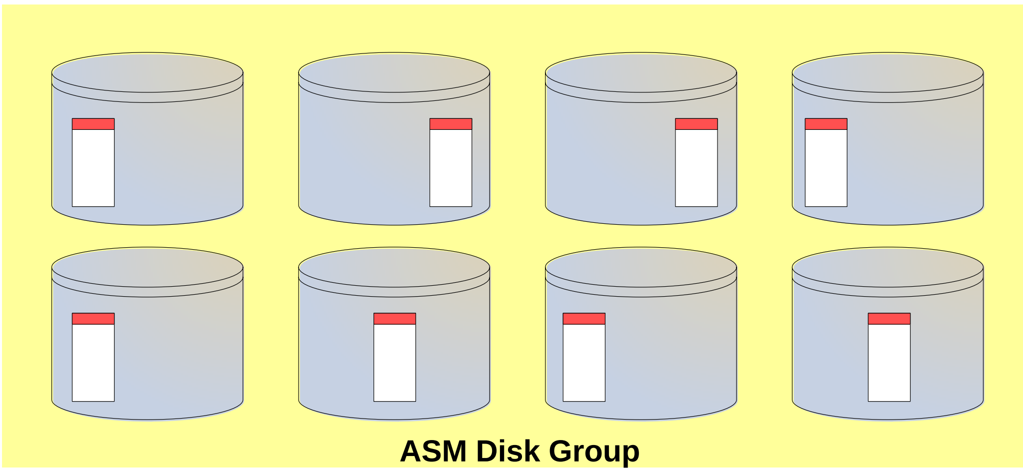
Coarse grain striping spreads allocation units across the disks in a disk group. This is what provides load balancing for disk groups. When a file is allocated, ASM spreads allocation units evenly across all of the disks. Sometimes the distribution cannot be perfectly even, but over time it will tend to be nearly equal. The above diagram shows a file with five allocation units striped across five disks in an external redundancy disk group containing 8 disks in total.

For the first 20,000 extents, the extent size is equal to the AU size. After 20,000 extents and up to 40,000 extents, then extent sets are always allocated 8 at a time with the extent size equal to 4*AU size. If the AU size is 1 MB, this means the ASM file will grow 64 MB at a time (8 * 4 * 1 MB). If the file is coarse grained striped then it is striped across the 8 extent sets with stripes of 1 AU. Striping is always done at the AU level, not at the extent level. Thus every AU of a coarse grained file is on a different disk than the previous AU of that file no matter how large the file. After 40,000 extents, the extents are still allocated 8 at a time, but with an extent size equal to 16*AU size.

Fine Grained Striping

Fine grain striping puts 128 KB stripe units across groups of allocation units to improve latency.

- Disk group with 8 disks and external redundancy
- Default AU size of 1 MB in use
- First 1 MB extent written as 128 KB stripes across 8 AUs



ORACLE

Fine Grained Striping

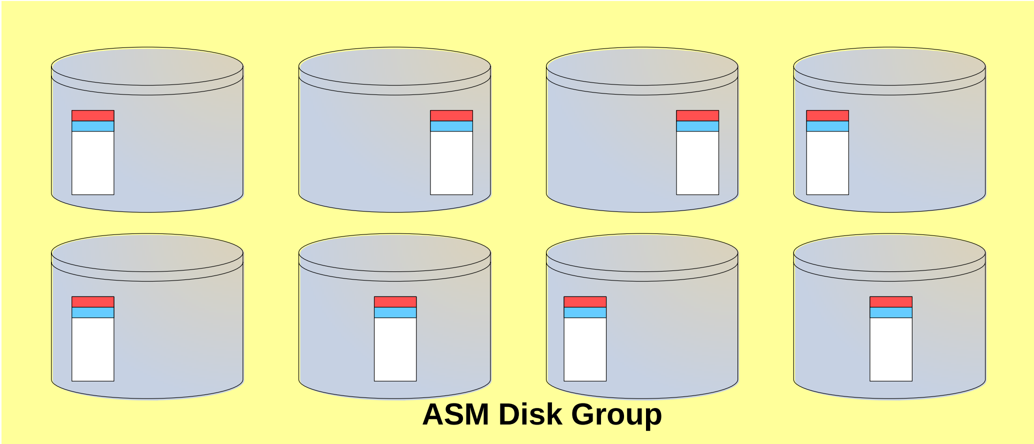
Fine grain striping splits data extents into 128 KB chunks and it is provided to improve latency for certain types of files by spreading the load for each extent across a number of disks. Fine grain striping is used by default for control files and online redo log files.

The diagram on this page shows how fine grain striping works. In this example, the first 1 MB extent of a new file ends up occupying the first 128 KB of 8 different allocation units spread across the eight disks in the disk group. Consequently, a one megabyte read or write is spread across eight disks instead of one...

Fine Grained Striping

Example:

- Disk group with 8 disks and external redundancy.
- Default AU size of 1 MB in use.
- Next 1 MB extent written as 128 KB stripes across the same 8 allocation units until they are full.

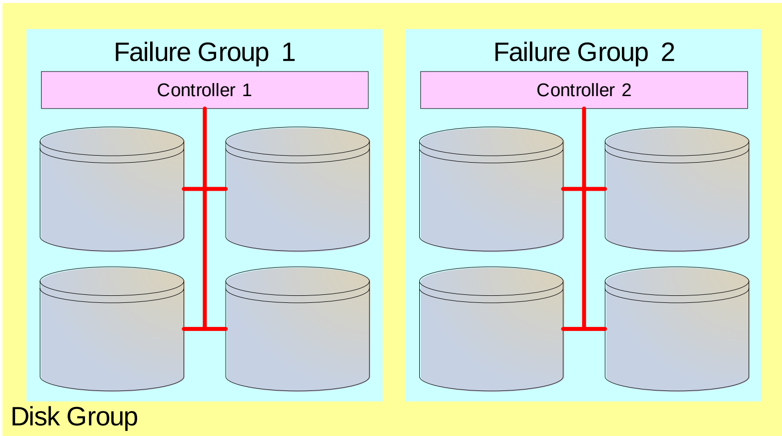


Fine Grained Striping (continued)

Continuing the previous example, the next one megabyte extent of space is spread across the second 128 KB of each of the same allocation units. This pattern continues until the first set of allocation units is filled and another set in allocated.

ASM Failure Groups

- A set of disks sharing a common resource whose failure needs to be tolerated.
- Mirrored extent copies stored in separate failure groups.
- Storage hardware dictates failure group boundaries.
  - Example based on isolating disk controllers:



ASM Failure Groups

Within a disk group, disks may be collected into failure groups. Failure groups are the way a storage or database administrator specifies the hardware boundaries that ASM mirroring operates across.

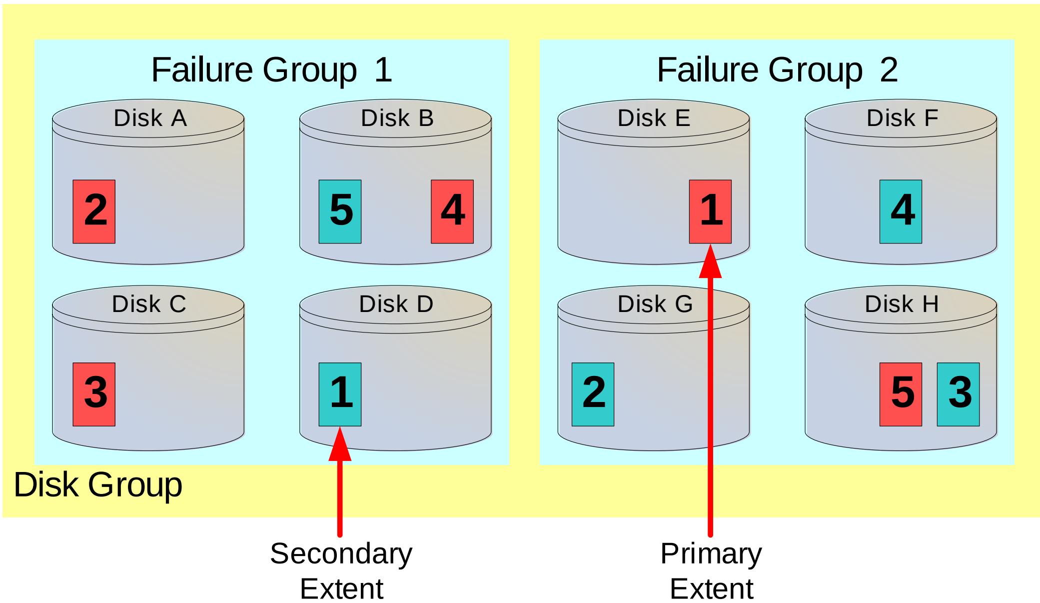
For example, all the disks attached to a single disk controller could be specified to be within a common failure group. This would lead to file extents being mirrored on disks connected to separate controllers. Additionally, an administrator can configure ASM to chose a default failure group policy. The default policy is that each individual disk is in its own failure group.

You can group disks into failure groups using whatever criteria you need. Failure groups can be used to protect from the failure of individual disks, disk controllers, I/O network components and even entire storage systems. Typically, an administrator would analyze their storage environment and would organize failure groups to mitigate against specific failure scenarios.

It is up to the database or storage administrator to determine what is the best failure group configuration for their installation.

Stripe and Mirror Example

Normal redundancy disk group with 8 disks in total, spread across two failure groups.



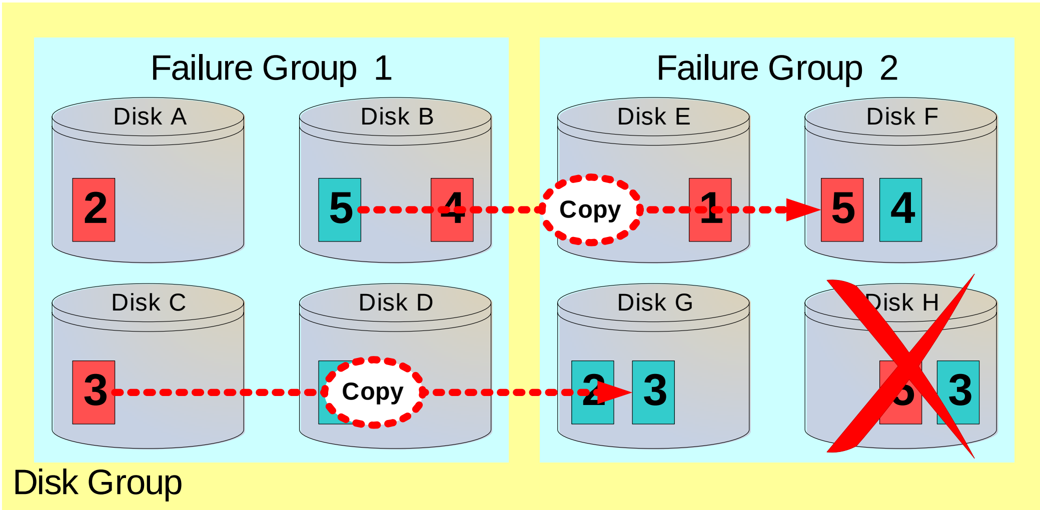
Stripe and Mirror Example

The diagram on this page illustrates and example showing striping and mirroring in a normal redundancy disk group. The red blocks represent a file with five extents being striped across five of the eight disks in the disk group. The blue blocks represent mirrored copies of the file's extents. Note that regardless of the distribution of extents across the various disks and failure groups, each extent has exactly one copy in each failure group.

When a file is allocated, the primary extents are allocated for performance and the secondary copies are allocated for integrity. For this reason all database reads are made against the primary extents by default.

Failure Example

If disk H fails, then the extents it contained are re-created on surviving disks from surviving mirrors.



Failure Example

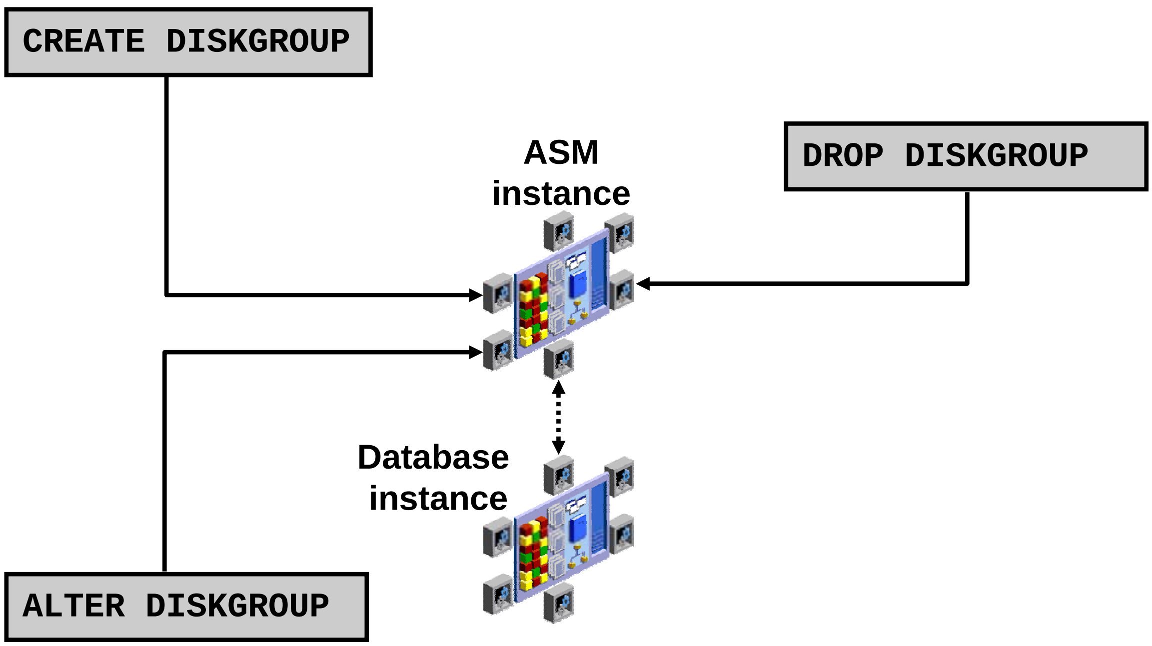
Extending the example on the previous page, imagine that disk H fails and the data it contains is no longer accessible. This failure requires that all the extents on the failed disk need to be recovered and copied to another disk.

Extents three and five are copied from the surviving copy to a free region on another disk in the same failure group. In this example, extent five is copied from disk B to Disk F and extent three is copied from Disk C to Disk G.

The last step after a disk fails is that ASM drops the failed disk from the disk group.

The removal of a disk triggers essentially the same process, however in that case the extents on the disk being removed are first copied to an available alternative location.

Managing Disk Groups



Managing Disk Groups

The main goal of an ASM instance is to manage disk groups and protect their data. ASM instances also communicate file layout to database instances. In this way, database instances can directly access files stored in disk groups.

There are several disk group administrative commands. They all require the `SYSASM` or `SYSDBA` privilege and must be issued from an ASM instance.

You can add new disk groups. You can also modify existing disk groups to add new disks, remove existing ones, and perform many other operations. You can remove existing disk groups.

Creating and Dropping Disk Groups Using SQL*Plus

```
$ . oraenv
ORACLE_SID = [orcl] ? +ASM
The Oracle base for ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is
/u01/app/oracle
$ sqlplus / AS SYSASM
SQL*Plus: Release 11.2.0.1.0 - Production on Wed Jul 8 20:46:46 2009
Copyright (c) 1982, 2009, Oracle.  All rights reserved.
Connected to an idle instance.
SQL> CREATE DISKGROUP dgroupA NORMAL REDUNDANCY
FAILGROUP controller1 DISK
   '/devices/A1' NAME diskA1 SIZE 120G FORCE,
   '/devices/A2',
FAILGROUP controller2 DISK
   '/devices/B1',
   '/devices/B2';
```

```
SQL> DROP DISKGROUP dgroupA INCLUDING CONTENTS;
```


Creating and Dropping Disk Groups

Assume that ASM disk discovery identified the following disks in the `/dev` directory: `A1`, `A2`, `B1`, and `B2`. Also, assume that disks `A1` and `A2` are on a separate disk controllers from disks `B1` and `B2`. The first example in the slide illustrates how to configure a disk group called `DGROUPA` with two failure groups: `CONTROLLER1` and `CONTROLLER2`.

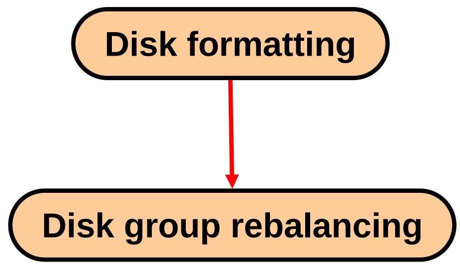
The example also uses the default redundancy attribute, `NORMAL REDUNDANCY`, for the disk group. You can optionally provide a disk name and size for the disk. If you do not supply this information, ASM creates a default name and attempts to determine the size of the disk. If the size cannot be determined, an error is returned. `FORCE` indicates that a specified disk should be added to the specified disk group even though the disk is already formatted as a member of an ASM disk group. Using the `FORCE` option for a disk that is not formatted as a member of an ASM disk group returns an error.

As shown by the second statement in the slide, you can delete a disk group along with all its files. To avoid accidental deletions, the `INCLUDING CONTENTS` option must be specified if the disk group still contains any files besides internal ASM metadata. The disk group must be mounted in order for it to be dropped. After ensuring that none of the disk group files are open, the group and all its drives are removed from the disk group. Then the header of each disk is overwritten to eliminate the ASM formatting information.

Adding Disks to Disk Groups

```
ALTER DISKGROUP dgroupA ADD DISK
  '/dev/sde1' NAME A5,
  '/dev/sdf1' NAME A6,
  '/dev/sdg1' NAME A7,
  '/dev/sdh1' NAME A8;
```

```
ALTER DISKGROUP dgroupA ADD DISK '/devices/A*';
```



Adding Disks to Disk Groups

This example shows how to add disks to a disk group. You execute an `ALTER DISKGROUP ADD DISK` command to add the disks. The first statement adds four new disks to the `DGROUPA` disk group.

The second statement demonstrates the interactions of discovery strings. Consider the following configuration:
`/devices/A1` is a member of disk group `DGROUPA`.
`/devices/A2` is a member of disk group `DGROUPA`.
`/devices/A3` is a member of disk group `DGROUPA`.
`/devices/A4` is a candidate disk.
The second command adds `A4` to the `DGROUPA` disk group. It ignores the other disks, even though they match the discovery string, because they are already part of the `DGROUPA` disk group. The diagram shows that, when you add a disk to a disk group, the ASM instance ensures that the disk is addressable and usable. The disk is then formatted and rebalanced. The rebalance process is time consuming because it moves extents from all files onto the new disk.

Note: Rebalancing does not block any database operations. The main impact that a rebalance process has is on the I/O load on the system. The higher the power of the rebalance, the more I/O load it puts on the system. Thus less I/O bandwidth is available for database I/Os.

Miscellaneous ALTER Commands

- Remove a disk from `dgroupA`:

```
ALTER DISKGROUP dgroupA DROP DISK A5;
```

- Add and drop a disk in a single command:

```
ALTER DISKGROUP dgroupA
 DROP DISK A6
 ADD FAILGROUP controller3
         DISK '/dev/sdi1' NAME A9;
```

- Cancel a disk drop operation:

```
ALTER DISKGROUP dgroupA UNDROP DISKS;
```


Miscellaneous ALTER Commands

The first statement in the slide shows how to remove one of the disks from the DGROUPA disk group. The second statement shows how you can add and drop a disk in a single command. The big advantage in this case is that rebalancing is not started until the command completes. The third statement shows how to cancel a disk drop operation. The UNDROP command operates only on pending drops of disks; it has no effect on drops that have completed.

The following statement rebalances the DGROUPB disk group, if necessary:

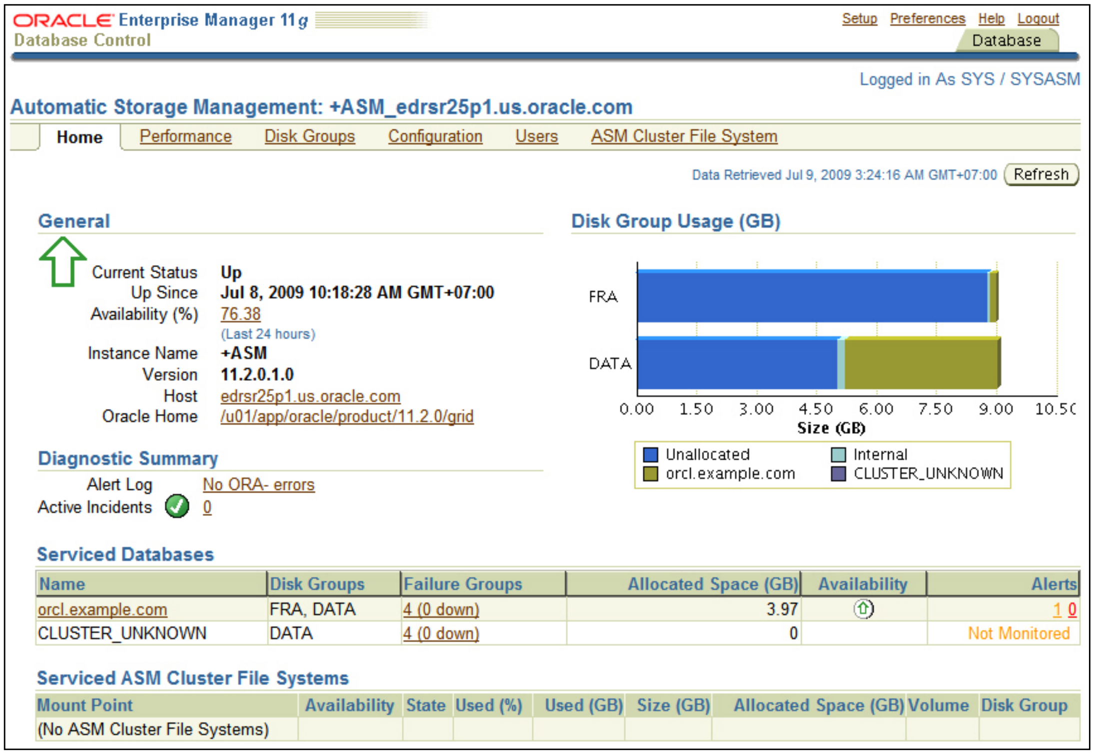
```
ALTER DISKGROUP dgroupB REBALANCE POWER 5;
```

This command is generally not necessary because it is automatically done as disks are added, dropped, or resized. However, it is useful if you want to use the POWER clause to override the default speed defined by the initialization parameter ASM_POWER_LIMIT. You can change the power level of an ongoing rebalance operation by reentering the command with a new level. A power level of zero causes rebalancing to halt until the command is either implicitly or explicitly reinvoked. The following statement dismounts DGROUPA:

```
ALTER DISKGROUP dgroupA DISMOUNT;
```

The MOUNT and DISMOUNT options allow you to make one or more disk groups available or unavailable to the database instances. The ability to manually unmount and mount is useful in a clustered ASM environment supporting a single instance, when that instance is failed over to a different node.

ASM Management Using Enterprise Manager

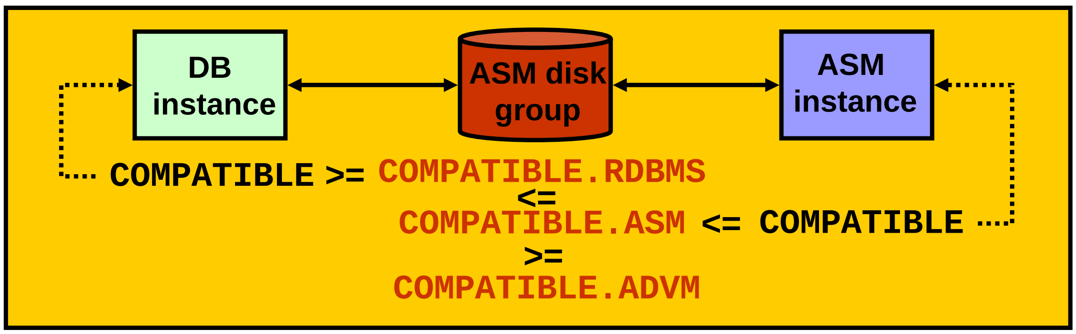


ASM Management Using Enterprise Manager

Oracle Enterprise Manager (EM) is Oracle’s family of management tools spanning database, middleware, applications, networks, IT infrastructure and more. Enterprise Manager delivers a browser based environment that provides a point and click alternative for common ASM administration tasks.

ASM Disk Group Compatibility

- Compatibility of each disk group is separately controllable:
  - ASM compatibility controls ASM metadata on-disk structure
  - RDBMS compatibility controls minimum consumer client level
  - ADVM compatibility determines whether a disk group can contain Oracle ASM volumes
- Setting disk group compatibility is irreversible.



ASM Disk Group Compatibility

There are three kinds of compatibility applicable to ASM disk groups; dealing with the persistent data structures that describe a disk group, the capabilities of the clients (consumers of disk groups), and the ability to contain volumes in a disk group. These attributes are called *ASM compatibility* , *RDBMS compatibility*, and *ADVM compatibility* respectively. The compatibility of each disk group is independently controllable. This is required to enable heterogeneous environments with disk groups from both Oracle Database 10*g* and Oracle Database 11*g*. These three compatibility settings are attributes of each ASM disk group:

- RDBMS compatibility refers to the minimum compatible version of the RDBMS instance that would allow the instance to mount the disk group. This compatibility dictates the format of messages that are exchanged between the ASM and database (RDBMS) instances. An ASM instance has the capability to support different RDBMS clients running at different compatibility settings. The database compatible version setting of each instance must be greater than or equal to the RDBMS compatibility of all disk groups used by that database. Database instances are typically run from a different Oracle home than the ASM instance. This implies that the database instance may be running a different software version than the ASM instance. When a database instance first connects to an ASM instance, it negotiates the highest version that they both can support.

ASM Disk Group Compatibility (continued)

The compatibility parameter setting of the database, software version of the database, and the RDBMS compatibility setting of a disk group determine whether a database instance can mount a given disk group.

- ASM compatibility refers to the persistent compatibility setting controlling the format of data structures for ASM metadata on disk. The ASM compatibility level of a disk group must always be greater than or equal to the RDBMS compatibility level of the same disk group. ASM compatibility is concerned only with the format of the ASM metadata. The format of the file contents is up to the database instance. For example, the ASM compatibility of a disk group can be set to 11.0 while its RDBMS compatibility could be 10.1. This implies that the disk group can be managed only by ASM software whose software version is 11.0 or higher, whereas any database client whose software version is higher than or equal to 10.1 can use that disk group.
- The ADVM compatibility attribute determines whether the disk group can contain Oracle ASM volumes in the disk group. The value must be set to 11.2 or higher. Before setting this attribute, the `COMPATIBLE.ASM` value must be 11.2 or higher. Also, the ADVM volume drivers must be loaded.

The compatibility of a disk group needs to be advanced only when there is a change to either persistent disk structures or protocol messaging. However, advancing disk group compatibility is an irreversible operation. You can set the disk group compatibility by using either the `CREATE DISKGROUP` or `ALTER DISKGROUP` commands.

Note: In addition to the disk group compatibilities, the *compatible* parameter (database *compatible version*) determines the features that are enabled; it applies to the database or ASM instance depending on the *instance_type* parameter. For example: Setting it to 10.1 would preclude use of any features introduced in Oracle Database 11*g* (disk online/offline, variable extents, and so on).

ASM Disk Group Attributes

Name	Property	Values	Description
au_size	Create, Alter	1|2|4|8|16|32|64MB	Size of allocation units in the disk group
compatible.rdbms	Create, Alter	Valid database version	Format of messages exchanged between DB and ASM
compatible.asm	Create, Alter	Valid ASM instance version	Format of ASM metadata structures on disk
compatible.advm	Create, Alter	Valid ASM instance version	Allows Oracle ASM volumes in disk group
disk_repair_time	Create, Alter	0 M to 2^{32} D	Length of time before removing a disk once OFFLINE
template.*tname*.redundancy	Alter	UNPROTECT|MIRROR|HIGH	Redundancy of specified template
template.*tname*.stripe	Alter	COARSE|FINE	Striping attribute of specified template

```
CREATE DISKGROUP DATA2 NORMAL REDUNDANCY
DISK '/dev/sda1','/dev/sdb1'
ATTRIBUTE 'compatible.asm'='11.2';
```

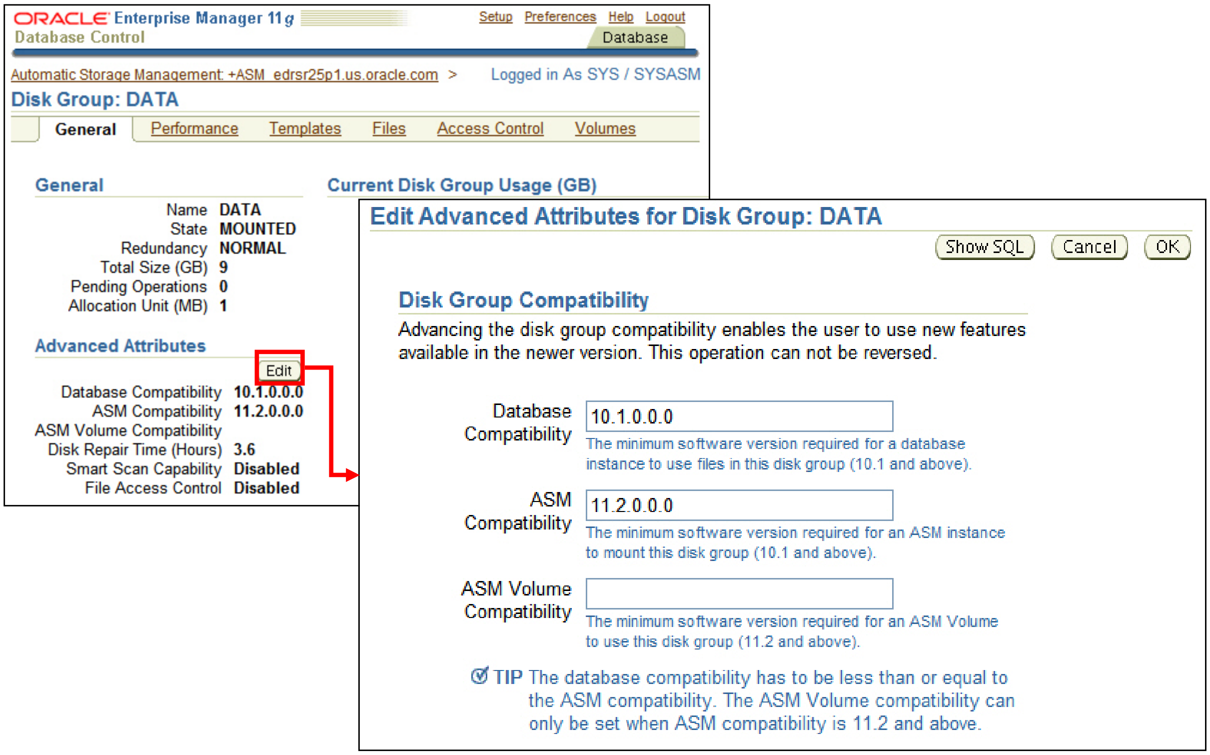

ASM Disk Group Attributes

Whenever you create or alter an ASM disk group, you have the ability to change its attributes using the new `ATTRIBUTE` clause of the `CREATE DISKGROUP` and `ALTER DISKGROUP` commands. These attributes are briefly summarized in the table given in the slide:

- ASM enables the use of different allocation unit (AU) sizes that you specify when you create a disk group. The AU can be 1, 2, 4, 8, 16, 32, or 64 MB in size.
- RDBMS compatibility: See the slide “ASM Disk Group Compatibility” for more information.
- ASM compatibility: See the slide “ASM Disk Group Compatibility” for more information.
- You can specify the `DISK_REPAIR_TIME` in units of minute (M), hour (H), or day (D). If you omit the unit, then the default is H. If you omit this attribute, then the default is 3.6H. You can override this attribute with an `ALTER DISKGROUP ... DISK OFFLINE` statement.
- You can also specify the redundancy attribute of the specified template.
- You can also specify the striping attribute of the specified template.

Note: For each defined disk group, you can look at all defined attributes through the `V$ASM_ATTRIBUTE` fixed view.

Using Enterprise Manager to Edit Disk Group Attributes



Using Enterprise Manager to Edit Disk Group Attributes

Enterprise Manager provides a simple way to store and retrieve environment settings related to disk groups.

You can set the compatible attributes from both the create disk group page and the edit disk group advanced attributes page. The `disk_repair_time` attribute is added to only edit the disk group advanced attributes page.

Note: For pre-11*g* ASM instances, the default ASM compatibility and the client compatibility are each 10.1. For 11*g* ASM instances, the default ASM compatibility is 11.2 and the database compatibility is 10.1.

Retrieving ASM Metadata

- Using SQL*Plus

```
SQL> SELECT f.type, f.redundancy, f.striped, f.modification_date,
a.system_created, a.name FROM v$asm_alias a, v$asm_file f WHERE
a.file_number = f.file_number and a.group_number = f.group_number
and type='DATAFILE';
TYPE      REDUND  STRIPE  MODIFICAT  S  NAME
--------  ------  ------  ---------  -  ---------------------
DATAFILE  MIRROR  COARSE  08-JUL-09  Y  SYSTEM.256.689832921
DATAFILE  MIRROR  COARSE  08-JUL-09  Y  SYSAUX.257.689832923
..
```

- Using `asmcmd`

```
ASMCMD> ls -l +DATA/orcl/datafile
Type      Redund  Striped  Time             Sys  Name
DATAFILE  MIRROR  COARSE   JUL 08 21:00:00  Y SYSTEM.256.689832921
DATAFILE  MIRROR  COARSE   JUL 08 21:00:00  Y SYSAUX.257.689832923
..
```


Retrieving ASM Metadata

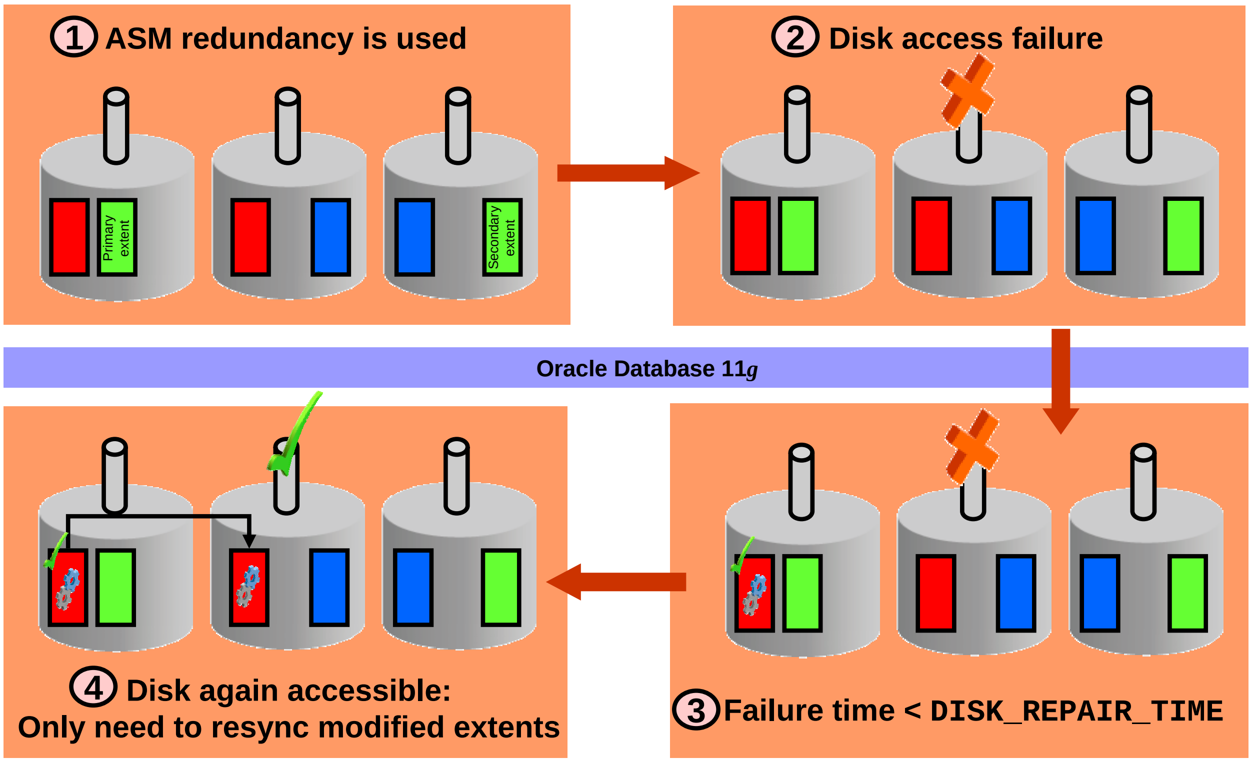
The ASM instance hosts memory-based metadata tables presented as dynamic performance views. This data can be queried using SQL*Plus, asmcmd, or Enterprise Manager.

Using SQL*Plus requires knowledge of the SQL language and may involve join multiple dynamic performance views to retrieve relevant information. This first example on the slide shows a join between `v$asm_file` and `v$asm_alias` to display metadata regarding the data files of a database. If this query is performed against the ASM instance, it could retrieve data files from multiple databases they way the syntax is designed. You would need to use additional filter conditions to restrict the output to a single database.

The `asmcmd` utility has the advantage of being able to connect to the ASM instance and retrieve metadata without knowledge of the SQL language. It used a style similar to Unix notation. The second example on this slide uses asmcmd to retrieve the same metadata as the SQL example did. Another advantage of this example is that the output is restricted to the data files of a single database since the path that is being listed contains the database name of `orcl`, and the file type of `data file`. Therefore, what appears as directories in `asmcmd` would require SQL filter conditions using the `WHERE` clause to give the same result.

Note: Enterprise Manager Database Control is able to display most of the ASM metadata by simply navigating among the various ASM web pages.

ASM Fast Mirror Resync Overview



ASM Fast Mirror Resync Overview

ASM fast mirror resync significantly reduces the time required to resynchronize a transient failure of a disk. When a disk goes offline following a transient failure, ASM tracks the extents that are modified during the outage. When the transient failure is repaired, ASM can quickly resynchronize only the ASM disk extents that have been affected during the outage.

This feature assumes that the content of the affected ASM disks has not been damaged or modified.

When an ASM disk path fails, the ASM disk is taken offline but not dropped if you have set the `DISK_REPAIR_TIME` attribute for the corresponding disk group. The setting for this attribute determines the duration of a disk outage that ASM tolerates while still being able to resynchronize after you complete the repair.

Note: The tracking mechanism uses one bit for each modified allocation unit. This ensures that the tracking mechanism very efficient.

Summary

In this lesson, you should have learned how to:

- Manage the ASM instance using SQL*plus, asmcmd, and Enterprise Manager
- Create and drop ASM disk groups
- Specify ASM compatibility attributes
- Extend ASM disk groups
- Compare methods of retrieving ASM metadata

ORACLE