CHAPTER 20

Recovering Oracle Databases

In this chapter you will learn how to

- Recover from loss of a controlfile
- Recover from loss of a redo log file
- Recover from loss of a system-critical datafile
- Recover from loss of a nonsystem-critical datafile

It is impossible to corrupt an Oracle database. The mechanism of instance recovery, where redo and undo are used to return the database to a consistent state after an instance failure, guarantees that. It is, however, possible to lose data following media failure—if the DBA has not taken appropriate precautions. The precautions are simple: to run the database in archivelog mode; to multiplex the controlfile, the online logfiles, and the archive log files; and to back up datafiles and archive log files. Following media failure, the backups and the archive logs can be used to recover the database up to the point of the failure, without loss of even one committed row of data. But whereas instance recovery is automatic—indeed, unavoidable—media recovery is a manual process. This chapter will go through elementary recovery techniques. More advanced techniques, applicable to more complex problems, will be covered in later chapters.

Recovery Structures and Processes

Following media failure, there are different techniques for recovery, depending on which files were damaged. The database consists of three file types: the controlfile, the online redo log files, and the datafiles. Recovery from damage to the controlfile or the online redo log files is a trivial exercise, provided that they were multiplexed. Recovery from damage to one or more datafiles is more complex, but still straightforward.

A damaged controlfile can be replaced with a multiplexed copy or re-created with a CREATE CONTROLFILE command. In extreme circumstances, it can be restored from a backup, but this should never be necessary following media failure, if you have followed a suitable multiplexing strategy.

A damaged online redo log file can be regenerated. Oracle provides an ALTER DATABASE CLEAR LOGFILE GROUP # command (where # is the number of the group with a damaged member), which will delete and re-create the members of a logfile group. If the database is running in archivelog mode (and it should be), the logfile group must have been archived before Oracle will permit execution of the clear logfile command. This is because clearing an unarchived log file group would mean that the archive log stream would be missing one logfile, and therefore that recovery would not be possible. There is a variation on the command, ALTER DATABASE CLEAR UNARCHIVED LOGFILE GROUP #, which will delete and re-create a logfile even if it has not been successfully archived, but after executing this command it is absolutely vital to perform a whole database backup.

A damaged datafile requires use of backups and archive logs. Following media failure resulting in damage to a datafile, there are two options for recovery: complete recovery, meaning no loss of data, and incomplete recovery, where you deliberately lose work by stopping the recovery process before it has completed. Incomplete recovery is an advanced procedure dealt with in Chapter 27. Complete recovery is a two-stage process. First, the damaged file must be restored from a backup. Second, the restored file must be recovered, by using redo information in the archive logs to bring it forward in time until it is synchronized with the rest of the database.

EXAM TIP In the Oracle environment, "restore" means to replace a damaged or missing file with a backup; "recover" means to synchronize the file with the rest of the database by use of archive logs.

Since online redo logs are never backed up by RMAN, RMAN cannot be used to recover from damage to them; repairing online logfiles damaged by media failure can by done only with SQL*Plus, or through Database Control. The controlfile and datafiles can be restored and recovered by RMAN; indeed, if you backed them up into backup sets, RMAN is your only option.

To open a database, all the controlfile copies, at least one member of each online logfile group, and all the online datafiles must be present and synchronized. If, during a startup, SMON finds that this is not the case, the startup will not complete. If a controlfile copy is damaged or missing, the startup will stop in NOMOUNT mode. A message is written out to the alert log detailing which copy (or copies) of the controlfile is damaged. Assuming that the controlfiles are fine, SMON proceeds to open the database. During this phase, it checks the headers of all the online datafiles. If any are missing or damaged, appropriate error messages are written out to the alert log, and the database remains in mount mode. If all the online files are present and not damaged, but one or more of them are not synchronized, SMON will attempt to synchronize them by using the online redo logs. This is the process of instance recovery, detailed in Chapter 18, and will happen automatically. If the online logs required are not available, then the database cannot be opened. If one or more datafiles have been restored from a backup, they will almost certainly be so far out-of-date that the online redo logs will not go far enough back in time to recover them: this is when you must use archive log files for the recovery, which is a procedure that must be initiated manually—from SQL*Plus if you are backing up with operating system commands, or with RMAN if (as Oracle strongly advises) you have committed to using RMAN for your backups.

If the media damage occurs while the database is open, the effect will depend on which files were affected. Damage to any controlfile copy will result in the instance terminating immediately. Damage to a datafile that is part of the SYSTEM tablespace or the active undo tablespace will have the same effect. But damage to an online log will not terminate the instance, as long as there is a surviving member of the logfile group. In fact, the instance will continue to function, and your end users will not even notice. But error messages will be written out to the alert log, and the situation should be corrected without delay; such corrections can and should be done online, while people continue to work. Damage to a datafile that is part of a tablespace other than SYSTEM or the active undo tablespace will also not result in an instance failure, but clearly the end users may have problems, because a part of the database will be missing. How your application will react to this is unpredictable—it will depend completely on how the application is structured. The restore and recovery of damaged datafiles can be done online, provided that they are not datafiles belonging to SYSTEM or the undo tablespace. Finally, damage to the tempfiles that make up your temporary tablespaces may not be noticed by the end users at all. Oracle does not validate the

existence of tempfiles until they are needed, and a well-tuned database may never need them. This means that tempfiles can be missing for some time before there is any noticeable effect. It also means that a damaged tempfile can be dropped and re-created at any time, unless it happens to be in use at that moment.

As with backups, a restore can be done with RMAN or with operating system utilities. But if your RMAN backups were to backup sets, rather than as image copies, the restore can be done only with RMAN: there is no other way to extract datafiles from a backup set. Recovery after a restore can be carried out with SQL*Plus commands or with RMAN, but the same restriction applies: only RMAN can extract archive logs from a backup set.

Recovery from Media Failure

Restore and recovery following media failure is covered in much greater detail in later chapters and the second OCP examination, but it is necessary to know the rudiments of recovery from simple problems for the first examination too. These simple problems are loss of one copy of a multiplexed controlfile and an online redo log file, and complete recovery following loss of critical and noncritical datafiles.

Recovery from Loss of a Multiplexed Controlfile

As long as a surviving multiplexed copy of the controlfile exists, recovery from loss of a controlfile is simple. Just replace it with a surviving copy of the controlfile. To restore the damaged or missing controlfile copy from a backup would be useless in these circumstances, because all copies of the controlfile must be identical; clearly, a restored copy would not be synchronized with the surviving copies, nor with the rest of the database.

Virtually the moment the damage occurs, the instance will terminate. As ever, the DBA's first reaction to a crashed instance should be to attempt a startup. This will fail, in NOMOUNT mode, with an appropriate error message. The alert log will state which controlfile copy is missing, and also—in the section listing the nondefault initialization parameters—how many controlfiles there actually are, and where they are. At this point, you have three options. First, you could edit the parameter file to remove the reference to the missing or damaged controlfile, as shown in Figure 20-1.

This is fine, but your database will now be running on one fewer multiplexed copies, which will presumably be in breach of your security guidelines. A better option is therefore to replace the damaged file with a copy made from a surviving copy or indeed to change the CONTROL_FILES initialization parameter to replace the reference to the damaged file with a reference to a brand new file, and copy the surviving controlfile to that.

EXAM TIP Recovering from loss of a controlfile will entail downtime. It cannot be done online.

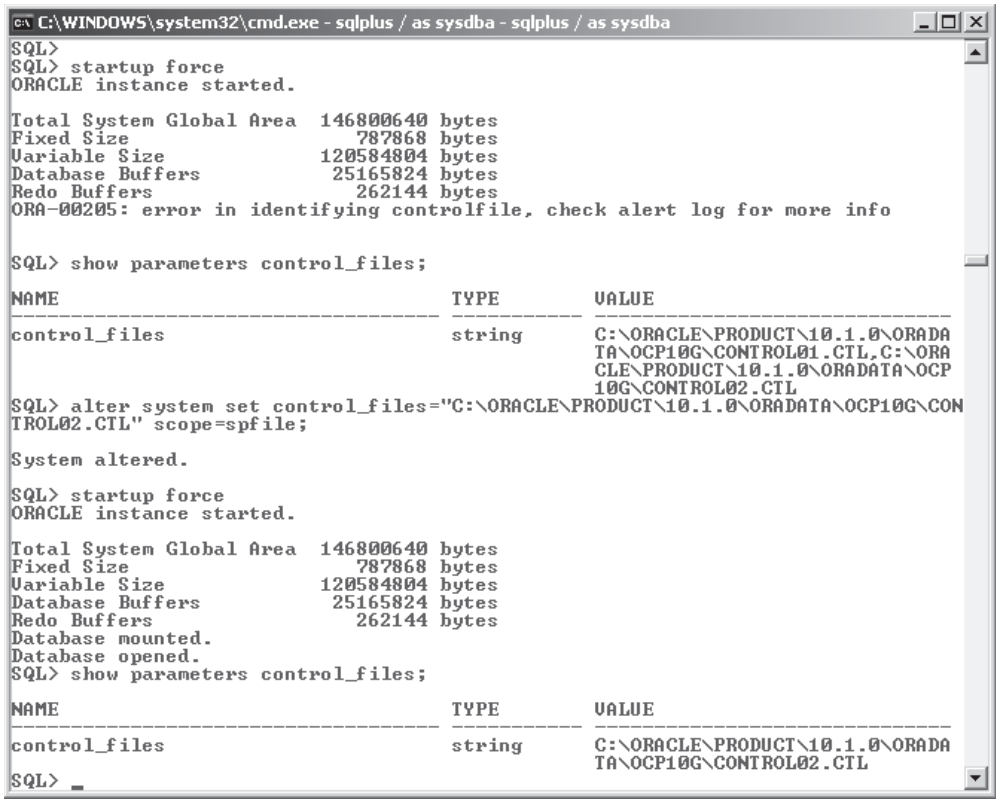

Figure 20-1 Removing the reference to a damaged controlfile

Exercise 20-1: Recovering from Loss of a Controlfile

In this exercise, you will simulate the loss of a multiplexed controlfile and replace it with a copy.

1. Connect to your database with SQL*Plus, and ensure that your controlfile is multiplexed with this query:

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

   This query must return at least two rows. If it does not, multiplex your controlfile by following the instructions given in Chapter 18, illustrated in Figure 18-5.

2. Simulate damage to a controlfile by crashing the database and renaming one of your controlfiles. Note that on Windows you may have to stop the

Windows service before Windows will let you rename the file, and start it again afterward.

3. Issue a startup command. The startup will stop in nomount mode, with an "ORA-00205: error in identifying controlfile, check alert log for more info" error message.
4. Copy your surviving controlfile to the name and location of the file you renamed.
5. Issue another startup command, which will be successful.

TIP Many DBAs do not like to copy a surviving controlfile over a damaged one, because it is all too easy to copy accidentally the damaged controlfile over the surviving one. It is safer to copy the surviving controlfile to a new file, and edit the control_files parameter to change the reference to the damaged file to the new file.

Recovery from Loss of a Multiplexed Online Redo Log File

Provided that the online redo log files are multiplexed, loss of one member will not cause any downtime, but there will be messages in the alert log telling you that there is a problem. If you can stand the downtime, you can shut down the database and copy a surviving member of the group over the damaged or missing member, but clearly this is not an option if the database is to remain open.

For open recovery, use the ALTER DATABASE CLEAR LOGFILE command to delete the existing files (or at least, those that still exist) and create new ones as shown in Figure 20-2. This can be done only if the logfile is inactive. If you attempt to clear the current logfile group, or the previous one that is still active, you will receive an error. Furthermore, if the database is in archivelog mode, the logfile group must have been archived.

EXAM TIP Recovery from loss of a multiplexed online redo log file can be done while the database is open, and therefore does not entail any downtime.

Exercise 20-2: Recovering a Lost Multiplexed Online Log File

This exercise will simulate loss of a logfile member and then, while the database is open, diagnose the problem and clear it.

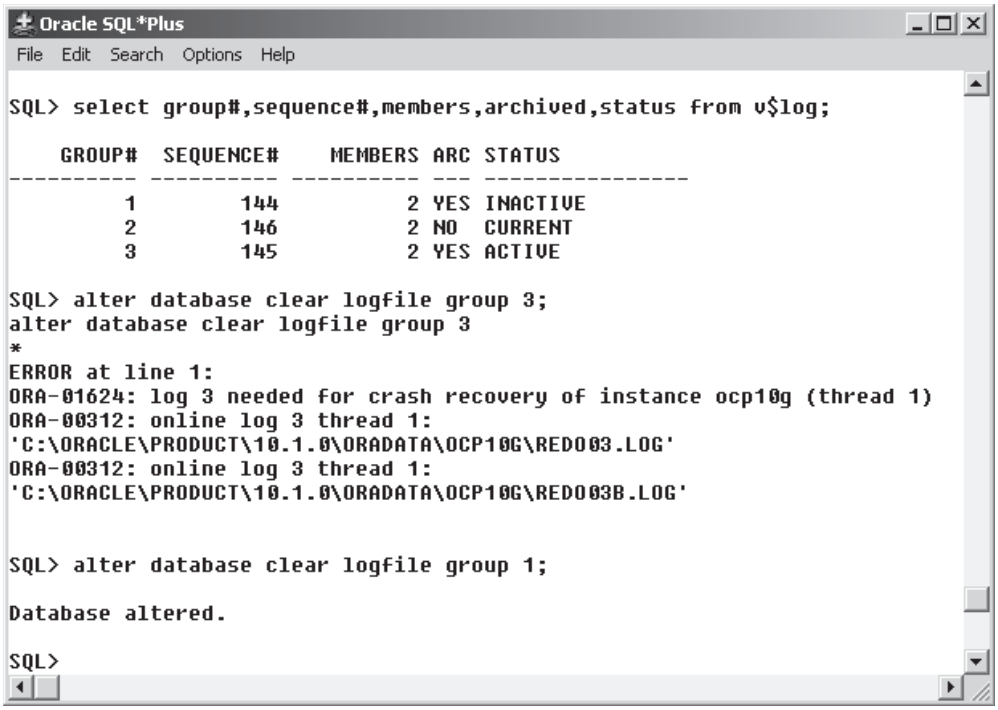

Figure 20-2 Clearing a logfile group with SQL*Plus

1. Using SQL*Plus, connect to your database as user SYS with SYSDBA privilege.

```
SQL> connect / as sysdba;
```

2. Observe the state of your online logs with the following query:

```
SQL> select group#,status,member from v$logfile order by group#;
   GROUP# STATUS  MEMBER
------------------------------------------------------------------------
        1         C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\REDO01.LOG
        1         C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\REDO01B.LOG
        2         C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\REDO02.LOG
        2         C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\REDO02B.LOG
        3         C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\REDO03.LOG
        3         C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\REDO03B.LOG
```

   Confirm that you do have at least two members of each group and that all the members have the STATUS column on NULL, as in the example here. If any groups do not have two members, multiplex them immediately by following the instructions given in Chapter 18, Exercise 18-2. If any members do not have a STATUS of NULL, execute the command

```
SQL> alter system switch logfile;
```

   a few times to cycle through the groups, and then re-run the query.

3. Shut down the database:

```
SQL> shutdown immediate;
```

4. Using an operating system command, simulate media failure by deleting one of the members. For example, on Windows,

```
SQL> host del C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\REDO01.LOG
```

or on Unix,

```
SQL> host rm /oracle/product/10.1.0/oradata/ocp10g/redo01.log
```

5. Start up the database and simulate user activity by performing a few log switches.

```
SQL> startup;
SQL> alter system switch logfile;
SQL> alter system switch logfile;
SQL> alter system switch logfile;
```

6. Check the state of your logfile members.

```
SQL> select group#,status,member from v$logfile order by group#;
   GROUP# STATUS  MEMBER
---------- ------- ------------------------------------------------------
         1 INVALID C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\REDO01.LOG
         1         C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\REDO01B.LOG
         2         C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\REDO02.LOG
         2         C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\REDO02B.LOG
         3         C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\REDO03.LOG
         3         C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\REDO03B.LOG
```

Note that the missing file is now marked as being INVALID.

7. Connect to your database as user SYSTEM, using Database Control.
8. From the database home page, take the Administration tab, and then the Redo Logs link in the Storage section to bring up the window shown in Figure 20-3.
9. If the group with the problem (group number 1 in the example shown) is not INACTIVE, use the Switch Logfile choice in the Actions drop-down list and click Go to force log switches until it is inactive.
10. Clear the logfile group by selecting its radio button using the Clear Logfile choice in the Actions drop-down list, and clicking Go.
11. In your SQL*Plus session, confirm that the problem has been fixed.

```
SQL> select group#,status,member from v$logfile order by group#;
   GROUP# STATUS  MEMBER
---------- ------- ------------------------------------------------------
         1         C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\REDO01.LOG
         1         C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\REDO01B.LOG
         2         C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\REDO02.LOG
         2         C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\REDO02B.LOG
         3         C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\REDO03.LOG
         3         C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\REDO03B.LOG
```

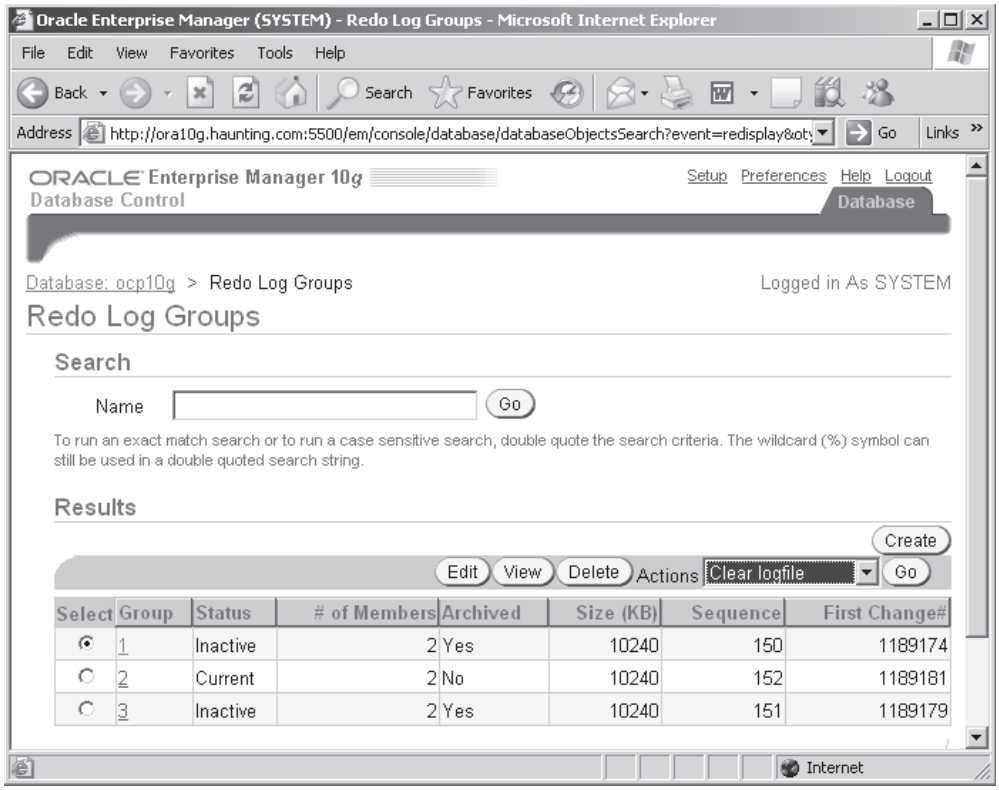

Figure 20-3 Redo logs, as shown in Database Control

Recovery from Loss of Datafiles

Media failure resulting in damage to one or more datafiles requires use of restore and recover routines: a backup of the datafile must be restored, and then archive redo logs applied to it to synchronize it with the rest of the database. There are various options available, depending on whether the database is in archivelog mode or not, and whether the file damaged is one that is critical to Oracle's running or a noncritical file containing "only" user data.

Recovery of Datafiles in Noarchivelog Mode

There is no concept of recovery when in noarchivelog mode, because the archive log files needed for recovery do not exist. Therefore, only a restore can be done. But if a restored datafile is not synchronized with the rest of the database by application of

archive redo log files, it cannot be opened. The only option when in noarchivelog mode is therefore to restore the whole database: all the datafiles, and the controlfile. Provided that all these files are restored from a whole offline backup, after the restore you will have a database where all these files are synchronized, and thus a database that can be opened. But you will have lost all the work done since the backup was taken.

Once the full restore has been done, the database will still be missing its online redo log files, because they were never backed up. For this reason, the post-restore startup will fail, with the database being left in mount mode. While in mount mode, issue ALTER DATABASE CLEAR LOGFILE GROUP <group number> commands to re-create all the logfile groups. Then open the database. If you do the restore through the Database Control interface to RMAN, this process will be fully automatic.

In noarchivelog mode, loss of any one of possibly hundreds of datafiles can be corrected only by a complete restore of the last backup. The whole database must be taken back in time, with the loss of user's work. Furthermore, that last backup must have been a whole, offline backup, which will have entailed downtime. It should by now be apparent that the decision to operate your database in noarchivelog mode should not be taken lightly.

EXAM TIP If in noarchivelog mode, your only option following loss of a datafile is a whole database restore. There can be no recovery.

Recovery of a Noncritical File in Archivelog Mode

In an Oracle database, the datafiles that make up the SYSTEM tablespace and the currently active undo tablespace (as specified by the UNDO_TABLESPACE parameter) are considered to be "critical." Damage to any of these will result in the instance terminating immediately. Furthermore, the database cannot be opened again until the damage has been repaired by a restore and recover exercise. Damage to the other datafiles, which make up tablespaces for user data, will not as a rule result in the instance crashing. Oracle will take the damaged files offline, making their contents inaccessible, but the rest of the database should remain open. How your application software will react to this will depend on how it is structured and written.

TIP Is it safe to run your application with part of the database unavailable? This is a matter for discussion with your developers and business analysts, and an important point to consider when deciding on how to spread your segments across tablespaces.

If your backups were done with RMAN, the restore and recovery operation of a damaged datafile will be completely automatic. RMAN will carry out the restore in the most efficient manner possible, making intelligent use of full and incremental backups and then applying the necessary archivelogs. If RMAN is linked to a tape library, it will load the tapes automatically to extract the files it needs.

The restore and complete recovery of a datafile can succeed only if all the archive log files generated since the last backup of the datafile are available. Either they must still be on disk in the archive log destination directories, or if they have been migrated to tape, they will be restored during the recovery operation. RMAN will do the extract from a backup set and restore to disk automatically. If for some reason an archive logfile is missing or corrupted, the recovery will fail, but since archive log destinations and RMAN backup sets can and should be multiplexed, you should never find yourself in this situation. If you do, the only option is a complete restore, and an incomplete recovery up to the missing archive, as described in Chapter 27, which will mean loss of all work done subsequently.

Exercise 20-3: Recovering from Loss of a Noncritical Datafile

First, create a tablespace and a segment within it, and back it up. Then simulate damage to the datafile. Diagnose the problem, and resolve it. The database will stay open for use throughout the whole exercise. At various points you will be asked to supply host operating system credentials, if you have not saved them in previous exercises: give a suitable Windows or Unix login, such as the Oracle owner.

1. Connect to your database as user SYSTEM using SQL*Plus, and create a tablespace. For example, on Windows,

```
SQL> create tablespace noncrit
 2  datafile 'C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\noncrit.dbf' size 2m;
```

   or on Unix,

```
SQL> create tablespace noncrit
 2  datafile '/oracle/product/10.1.0/oradata/ocp10g/noncrit.dbf' size 2m;
```

2. Create a table within the new tablespace and insert a row into it.

```
SQL> create table ex203 (c1 date) tablespace noncrit;
SQL> insert into ex203 values(sysdate);
SQL> commit;
```

3. Using Database Control, connect to your database as user SYSTEM.
4. From the database home page, take the Maintenance tab, then the Schedule Backup link in the Backup/Recovery section.
5. In the Schedule Backup: Strategy window, select Customized in the Backup Strategy drop-down box.
6. Select the Tablespaces radio button, and click Next.
7. In the Schedule Backup: Tablespaces window, click Add.
8. In the Tablespaces: Available Tablespaces window, select the radio button for your new NONCRIT tablespace, and click Select.
9. In the Schedule Backup: Tablespaces window, click Next.
10. In the Schedule Backup: Options window, leave everything on defaults and click Next.

11. In the Schedule Backup: Settings window, leave everything on defaults and click Next.
12. In the Schedule Backup: Schedule window, leave everything on defaults and click Next to schedule an immediate backup.
13. In the Schedule Backup: Review click Submit to run the backup.
14. Simulate a disk failure by corrupting the new datafile. On Windows, open the file with Windows Notepad, delete a few lines from the beginning of the file, and save it; it is important to use Notepad because it is one of the few Windows utilities that will ignore the file lock that Oracle places on datafiles. On Unix you can use any editor you please, such as vi. Make sure that the characters deleted are at the start of the file, to ensure that the file header is damaged.
15. Confirm that the file is damaged by attempting to query the table:

```
SQL> select * from ex203;
select * from ex203
              *
ERROR at line 1:
ORA-01578: ORACLE data block corrupted (file # 7, block # 9)
ORA-01110: data file 7:
'C:\ORACLE\PRODUCT\10.1.0\ORADATA\OCP10G\NONCRIT.DBF'
```

    If the damage is not yet apparent, repeat Step 14 until it is.

16. In your Database Control session, take the Maintenance tab from the database home page, and then the Perform Recovery link in the Backup/Recovery section.
17. In the Perform Recovery: Type window, select Datafiles in the Object Type drop-down box, and click Next.
18. In the Perform Recovery: Datafiles window, the new datafile will be listed. Select it, and click Next.
19. In the Perform Recovery: Review window shown in Figure 20-4, leave everything on defaults and click Submit.
20. When the operation has completed, return to your SQL*Plus prompt and bring the file online, specifying it by name or by number.

```
SQL> alter database datafile 7 online;
```

21. Confirm that the tablespace and the tables within it are now usable, with no loss of data.

```
SQL> select * from ex203;
C1
---------
21-OCT-04
```

22. Tidy up the database.

```
SQL> drop tablespace noncrit including contents and datafiles;
```

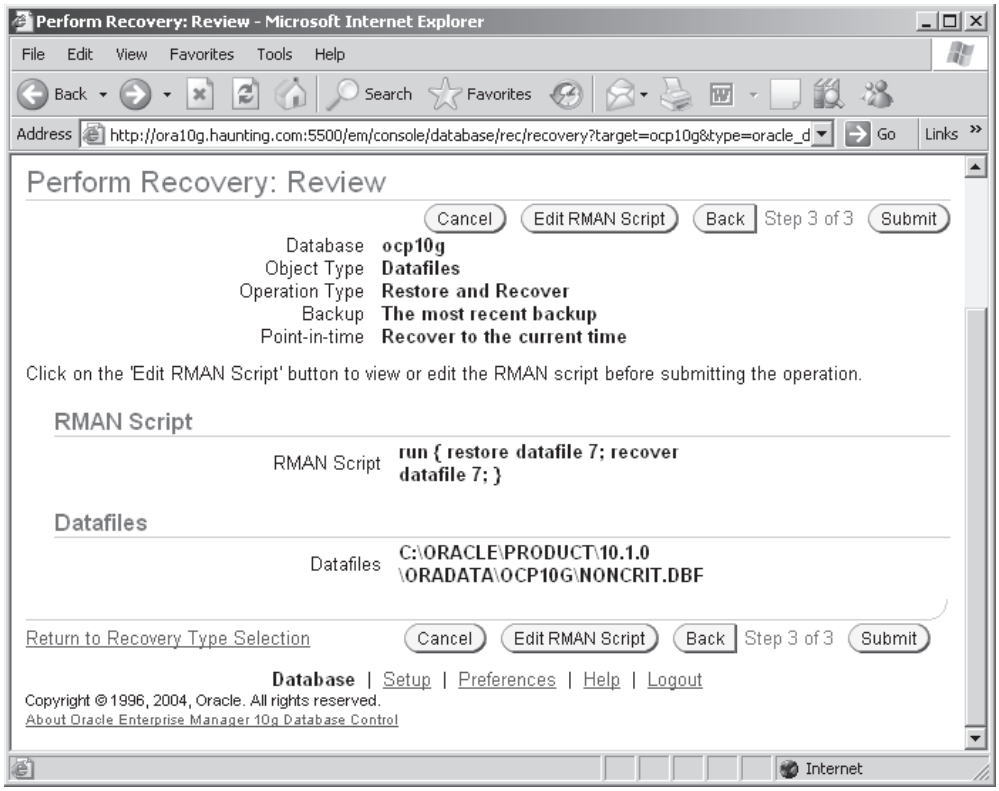

Figure 20-4 RMAN datafile restore and recover, with Database Control

Recovering from Loss of a Critical Datafile

The datafiles that make up the SYSTEM and currently active undo tablespace are considered critical by Oracle, meaning that it is not possible to keep the database open if they are damaged. If any portion of the SYSTEM tablespace were not available, parts of the data dictionary would be missing. Oracle cannot function without a complete data dictionary. If parts of the undo tablespace were not available, it would be possible that undo data required for maintaining transactional integrity and isolation would not be available, and Oracle can't take that chance either. Therefore, damage to these datafiles will cause the instance to terminate immediately.

TIP The critical datafiles should be on disk systems with hardware redundancy, such as RAID level 1 disk mirroring, so that in the case of media failure the files will survive and the database will remain open.

If the database does crash because of damage to critical datafiles, as ever, the first action is to attempt a startup. This will stop in mount mode, with error messages written to the alert log showing the extent of the damage. To recover, follow the same routine as that for a noncritical file, and then open the database. The restore and recover process is identical to that for a noncritical file, but it must be carried out in mount mode.

EXAM TIP Loss of a critical datafile will not mean loss of data, but it will mean loss of time.

Chapter Review

This chapter is a basic introduction to recovery techniques. First you saw how to recover from loss of a multiplexed controlfile; this is a simple exercise, but it does involve downtime. Second, you saw how to recover from the loss of a multiplexed online redo log file. This is also simple; and furthermore, it does not involve any downtime at all. Third, you saw the use of RMAN through Database Control to restore and recover datafiles. This is a more complex exercise, but the GUI tool makes it straightforward. If the files are noncritical, the database can remain open and available for use, though segments with extents in the affected files will not be available until the restore and recover is complete and the files are brought back online.

And as a final word, always remember that loss or damage to files resulting from media failure is never a reason to lose data—if the database has been suitably protected by file multiplexing, archiving, and backups.

Questions

1. Loss of which of these files will cause an open database to crash? (Choose three answers.)
   - **A.** A multiplexed controlfile
   - **B.** A multiplexed online logfile
   - **C.** A multiplexed archive log file
   - **D.** An active undo tablespace datafile
   - **E.** An active temporary tablespace tempfile
   - **F.** A datafile from the SYSAUX tablespace
   - **G.** A datafile from the SYSTEM tablespace
   - **H.** A datafile containing critical user data
2. Loss of which of these files will prevent the database from opening? (Choose five answers.)

A. A multiplexed controlfile
B. A multiplexed online logfile
C. A multiplexed archive log file
D. An active undo tablespace datafile
E. An active temporary tablespace tempfile
F. A datafile from the SYSAUX tablespace
G. A datafile from the SYSTEM tablespace
H. A datafile containing user data

3. A copy of a multiplexed controlfile is damaged. What should you do? (Choose the best answer.)
   A. Replace it with a surviving copy.
   B. Restore it with RMAN.
   C. Restore it with operating system commands.
   D. Re-create it with the CREATE CONTROLFILE command.
4. How could you diagnose problems with a multiplexed online logfile group member? (Choose the best answer.)
   A. Query the V$LOG view.
   B. Query the V$LOGFILE view.
   C. Query the V$LOGFILE_MEMBER view.
   D. You do not need to diagnose it; the instance will crash when the problem occurs.
5. You issue the command ALTER DATABASE CLEAR LOGFILE GROUP 2 and it fails with the message "ORA-01624: log 2 needed for crash recovery of instance ocp10g (thread 1)." What could be an explanation for this? (Choose the best answer.)
   A. Logfile group 2 has not been archived.
   B. Logfile group 2 is being used for recovery.
   C. Logfile group 2 is active.
   D. The group is not multiplexed.
6. Your database is in noarchivelog mode, and you lose a noncritical datafile. What can you do to minimize loss of data?
   A. Restore the one damaged file, and leave the rest of the database up-to-date.
   B. Restore all the datafiles, but leave the controlfile up-to-date.
   C. Restore the whole database, and clear the online redo logs.
   D. Restore the one damaged file, and apply the online redo logs.

7. In noarchivelog mode, what restore and recover options are available to you? (Choose two answers.)
   - **A.** Whole database restore
   - **B.** Partial restore
   - **C.** Online restore of noncritical datafiles
   - **D.** Offline restore of critical datafiles
   - **E.** Automatic recovery after an instance crash
8. In archivelog mode, which of the following could result in loss of data?
   - **A.** Loss of a nonmirrored datafile that is part of the SYSTEM tablespace
   - **B.** Loss of a nonmirrored datafile that is part of the active undo tablespace
   - **C.** Loss of a nonmultiplexed archive log and a noncritical datafile
   - **D.** Loss of a member from two or more multiplexed online logfile groups
9. Which of the following operations require a database shutdown?
   - **A.** Recovering from loss of a multiplexed controlfile in archivelog mode
   - **B.** Recovering from loss of a multiplexed online redo log file in noarchivelog mode
   - **C.** Restore and recovery of the SYSAUX tablespace
   - **D.** None of the above
10. You have backed up your datafiles and controlfile into a backup set, but your archive logs have not been backed up. If you need to restore and recover a datafile, which of the following routines would work? (Choose two answers.)
    - **A.** Restore with RMAN, recover with SQL*Plus.
    - **B.** Restore and recover with RMAN.
    - **C.** Restore with operating system utilities, recover with SQL*Plus.
    - **D.** Restore with operating system utilities, recover with RMAN.
    - **E.** Restore and recover with SQL*Plus.
11. If media damage destroys a datafile, what will the effect be at the next startup?
    - **A.** The startup will stop in nomount mode.
    - **B.** The startup will stop in mount mode.
    - **C.** It depends on whether the file is part of critical tablespace or a user tablespace.
    - **D.** It depends on whether the database is in archivelog mode.
12. After a whole restore of a database in noarchivelog mode, what must be done before the database can be opened?
    - **A.** The database must be recovered.
    - **B.** The instance must be recovered.

C. The online logs must be cleared.

D. The database can be opened now, but work will have been lost.

13. You issue an ALTER DATABASE CLEAR LOGFILE GROUP 3 command and receive the message "ORA-00350: log 3 of instance ocp10g (thread 1) needs to be archived." What could be a cause of this? (Choose two answers.)

A. The database is not in archivelog mode.

B. The first multiplexed copy of group 3 needs to be archived.

C. If the instance crashed, this group would be needed for instance recovery.

D. An archive log destination is full.

E. The archiver process has failed.

14. During a recovery, it becomes apparent that an archive log is missing. What will be the result?

A. The recovery will succeed, but some data will be missing.

B. The recovery will fail.

C. The recovery will continue, if the damaged file was not from the SYSTEM tablespace or the active undo tablespace.

D. You must issue an ALTER DATABASE CLEAR ARCHIVE LOG FILE command to regenerate the missing archive.

15. Which of the following is correct about using image copies for restore? (Choose the best answer.)

A. You can restore them only with operating system utilities.

B. You can restore them only with RMAN.

C. If they were directed to tape, you can restore them only with RMAN.

D. You can restore them with either RMAN or operating system utilities.

E. Image copies can be used only for whole database restore.

Answers

1. **A, D,** and **G.** Loss of these types of files will typically cause the instance to terminate. The instance will survive the loss of the other file types.
2. **A, D, F, G,** and **H.** For an instance to open, all controlfile copies and online datafiles (no matter what tablespace they are a part of) must be available and synchronized.
3. **A.** It is fine to replace the damaged copy with a surviving copy, because all copies are bound to be identical. Note that B and C cannot be applied to one copy: they would replace all the copies, and then recovery would be necessary to synchronize them with the rest of the database. D is possible, but again it would apply to all copies and is not necessary. So all techniques could work, but A is the best in these circumstances.

4. **B.** This is the view that will point out an invalid logfile.
5. **C.** This is the error generated if you attempt to clear a logfile group before it has become inactive.
6. **C.** This is your only option for a database running in noarchivelog mode. There is no possibility of recovery, because the redo logs needed are not being archived before they are overwritten by log switches.
7. **A** and **E.** In noarchivelog mode, restoring the whole database is the only option. B, C, and D require the database to be in archivelog mode. Instance recovery is always enabled, no matter what mode the database is running in.
8. **C.** Given the loss of these files, the restore would work, but the recovery would fail.
9. **A.** Any controlfile operation needs downtime.
10. **A** and **B.** These options are both possible. You must use RMAN to extract files from a backup set, but if the archive logs are still on disk they can be applied with either RMAN or SQL*Plus.
11. **B.** This will be the case with any missing or damaged datafiles, whether critical or not.
12. **C.** This will re-create the online logs; only then can D be done. Neither A nor B is necessary after a whole offline backup, which is the only option for noarchivelog mode.
13. **D** and **E.** Either one of these conditions could cause this error, because they will cause the archiving to fail.
14. **B.** This should never happen, though, if you have multiplexed archive log destinations as you should.
15. **D.** Image copies don't require the special capabilities of RMAN because they are identical to a copy made with an operating system command.