# Oracle 11g DBA Fundamentals Overview

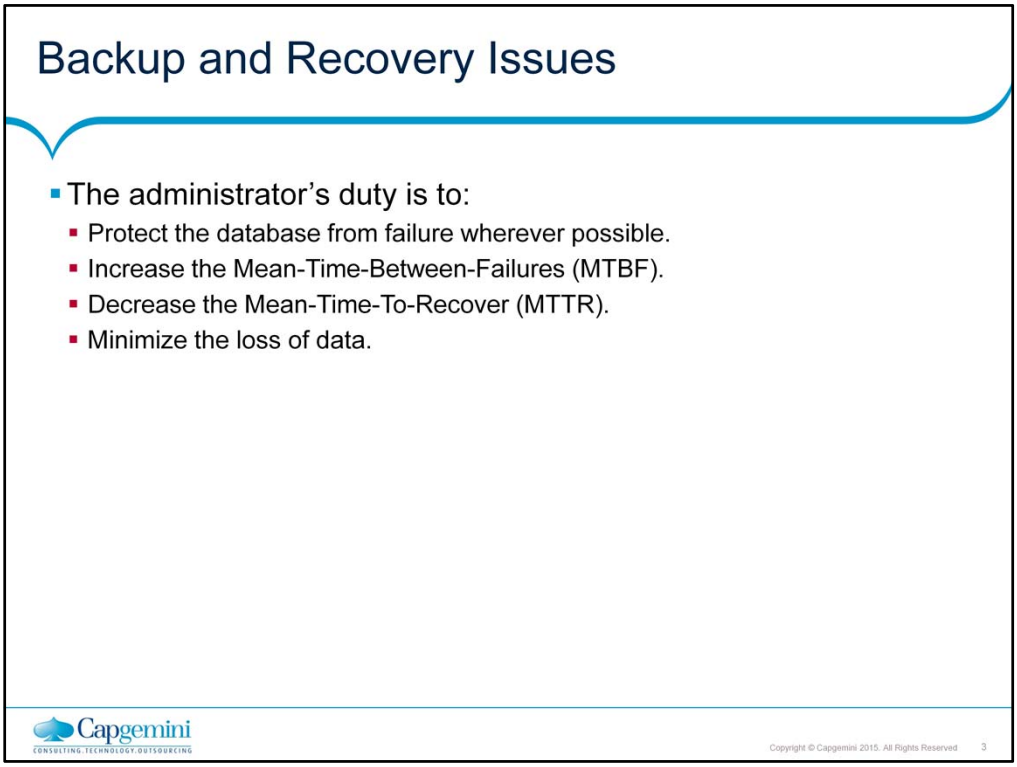
Lesson 13: Backup and Recovery Concepts

# Objectives

- After completing this you should be able to do the following:
  - Describe the basics of database backup, restore and recovery.
  - List the types of failure that may occur in an Oracle Database.
  - Describe ways to tune instance recovery.
  - Identify the importance of checkpoints, redo log files, and archived log files.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

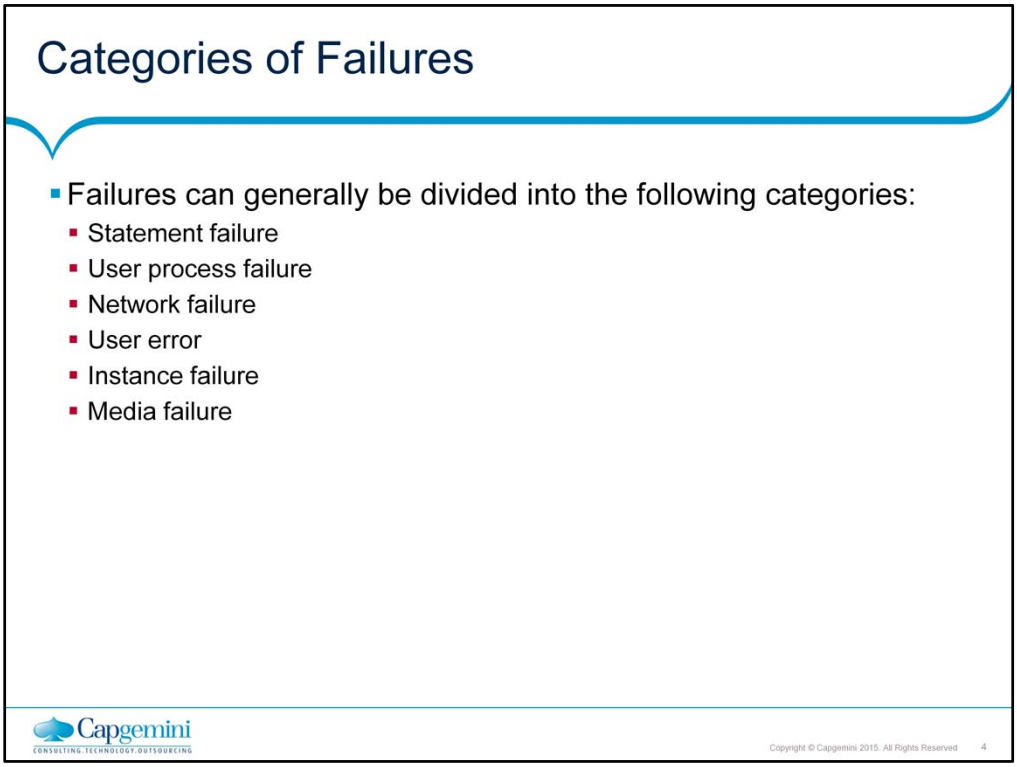
 2




Backup and Recovery Issues

The DBA's goal is to ensure the database is open and available when users need it. In support of that goal, the DBA, usually working with the system administrator:

- Anticipates and works against common causes of failure
- Works to increase the mean-time-between-failure, ensuring hardware is as reliable as possible, that critical components are protected by redundancy, and that operating system maintenance is performed in a timely manner. Oracle provides advanced configuration options to increase MTBF including:
  - Real Application Clusters (discussed in another course)
  - Streams (discussed in another course)
- Decreases the mean-time-to-recover, practicing recovery procedures in advance and configuring backups so they are readily available when needed
- Minimizes the loss of data. DBAs who follow accepted best practices can configure their databases so that no committed transaction is ever lost. Tools to assist in guaranteeing this include:
  - Archived redo logs (discussed later in this lesson)
  - Standby databases and Oracle Data Guard (discussed in another course)




Categories of Failures

Failures can be divided into a few broad categories:

Statement failure: A single database operation (select, insert, update, delete) fails.

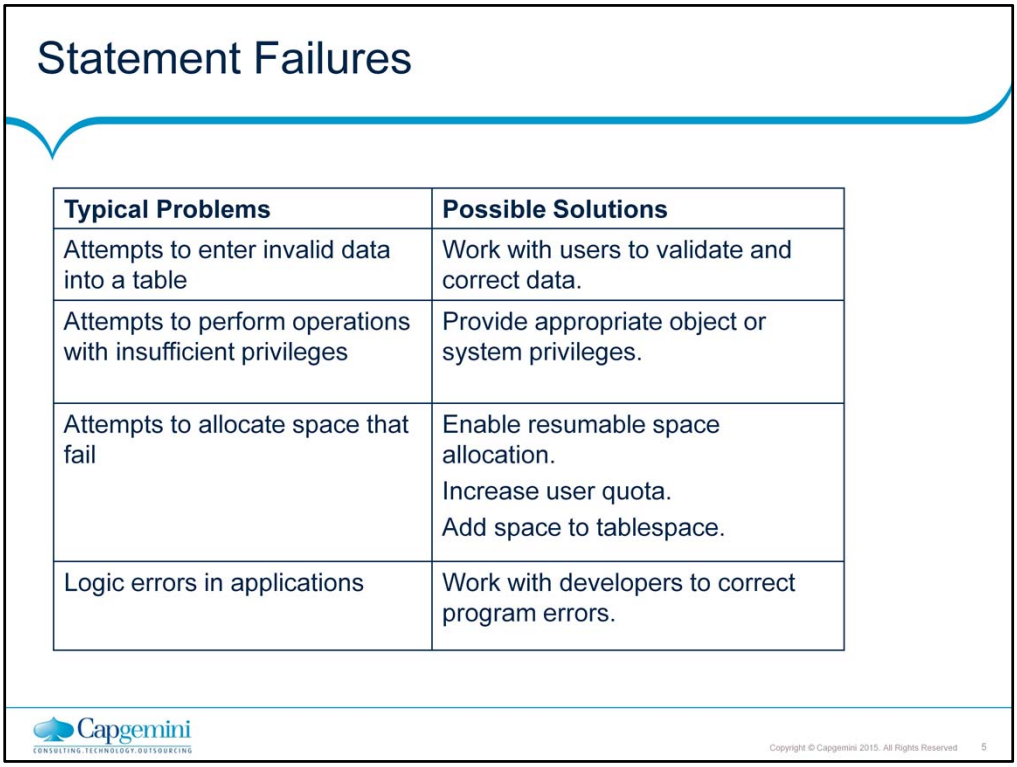
User process failure: A single database session fails.

Network failure: Connectivity to the database is lost.

User error: A user successfully completes an operation, but the operation was incorrect (dropping a table, entering incorrect data).

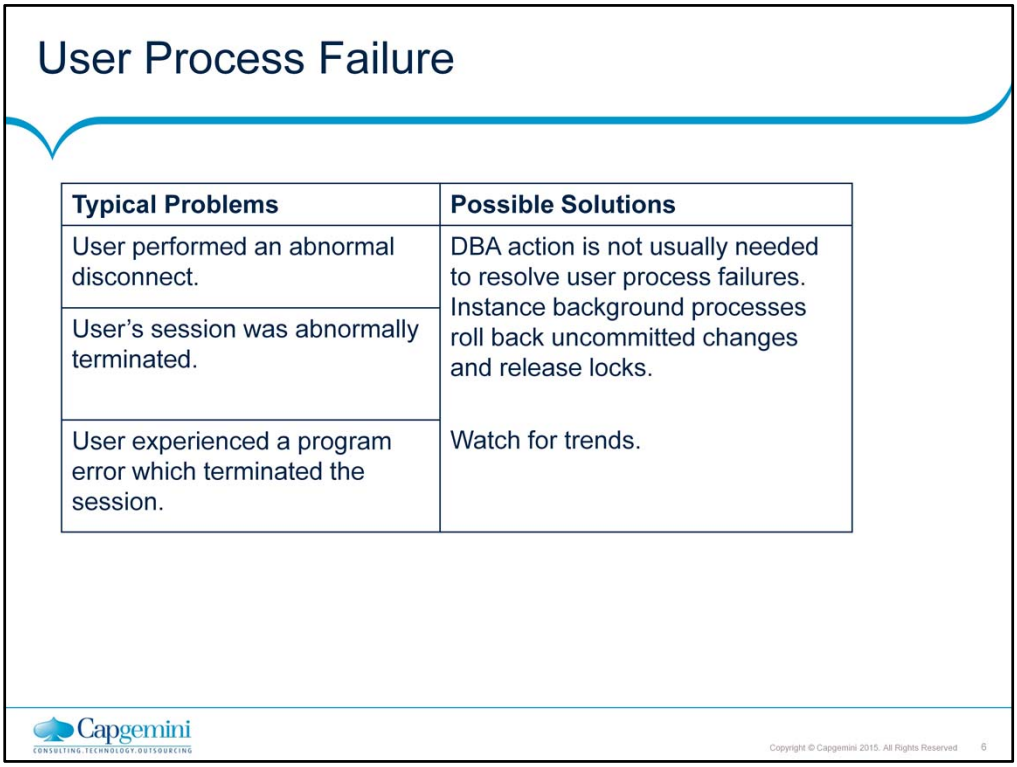
Instance failure: The database instance shuts down unexpectedly.

Media failure: One or more of the database files are lost (deleted, failed disk).




Statement Failures

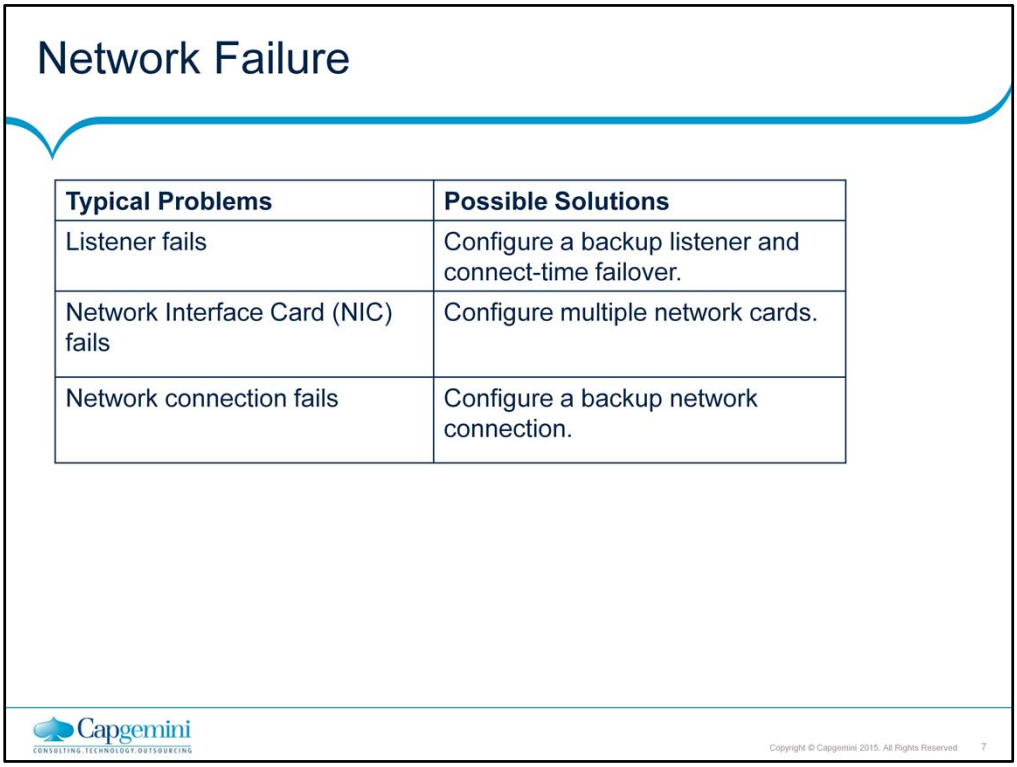
When a single database operation fails, DBA involvement may be needed to correct errors with user privileges or database space allocation.




User Process Failure

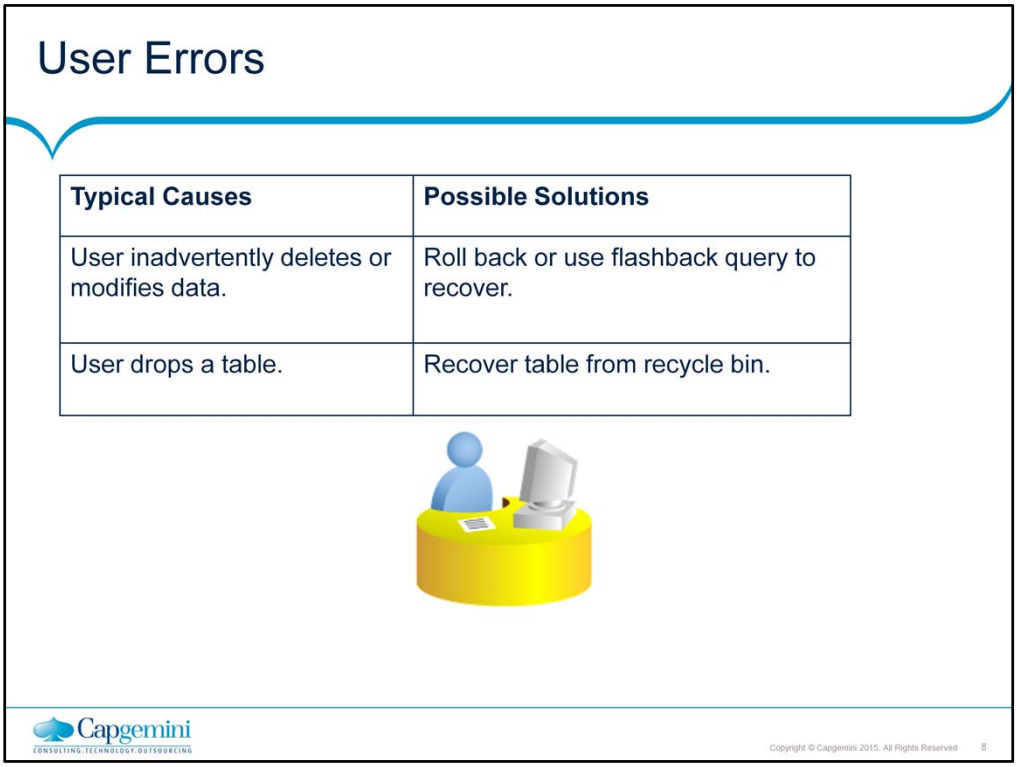
Users who are abnormally disconnected from the instance may have uncommitted work in progress that needs to be cleaned up. The PMON background process periodically polls server processes to ensure that their sessions are still connected. If PMON finds a server process whose user is no longer connected, PMON recovers from any ongoing transactions, including rolling back uncommitted changes and releasing any locks held by the failed session.

DBA intervention should not be required to recover from user process failure but the administrator should watch for trends. One or two users disconnecting abnormally is not a cause for concern. A small percentage of user process failures is normal. Consistent and systemic failures indicate other problems. A large percentage of abnormal disconnects may indicate a need for user training (teach them to log out rather than just terminating their programs). It may also be indicative of network or application problems.




Network Failure

The best solution to network failures is to provide redundant paths for network connections. Backup listeners, network connection, and network interface cards reduce the chance of network failures affecting system availability.

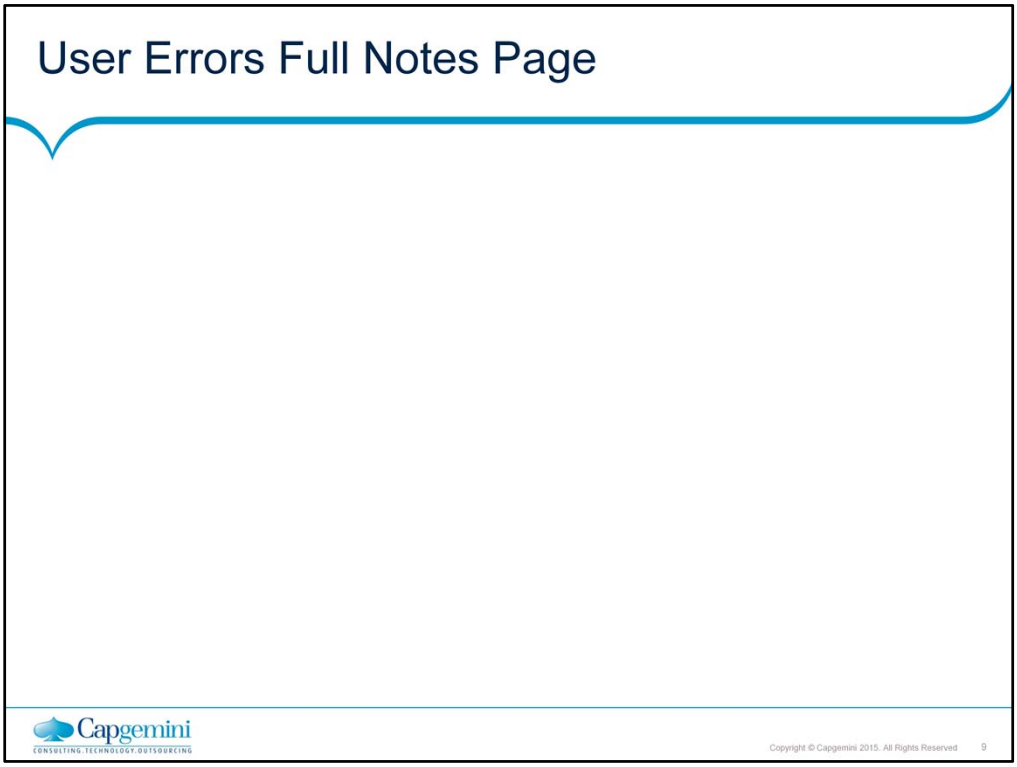



User Errors

Users may inadvertently delete or modify data. When that happens, the DBA may need to assist the user in recovering from the error. If the user has not yet committed or exited their program, they can simply roll back their operation. If the user has already committed the changes, flashback queries can be used to determine what the previous values were (and then the data can be updated to restore the original information.)

```
SQL> SELECT salary FROM employees WHERE
employee_id=100;
SALARY
------
25
SQL> SELECT salary FROM employees
2    AS OF TIMESTAMP(SYSTIMESTAMP-
INTERVAL'10' minute)
3    WHERE employee_id=100;
SALARY
------
24000
```

In cases where flashback queries are not possible because the undo retention period has been exceeded, the DBA may still be able to recover the original information through the use of Oracle LogMiner.




User Errors (continued)

Oracle LogMiner allows you to query your online redo logs and archived redo logs through an SQL interface. Transaction data may persist in online redo logs longer than it does in undo, and if you have configured archiving of redo information redo persists until you delete the archived files.

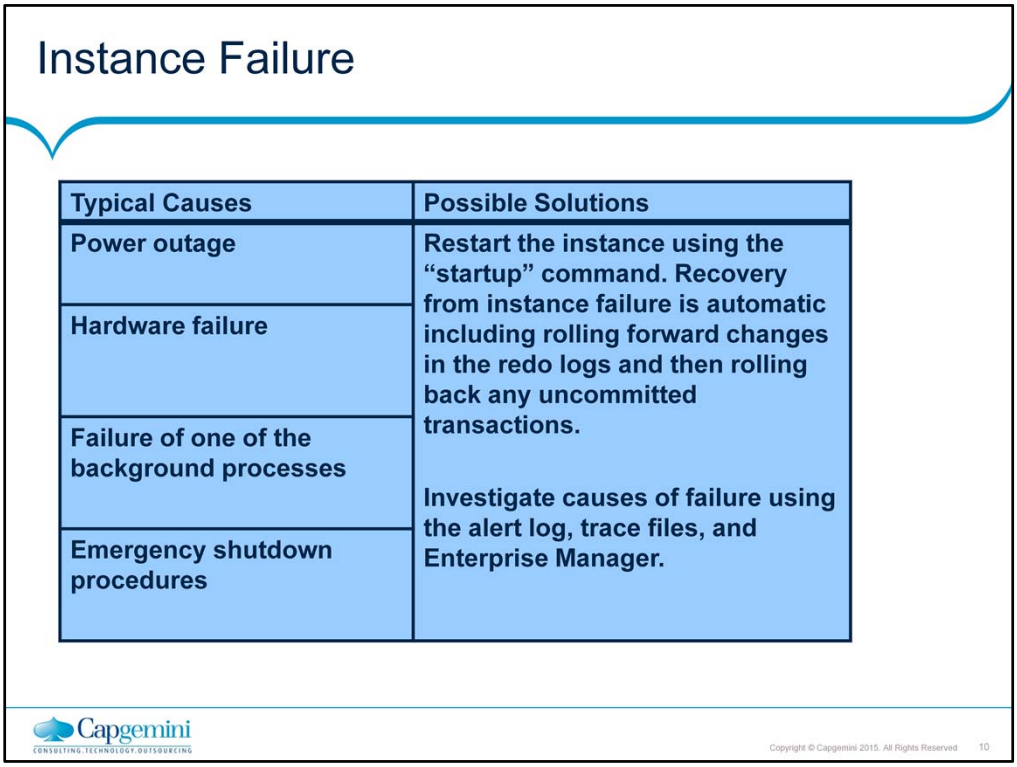
Oracle LogMiner is discussed in the Oracle Database 11g: Administration Workshop II course and in the Oracle Database: Utilities reference manual.

Users who drop tables can recover those tables from the recycle bin by flashing the table back to before the drop.

```
SQL> DROP TABLE hr.job_history;
Table dropped.

SQL> SELECT COUNT(*) FROM hr.job_history;
SELECT COUNT(*) FROM hr.job_history
              *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> FLASHBACK TABLE hr.job_history TO BEFORE
DROP;
Flashback complete.

SQL> SELECT COUNT(*) FROM hr.job_history;
  COUNT(*)
----------
        10
```

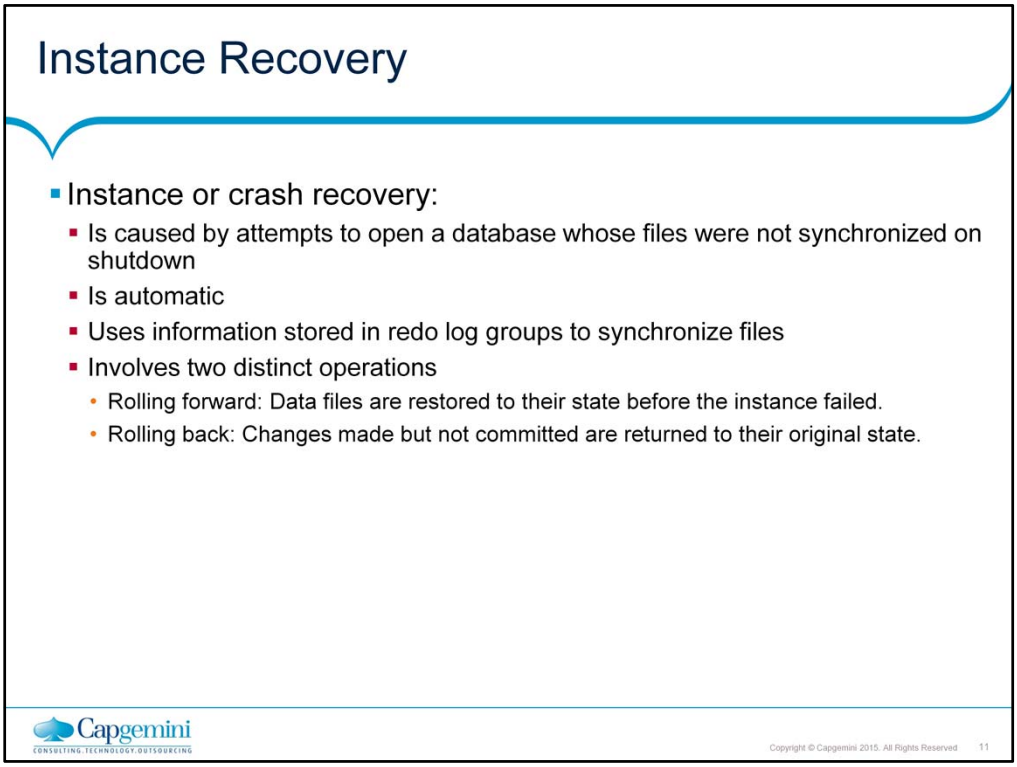
If the recycle bin has already been purged, or if the user dropped the table with the PURGE option, the dropped table can still be recovered by using point-in-time recovery (PITR) if the database has been properly configured.

PITR is discussed in the Oracle Database 11g: Administration Workshop II course and in the Oracle Database: Backup and Recovery Advanced User's Guide.



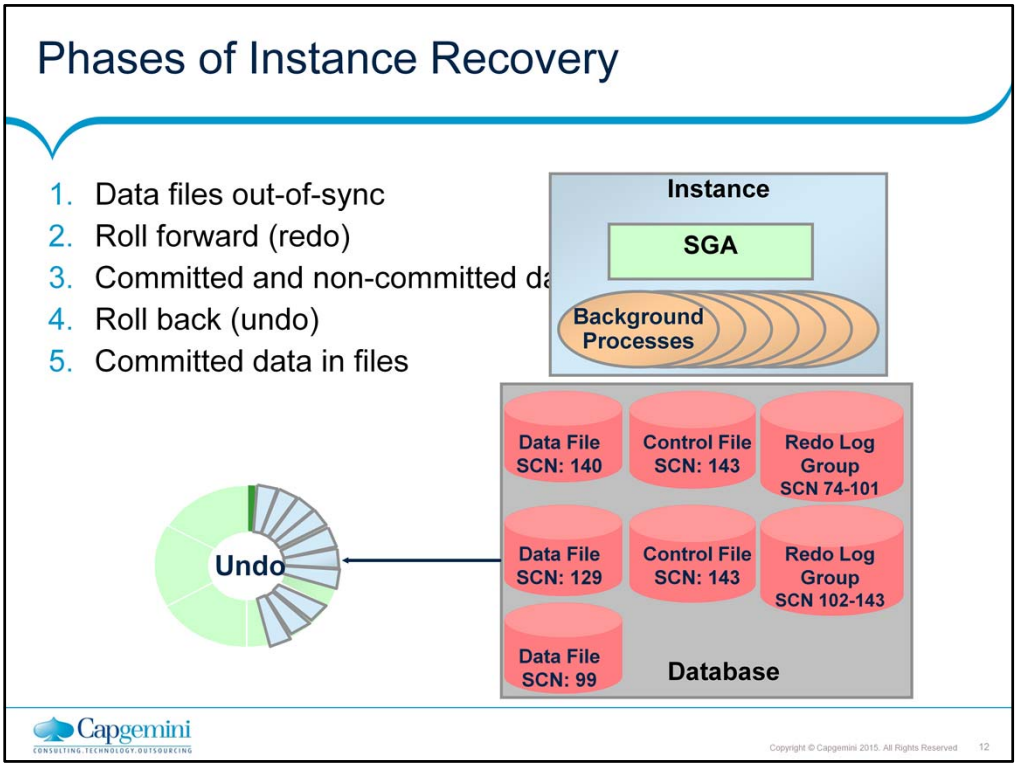

Instance Failure

Instance failure occurs when the database instance is shut down before synchronizing all database files. An instance failure can occur due to hardware or software failure, or through the use of the emergency shutdown commands SHUTDOWN ABORT and STARTUP FORCE. Administrator involvement in recovering from instance failure is usually limited to restarting the instance and working to prevent future occurrences.




Instance Recovery

The Oracle Database 11g automatically recovers from instance failure. All the DBA needs to do is start the instance normally. The instance will mount the control files and then attempt to open the data files. When it discovers the data files have not been synchronized during shutdown, the instance uses information contained in the redo log groups to roll the data files forward to the time of shutdown and then (because the undo tablespace was also rolled forward) roll back any uncommitted transactions.



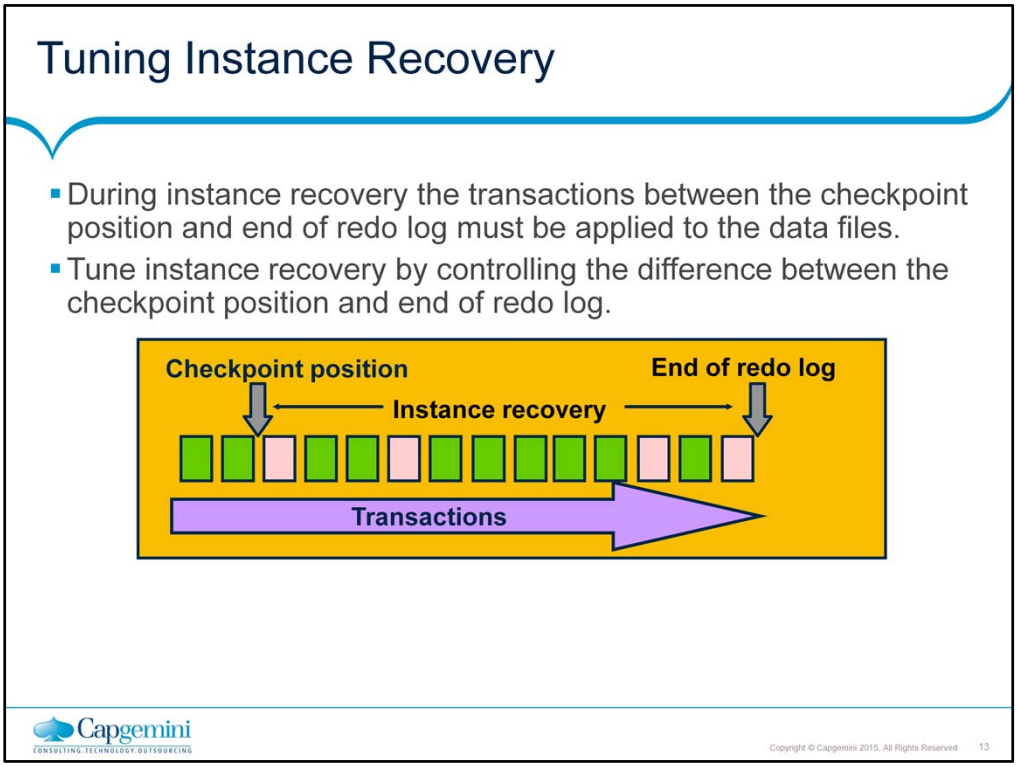

Phases of Instance Recovery

In order for an instance to open a data file, the system change number (SCN) contained within the data file's header must match the current SCN stored in the database's control files.

If the numbers do not match, the instance applies redo from the online redo logs, sequentially "redoing" transactions until the data files are up-to-date.

After all data files have been synchronized with the control files, the database is opened and users may now log in.

When redo was applied, all transactions were applied to bring the database up to the state as of the time of failure. This usually includes transactions that were in progress but had not yet been committed. After the database has been opened, those uncommitted transactions are rolled back. At the end of the rollback phase of instance recovery, the data files will contain only committed data.



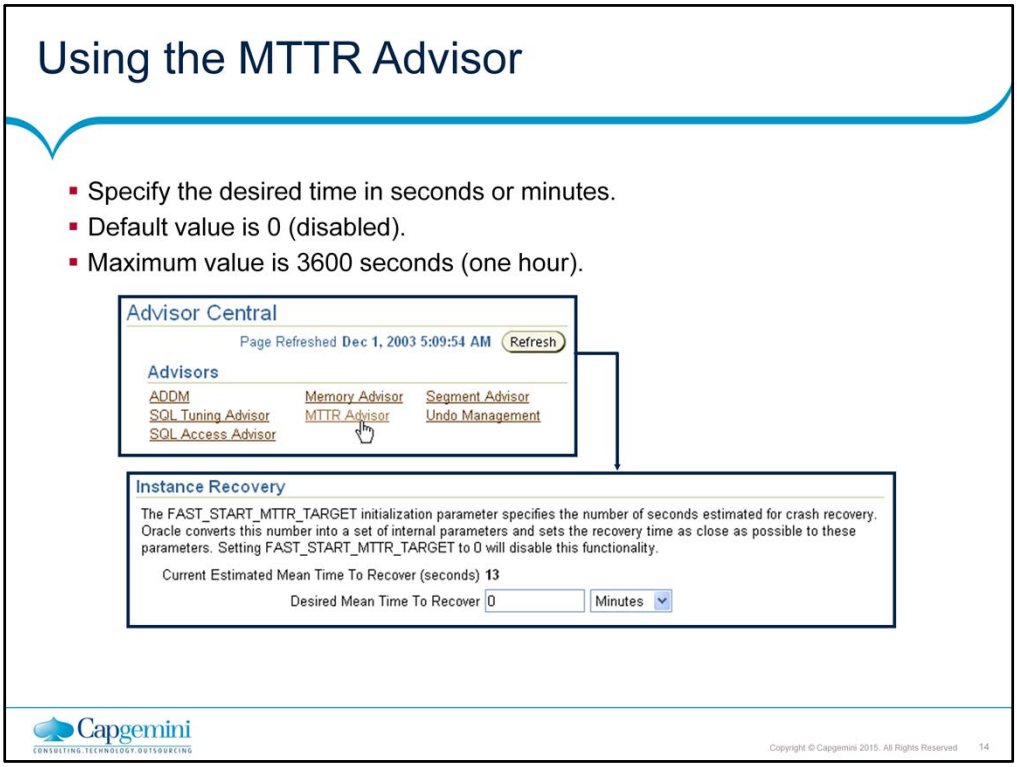

Tuning Instance Recovery

Transaction information is always recorded in the redo log groups before the instance returns commit complete for a transaction. The information in the redo log groups guarantees that the transaction can be recovered in case of a failure. That same transaction information also needs to be written to the data file. The data file write usually happens sometime after the information is recorded in the redo log groups because the data file write process is much slower than the redo writes (random writes for data files are slower than serial writes for redo log files).

To keep track of what has already been written to the data files, the database uses checkpoints. A checkpoint guarantees that as of the time the checkpoint occurs, all data up to a certain SCN is recorded in the data file. Transactions after the checkpoint position may or may not have yet been written to the appropriate data file. In the graphic above, the striped blocks have not yet been written to disk.

The time required for instance recovery is the time required to bring the data files from their last checkpoint to the latest SCN recorded in the control file. The administrator controls that time by setting a MTTR target (in seconds) and through the size of the redo log groups.

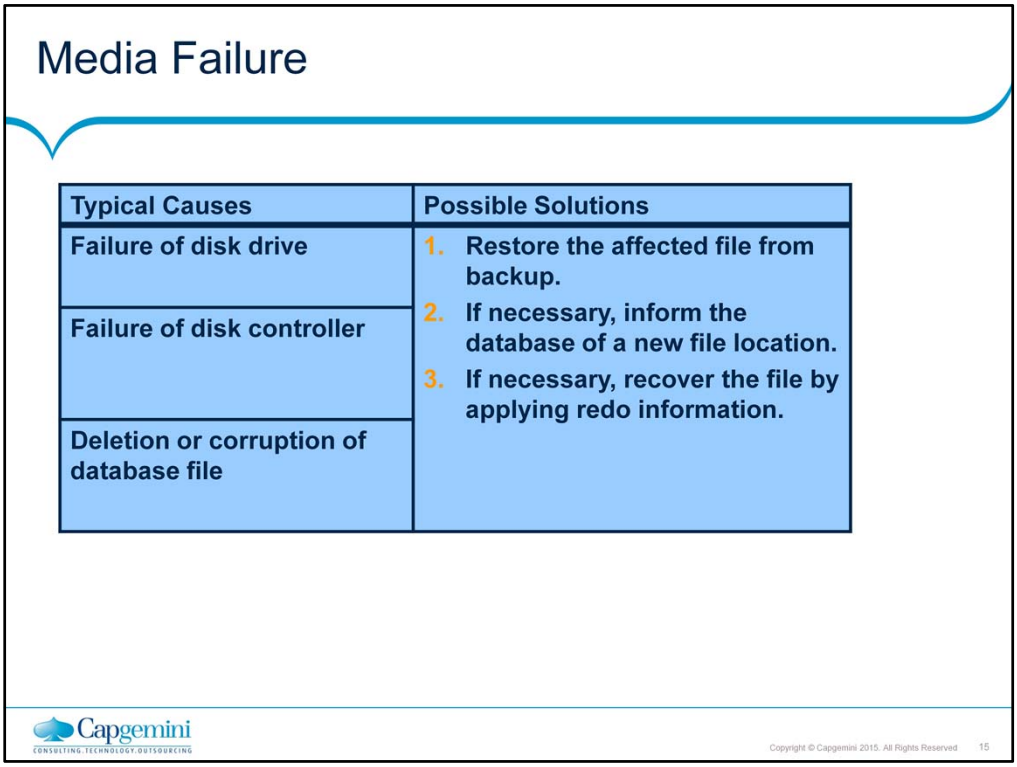
The distance between the checkpoint position and the end of the redo log group can never be more than 90% of the smallest redo log group.




MTTR Advisor

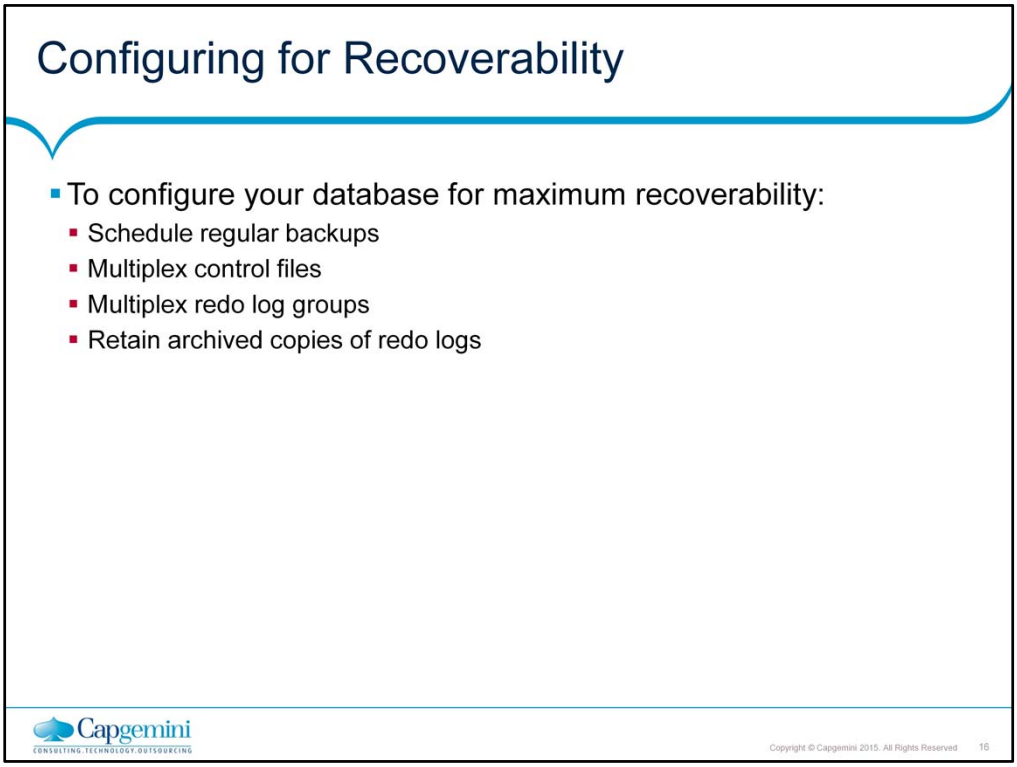
Click the MTTR Advisor from Enterprise Manager's Advisor Center for assistance in setting the MTTR target. The advisor allows you to specify a desired mean-time-to-recover and translates that into settings for the FAST_START_MTTR_TARGET initialization parameter.

The default setting of 0 disables the MTTR target, reducing the likelihood that writes to the log groups will wait on writes to the data files. This should be set to a value that supports the service level agreement for your system. Setting the MTTR target too small means that writes to the log groups wait for writes to the data files (impacting performance). Setting the MTTR target too large means that the instance takes longer to recover after a crash.




Media Failure

Oracle defines media failure as any failure which results in the loss or corruption of one or more database files (data, control, or redo log file). Recovering from media failure requires that you restore and recover the missing files. To ensure your database can be recovered from media failure, follow best practices as outlined in the next few pages. Recovery from media failure will be discussed in more detail in a future lesson.




Configuring for Recoverability

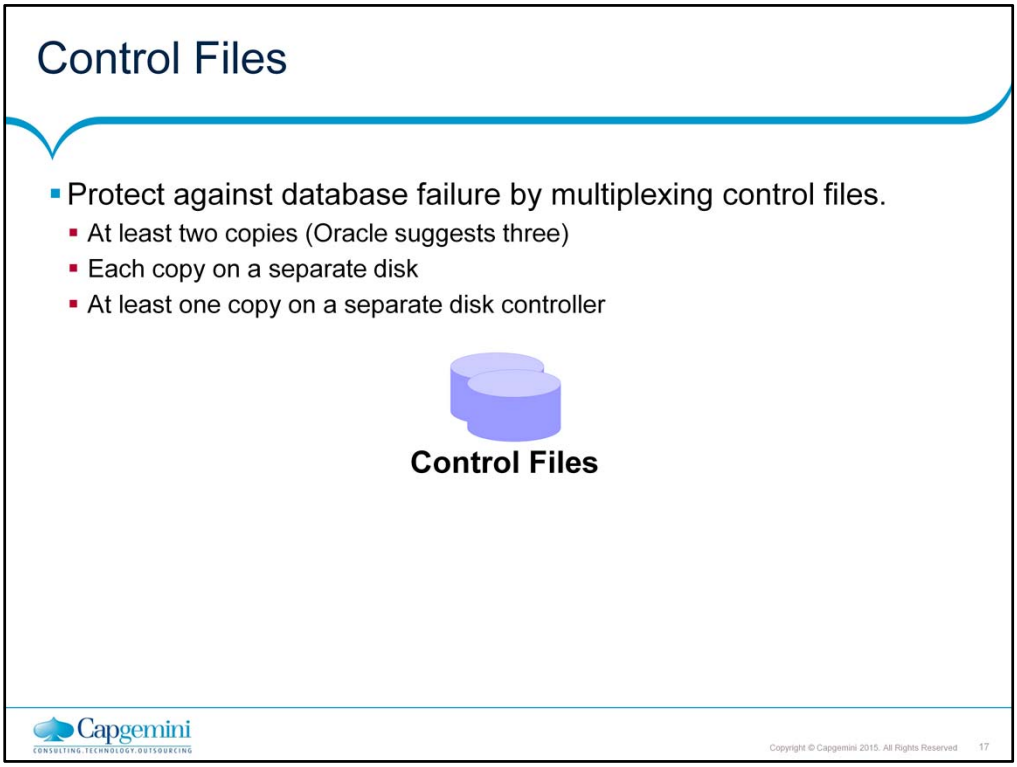
To provide the best protection for your data you should:

Schedule regular backups. Most media failures require that you restore the lost or damaged file from backup.

Multiplex control files. All control files associated with a database are identical. Recovering from the loss of single control file is not difficult. Recovering from the loss of all control files is much more challenging. Guard against losing all control files by having multiple copies (at least three).

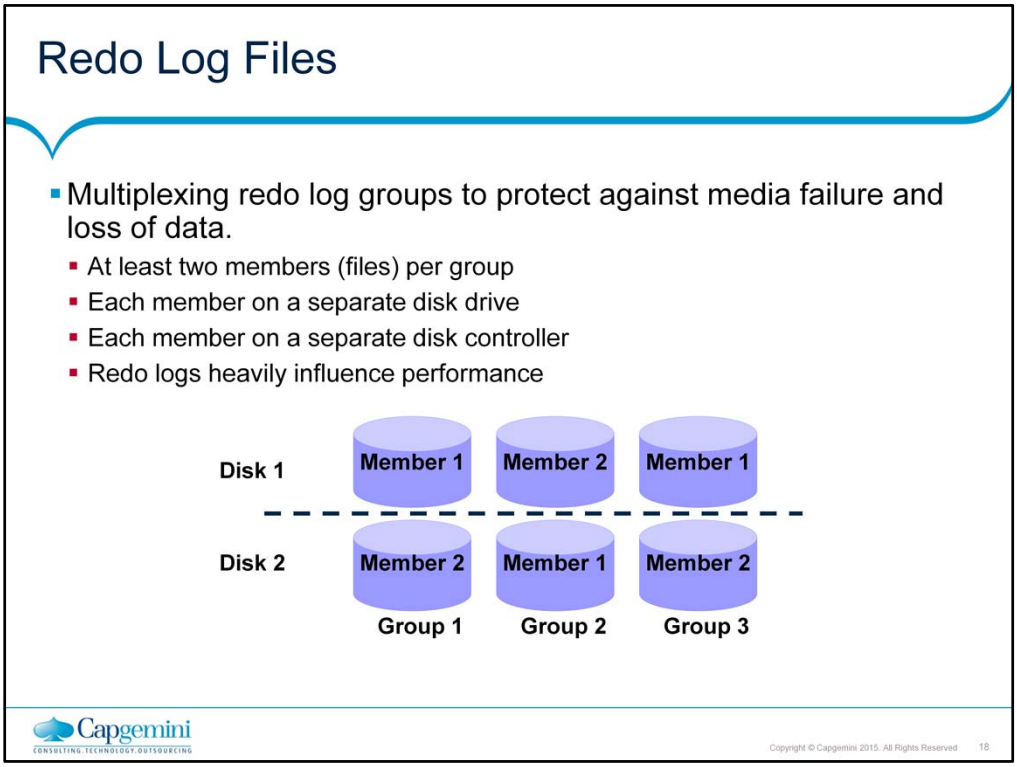
Multiplex redo log groups. To recover from instance or media failure, redo log information is used to roll data files forward to the last committed transaction. If your redo log groups rely on a single redo log file, then the loss of that file means data is likely to be lost. Ensure there are at least two copies of each redo log group.

Retain archived copies of redo logs. If a file is lost and restored from backup, the instance must apply redo information to bring that file up to the latest SCN contained in the control file. The default setting is to overwrite redo information once it has been written to the data files. Your database can be configured to retain redo information in archived copies of the redo logs. This is known as placing the database in ARCHIVELOG mode.




Control Files

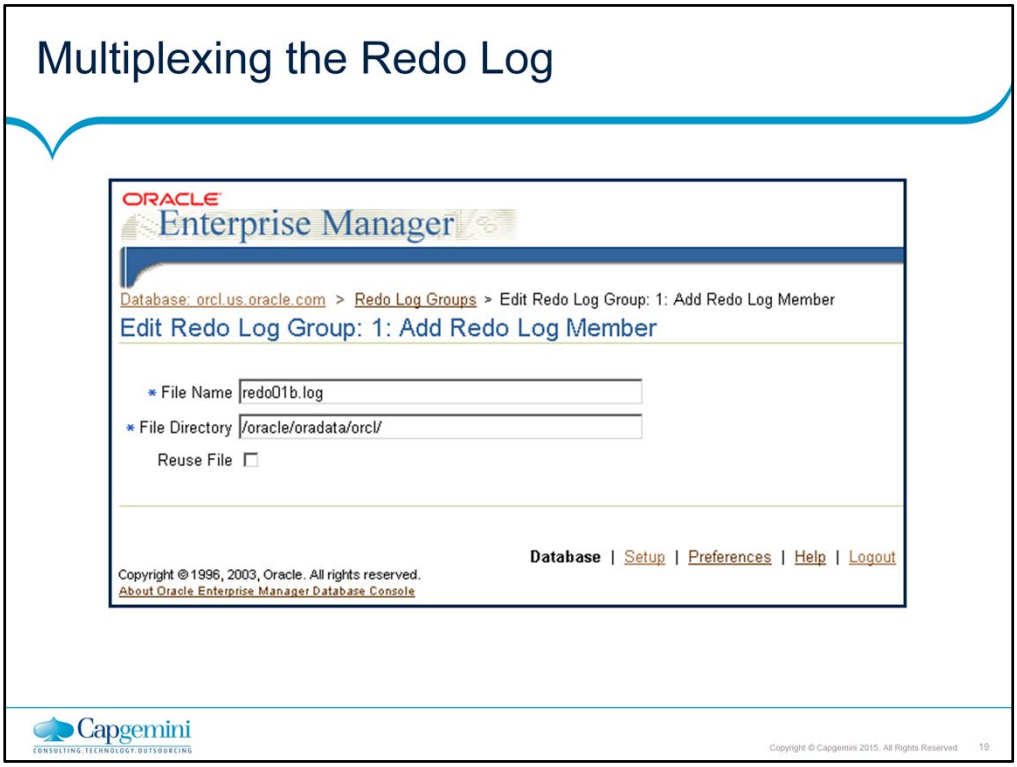
The control file is a small binary file that describes the structure of the database. It must be available for writing by the Oracle server whenever the database is mounted or open. Without this file, the database cannot be mounted and recovery or re-creation of the control file will be required.
Your database should have a minimum of two control files (three is preferred) on different disks to minimize the impact of a loss of one control file.
If your database was created with the DBCA, you should already have three control files.
Loss of a single control file will cause the instance to fail because all control files must be available at all times, but recovery is a simple matter of copying one of the other control files. Loss of all control files is slightly more difficult to recover from, but not usually catastrophic. Recovery from loss of control files will be discussed in a later lesson.




Redo Log Files

Redo log groups are made up one or more redo log files. Each log file within a group is a duplicate of the others. Oracle recommends that redo log groups have at least two files per group, with the files distributed on separate disks/controllers so that no single equipment failure will destroy an entire log group.

Loss of an entire log group is one of the most serious possible media failures because it can result in loss of data. Loss of a single member within a multiple-member log group is trivial, and will not affect database operation other than causing an alert to be published in the alert log. Recovery from loss of a single log file will be discussed in a later lesson. Recovery from loss of an entire log group requires advanced recovery techniques and is discussed in Oracle Database 11g: Workshop II. Remember that redo logs heavily influence database performance because a commit cannot complete until the transaction information has been written to the logs. You should place your redo log files on your fastest disks served by your fastest controllers. If possible, do not place any other database files on the same disks as your redo log files. Because only one group is written to at a given time, there is no harm in having members from several groups on the same disk.



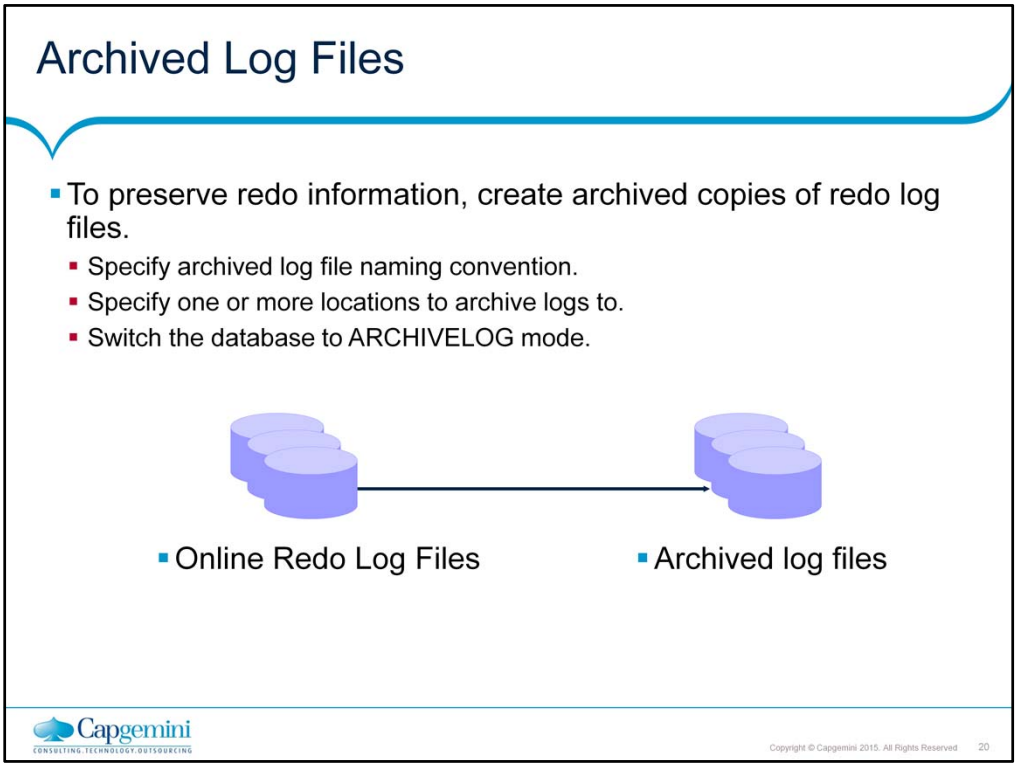

Multiplexing the Redo Log

You can multiplex your redo log by adding a member to an existing log group. Perform the following steps to add a member to a redo log group, this can be done when the database is open with no impact on user performance:

1. Navigate to the Redo Log Groups page.
2. Select a group and click the Edit button, or click the group number link.The Edit Redo Log Group page appears.
3. In the Redo Log Members section, click Add. The Add Redo Log Member page appears.
4. Enter the file name and the file directory. Click OK.
   Note: It is recommended that you store members on separate drives to protect against total loss of the redo log entries in the event of a disk failure.

Repeat these steps for every existing group.

When you add the redo log member to a group, the group's status is marked INVALID. This is the expected state because a member of the group has not yet been written to. When a log switch occurs and the invalid group becomes the current group, the status changes to CURRENT.



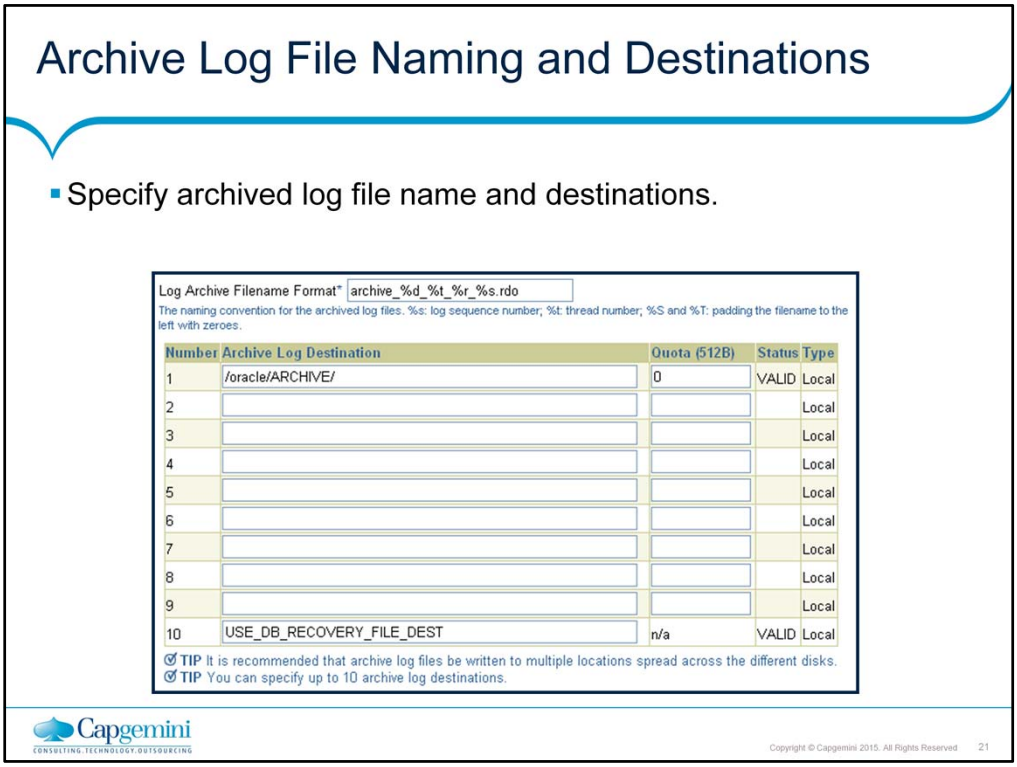

Archived Log Files

The instance treats the online redo log groups as a circular buffer in which to store transaction information, filling one group and then moving on to the next. After all groups have been written to, the instance begins overwriting information in the first log group.

To configure your database for maximum recoverability, you should instruct the database to make a copy of the online redo log group before allowing it to be overwritten. These copies are known as archived logs. To facilitate the creation of archive logs you should:

1. Specify a naming convention for your archived logs.
2. Specify a destination or destinations for storing your archived logs.
3. Place the database in ARCHIVELOG mode.

Note: The destination must exist prior to placing the database in ARCHIVELOG mode. When a directory is specified as a destination, there should be a trailing slash at the end of the directory name.




Archived Log File Naming and Destinations

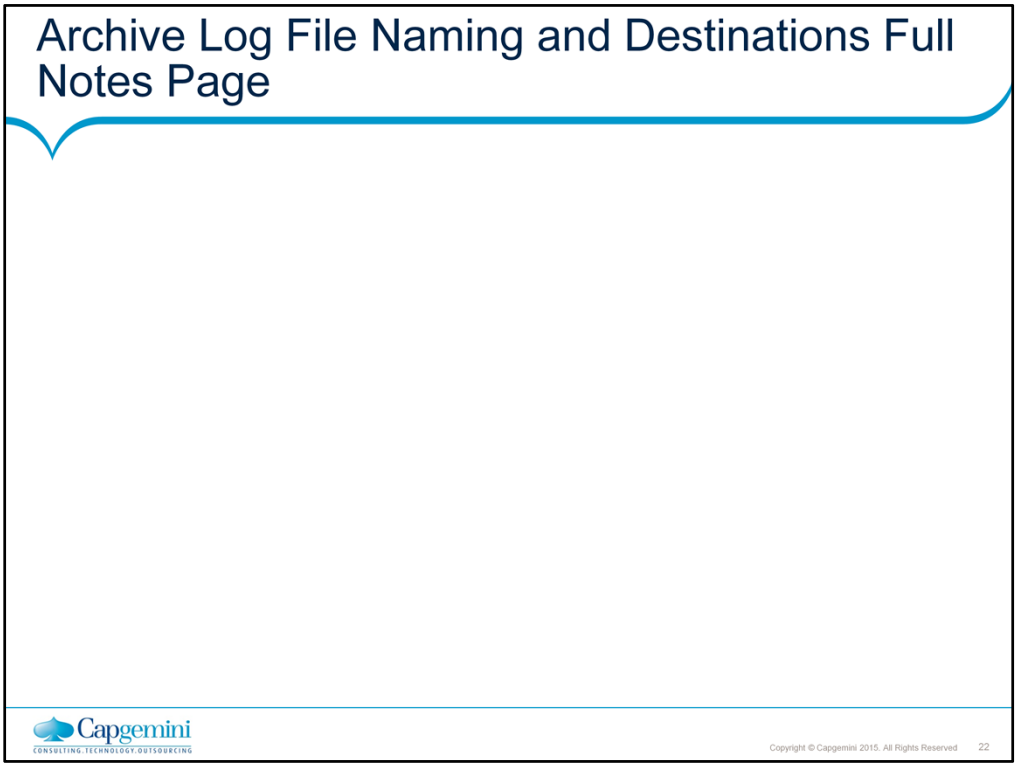
Configure log file naming an destinations by clicking Configure Recovery Settings from the Maintenance page.

Each archive log file must have a unique name to avoid overwriting older log files. You specify the naming format as show above. To help create unique file names, Oracle Database 11g allows several wildcard characters in the name format:

- %s: Includes the log sequence number as part of the file name
- %t: Includes the thread number as part of the file name
- %r: Resetlogs ID. Ensures that the archive log file name remains unique even after certain advanced recovery techniques that reset log sequence numbers
- %d: Includes the database ID as part of the file name

The format must include %s, %t, and %r. The use of %d is optional but it should be included if multiple databases share the same archive log destination.

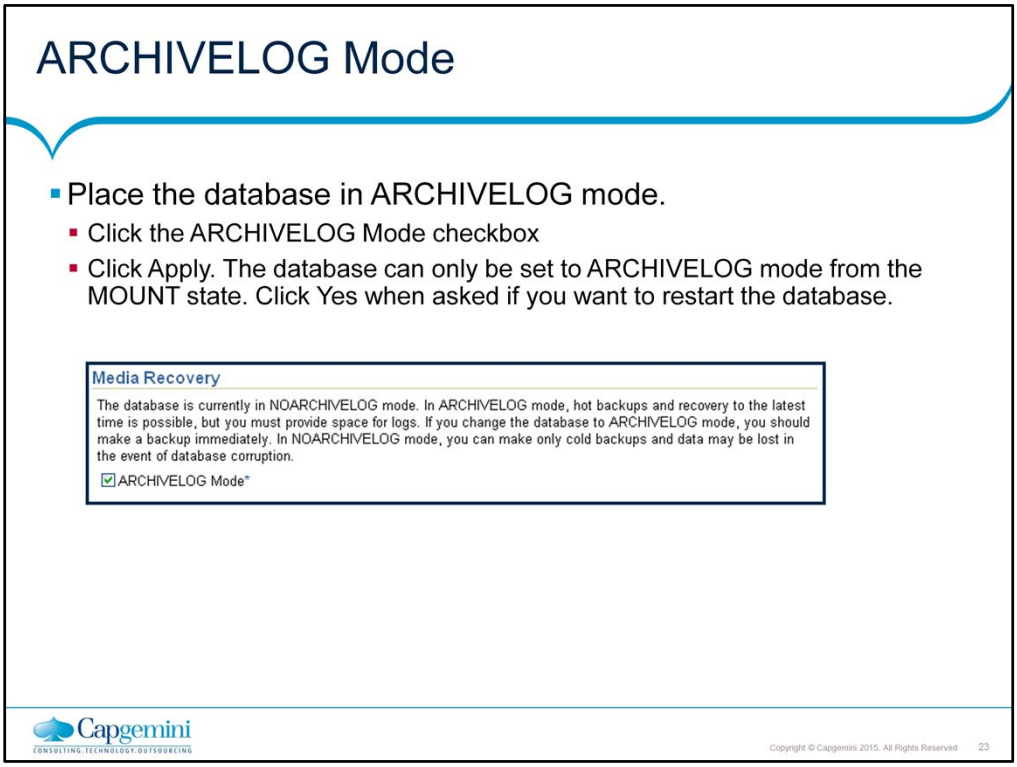
Archived log files can be written to as many as ten different destinations. Destinations may be local (a directory) or remote (an Oracle Net alias for a standby database). Local destinations should end in a slash (/), or a backslash (\) if using Windows.




Archive Log File Naming and Destinations (Continued)

The default destination (number 10) sends archived log files to a location determined by the DB_RECOVERY_FILE_DEST initialization parameter. DB_RECOVERY_FILE_DEST is also known as the flash recovery area. This destination is visible at the bottom of the Configure Recovery Settings properties page as the Flash Recovery Area Location. If you do not want archives sent to this location, simply delete USE_DB_RECOVERY_FILE_DEST.

In order to change recovery settings you must be connected as SYSDBA or SYSOPER.




ARCHIVELOG Mode

Placing the database in ARCHIVELOG mode prevents redo logs from being overwritten until they have been archived, and is the last step in configuring the database for archiving redo information.

The SQL command to place the database in ARCHIVELOG mode is:

```
SQL> ALTER DATABASE ARCHIVELOG;
```

This command can be issued only while the database is in the MOUNT state, so the instance must be restarted to complete this last step. You will be asked for operating system and database credentials during the restart of the database. The database credentials must be for a user with SYSDBA privileges.

After the instance is restarted, the changes you made to the archive processes, log format, and log destinations will take effect.

With the database in NOARCHIVELOG mode (the default), recovery is possible only up until the time of the last backup. All transactions made after that backup are lost.

In ARCHIVELOG mode, recovery is possible up until the time of the last commit. Most production databases are run in ARCHIVELOG mode.

# Summary

- In this lesson you should have learned how to:
  - Describe the basics of database backup, restore and recovery
  - List the types of failure that may occur in an Oracle Database
  - Identify the importance of checkpoints, redo log files, and archived log files
  - Configure ARCHIVELOG mode
  - Describe ways to tune instance recovery


Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

 24