

The number of recovery options provided by Oracle is overwhelming. It's true that there're a lot of ways recovery could be performed, even for a particular failure. Which way to adopt would depend on the type of failure that has occurred, the structures that have been affected, and the type of recovery that's desired

My attempt has been to come up with a "decision-tree" for typical recovery scenarios for Oracle8. (There could be syntactical errors in a few cases if used with Oracle7). It involves a logical flow of thought process that one could adopt, at times, to come out of a "messy" situation. Please note that recovery option for a database is primarily guided by the following:

1... Errors

2... Cost involved in the recovery operation : cost by way of downtime, cost by way of data loss etc.

It should be borne in mind that every recovery situation can be different from the next and that this document, in no way, is an exhaustive guide to Oracle Backup and Recovery.

Thanks a ton to Sandeep Kuttiyattur, Vickje Carbonneau, Greg Lindo for the technical review.

Promod Marar

NOARCHIVELOG MODE

(if redo information is available i.e. if logs have not been overwritten, then we can recover upto the point of failure
Possibility of this exists if a backup was taken recently)

- STARTUP MOUNT**

- List all the online redolog files with seq# and change#

```
SELECT V1.GROUP#, MEMBER, SEQUENCE#, FIRST_CHANGE#
FROM V$LOG V1, V$LOGFILE V2
WHERE V1.GROUP# = V2.GROUP#;
```

- SELECT FILE#, CHANGE# FROM VSRECOVER_FILE;**

- Is CHANGE# from step.3 greater than the lowest FIRST_CHANGE# from step.2 ?

Yes
(recovery is possible upto the pt.of failure)

- RECOVER DATAFILE '<PATH/NAME>';**
- ALTER DATABASE OPEN;**

No
Can you afford to drop the datafile?
(those datafiles belonging to temp or index tablespace can be dropped and recreated..**IF ANY PERMANENT OBJECTS RESIDE IN THIS TABLESPACE, THEY WILL BE LOST.**)

No | **Yes**

What's more recent ?

- Full db backup**
 - Was backup taken with db shutdown clean?
(Check alert log. It should show:
ALTER DATABASE CLOSE NORMAL
ALTER DATABASE DISMOUNT)
- Full export**
 - Recreate database full import

- ALTER DATABASE DATAFILE '<path/name>' OFFLINE DROP;
- ALTER DATABASE OPEN;
- DROP TABLESPACE <NAME> INCLUDING CONTENTS; (YOU'LL LOSE ALL DATA BELONGING TO THIS TABLESPACE)
- Recreate this tablespace

Yes

No

Do you want to restore all files to original location? (if media problem exists then you may want to restore to a different location)

Restore ALL files from the *previous* good backup if available (Refer NOTEA)

Recreate db and import using your good export dump file

IF BACKUP FOR REDO NOT AVAILABLE, REFER NOTE B

STARTUP

Yes

No

- Restore entire database (REFER NOTE A)
(ALL datafiles, controlfile, redologfiles, NOT JUST DAMAGED FILES)

- Restore all files to a new location using cp command (REFER NOTE A: Page 3)

-IF BACKUP FOR REDO IS NOT AVAILABLE, REFER NOTE B(Page 3)

- IF BACKUP FOR REDO IS UNAVAILABLE, REFER NOTE C (Page 3)

- Edit init.ora to show new location of controlfile

- STARTUP

- STARTUP MOUNT

- ALTER DATABASE RENAME FILE '<OLDPATH/NAME>' TO '<NEWPATH/NAME>';

This should be done for every datafile and redolog, which has been taken to a new location

- ALTER DATABASE OPEN;

Still unable to open database?

- **ALTER SESSION SET EVENTS 'IMMEDIATE TRACE NAME FILE_HDRS LEVEL 10';**
- Collect trace file from udump directory
ls -ltr will provide info on the latest trace file
- Check if backup was fuzzy. Refer Note:28985.1

NOTE A:

RESTORE FILES FROM THE BACKUP OF THE PROBLEM DATABASE ONLY. RESTORING FROM A DIFFERENT DATABASE BACKUP (EVEN IF THE DATABASE HAVE THE SAME NAME AND THE SAME DIRECTORY STRUCTURES) WILL NOT WORK, SINCE THEY ARE TWO DIFFERENT ENTITIES.

NOTE B:

IF BACKUP FOR REDOLOGS IS NOT AVAILABLE AND YOU WANT TO RESTORE FILES TO THE ORIGINAL LOCATION:

- Restore all the datafiles and the controlfiles
- **STARTUP MOUNT**
- **ALTER DATABASE BACKUP CONTROLFILE TO TRACE;**
- Collect the trace file from udump directory, which would have been generated for this statement and vi to it. This trace file will have the CREATE CONTROLFILE statement.
- Modify the controlfile statement by changing RESETLOGS to NORESETLOGS
- Keep only the CREATE CONTROLFILE statement...upto the ';'. Everything else prior to the CREATE CONTROLFILE and after the ';'....delete
- Save and exit.....remember the name of this trace file. We'll need to use this file to recreate the controlfile
- **STARTUP NOMOUNT**
- **@<trace file name>**This will recreate the controlfile in all the destination specified in init.ora
- **ALTER DATABASE OPEN RESETLOGS;**
(This will recreate all the redologs in their respective destination...The completion of this statement may take some time depending upon the number of redologs that need to be recreated)

NOTE C:

IF BACKUP FOR REDOLOGS IS NOT AVAILABLE AND YOU WANT TO RESTORE FILES TO A NEW LOCATION:

- Restore the datafiles and the controlfiles to the new location
- Edit init.ora to show the new location of the controlfile
- **STARTUP MOUNT**
- **ALTER DATABASE BACKUP CONTROLFILE TO TRACE;**
- Collect the trace file from udump directory, which would have been generated for this statement and vi to it. This trace file will have the CREATE CONTROLFILE statement.
- Modify the controlfile statement by changing RESETLOGS to NORESETLOGS

- Replace the <oldpath/name> of the datafiles with <newpath/name>
- (MAKE SURE THAT EVERY ONE OF THE NEW LOCATIONS SHOW UP IN THE CONTROLFILE)
- **STARTUP NOMOUNT**
- **@<trace file name>**This will recreate the controlfile
- **ALTER DATABASE OPEN RESETLOGS;**
(This will recreate all the redologs in their respective destination...The completion of this statement may take some time depending upon the number of redologs that need to be recreated)

IF NONE OF THIS WORKS, TRY TO CORRUPT DATABASE AND FORCE IT OPEN USING HIDDEN PARAMETERS. THERE'S NO 100% GUARANTEE THAT BY USING THESE HIDDEN PARAMETERS, THE DATABASE WOULD COME UP. BY USING THESE PARAMETERS, ORACLE SKIPS CERTAIN MANDATORY CHECKS TO OPEN THE DATABASE AND HENCE A VERY HIGH POSSIBILITY OF INCONSISTENCY. PLEASE NOTE THAT ORACLE DOES NOT SUPPORT A DATABASE AFTER USING THESE PARAMETERS FOR RECOVERY. YOU HAVE TO EXPORT THE ENTIRE DATABASE, RECREATE AND IMPORT. AGAIN, THERE'S NO GUARANTEE THAT THE EXPORT WILL SUCCEED.

FORCING THE DATABASE OPEN:

- **SHUTDOWN IMMEDIATE**
 - Backup the entire database. We may have to fall back on this backup if we're unable to force the database open
- A)
- In init.ora, replace ROLLBACK_SEGMENTS by **_CORRUPTED_ROLLBACK_SEGMENTS**
 - Shutdown database
 - **STARTUP MOUNT**
 - **SELECT * FROM V\$DATAFILE** (All datafiles should be online...If not, then **ALTER DATABASE DATAFILE <'fullpath/name'> ONLINE;**)
 - **RECOVER DATABASE UNTIL CANCEL USING BACKUP CONTROLFILE;**
Type "Cancel" when prompted
 - **ALTER DATABASE OPEN RESETLOGS;**

If able to open the database, go to C
- B) *If still unable to open the database,*
- Replace **_CORRUPTED_ROLBACK_SEGMENTS** back to **ROLLBACK_SEGMENTS**
 - Add **_ALLOW_RESETLOGS_CORRUPTION=TRUE**
 - Shutdown database
 - **STARTUP MOUNT**
 - **RECOVER DATABASE UNTIL CANCEL USING BACKUP CONTROLFILE;**
Type "Cancel" when prompted
 - **ALTER DATABASE OPEN RESETLOGS;**

C)

Export the entire database

- Now scratch out the database....remove all the files from the OS level.
- Go to ORACLE_HOME/dbs
- If there're any SGADEF<SID>.DBF and LK<SID> files, remove them. Note: The sgadef<sid>.dbf files will not be seen in 8.1
- Check if there's any stray shared memory and semaphores owned by oracle for this instance.

```
%ipcs -b
```

If they exist, remove them...

```
%ipcrm -m <shared memory id>
```

```
%ipcrm -s <semaphore id>
```

- Recreate the database
Make sure that the _CORRUPTED_ROLLBACK_SEGMENTS no longer appears in init.ora
- Do a full import

ARCHIVELOG MODE LOSS OF SYSTEM DATAFILE

Is backup available?

Yes

No

Was backup taken with database shutdown clean?
(Check alert log. It should show:
ALTER DATABASE CLOSE NORMAL;
ALTER DATABASE DISMOUNT)

Rebuild database from export

If no export, you've lost the database

Yes

No

- Restore only system datafile from backup (*)

Do you have a full export?

- STARTUP MOUNT**

- RECOVER DATAFILE '<PATH/NAME>'**

Yes

- Rebuild database
- Full import

No

*You could try to corrupt and force database open
There's no guarantee that this will work. (Please
refer to the last part of Noarchivelog section :
FORCING THE DATABASE OPEN : Page 4*

Are you being prompted for a non-existent log?

Not sure

Yes

No

- SELECT * FROM VSARCHIVED_LOG;**

- Note the seq# against ORA-280

Media Recovery Complete

**Log exists
No break in seq#**

**Log does not exist
Break in seq#**

- SELECT * FROM v\$log;
- Note the seq# of online redo log

Apply that log

- SHUTDOWN
- Restore ALL datafiles (*)
(NOT Controlfile)
- RECOVER DATABASE UNTIL
CANCEL;
- Type "cancel" when Oracle prompts
for the missing archive log

- If this seq# matches the one against ORA-280,
enter full path/name of one of the members of redo log
Query V\$logFILE to find out the location of redo member
- Keep repeating until you get "Media Recovery Complete"

- ALTER DATABASE OPEN RESETLOGS;**

ALTER DATABASE OPEN;

ALTER DATABASE OPEN;

If still error during open database, dump file header.

SVRMGR> **ALTER SESSION SET EVENTS 'IMMEDIATE TRACE NAME FILE_HDRS LEVEL 10';**

Collect the trace file from the udump directory

Refer to RJOBIN's "Reading File Header Information" Brown bag

(*) NOTE:

RESTORE FILES FROM THE BACKUP OF THE PROBLEM DATABASE ONLY. RESTORING FROM A DIFFERENT DATABASE BACKUP (EVEN IF THE DATABASE HAVE THE SAME NAME AND THE SAME DIRECTORY STRUCTURES) WILL NOT WORK, SINCE THEY ARE TWO DIFFERENT ENTITIES

ARCHIVELOG MODE LOSS OF NON-SYSTEM DATAFILE

Is this a rollback segment datafile?

No

Yes

Is database currently open or shut?

(Refer to Rollback Datafile section Page:)

OPEN

SHUT

STARTUP MOUNT

**SELECT V1.FILE#, NAME FROM VSBACKUP V1,VSDATAFILE V2
WHERE V1.STATUS='ACTIVE' AND V1.FILE#=V2.FILE#;**

If no rows returned

If any rows returned

- **SELECT * FROM VSRECOVER_FILE;**
(Note how many files need recovery)

ALTER DATABASE DATAFILE '<path/name>' END BACKUP;
(for all the datafiles listed against the query)

- **RECOVER DATAFILE '<path/name>';**
(This should be done for every datafile listed in VSRECOVER_FILE)

If database is in mount state,
ALTER DATABASE OPEN;

- Alternatively, If there're many datafiles in RECOVER status,
RECOVER DATABASE

If database is in mount state,
- **ALTER DATABASE OPEN;**

Errors

No errors

Can you afford to drop these files?
(those datafiles belonging to temp or index tablespace can be dropped and recreated. **..IF ANY PERMANENT OBJECTS RESIDE IN THIS TABLESPACE, THEY WILL BE LOST.**)

Problem resolved

No

Yes

Is there a backup of these files?

SHUTDOWN ABORT (if database is currently open)
STARTUP MOUNT
ALTER DATABASE DATAFILE '<path/name>' OFFLINE;
ALTER DATABASE OPEN;
DROP TABLESPACE <name> INCLUDING CONTENTS;
(YOU'LL LOSE ALL DATA BELONGING TO THIS TABLESPACE)
Recreate the tablespace

No

Yes

Was any datafile added AFTER the last backup?

Yes

No

**SELECT D.FILE# F#, D.NAME,D.STATUS,H.STATUS FROM
VSDATAFILE D, VSDATAFILE_HEADER H
WHERE D.FILE#=H.FILE#;**

(X)

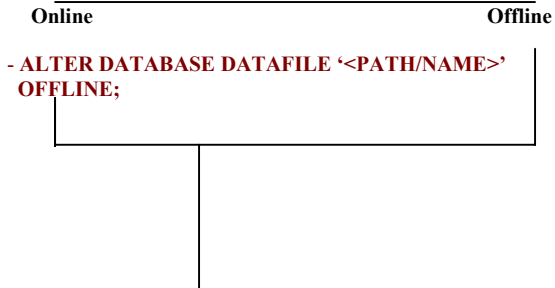
- **SELECT D.FILE# F#, D.NAME,D.STATUS,H.STATUS
FROM VSDATAFILE D, VSDATAFILE_HEADER H
WHERE D.FILE#=H.FILE#;**

(Y)

Refer A (Page 8)

(X)

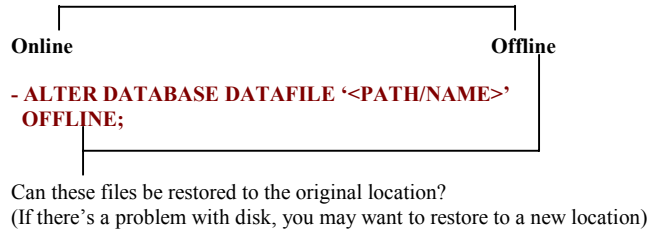
- What does STATUS column show?



- ALTER DATABASE CREATE DATAFILE '<original path/name>';
This will create a datafile of the same size as the previous one since it already is a part of the controlfile
- RECOVER DATAFILE '<path/name>;'
- Apply the logs (Assumes you have all logs needed for recovery)
- ALTER DATABASE DATAFILE '<path/name>' ONLINE;
- ALTER DATABASE OPEN;

(Y)

- What does STATUS column show?

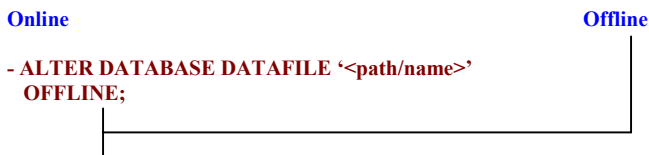


- | | |
|---|--|
| <ul style="list-style-type: none"> - Restore only those files (*) - RECOVER DATAFILE '<path/name>;' - Apply the logs (Assumes you've all logs needed for recovery) - ALTER DATABASE DATAFILE '<path/name>' ONLINE; - ALTER DATABASE OPEN; | <ul style="list-style-type: none"> - Restore files to new location (*) - STARTUP MOUNT - ALTER DATABASE RENAME FILE '<oldpath/name>' TO '<newpath/name>;' - This has to be done for every file taken to new location - RECOVER DATAFILE '<newpath/name>;' - Apply logs (Assumes you've all logs needed for recovery) - ALTER DATABASE DATAFILE '<newpath/name>' ONLINE; - ALTER DATABASE OPEN; |
|---|--|

A: If you've a backup of the missing/damaged files AND you've a backup of the missing / damaged files AND a datafile has been added AFTER the last backup:

```
- SELECT D.FILE# F#, D.NAME,D.STATUS,H.STATUS
FROM VSDATAFILE D, VSDATAFILE_HEADER H
WHERE D.FILE#=H.FILE#;
```

- What does the STATUS column show?



- Can these files be restored to the original location? (If there's a problem with disk, you may want to restore to a new location)

Yes
(X)

No
REFER B (Page 10)

(X)

2 options

*Using current controlfile
if available*

- Restore ONLY the damaged datafiles (*)
(NOT controlfile, redologs)

- STARTUP MOUNT

- ALTER DATABASE CREATE DATAFILE
'<path/name of datafile added since last backup>';

- RECOVER DATABASE;

- Apply the logs

- ALTER DATABASE OPEN;

Using backup controlfile

- Restore damaged datafile
and controlfile (NOT redo) (*)

- STARTUP MOUNT

- RECOVER DATABASE USING
BACKUP CONTROLFILE;

- Apply the logs until redo to be
applied to new datafile (until you
get errors ORA-1244, ORA-1110)

- SELECT NAME FROM V\$DATAFILE
WHERE FILE# = <get the file# from ORA-1110>
(The file name should be something like
UNNAMEDxxx)

- ALTER DATABASE CREATE DATAFILE
'<UNNAMEDxxx>' AS '<originalpath/name
of the datafile added since the last backup>'

- ALTER DATABASE RENAME DATAFILE
'<UNNAMEDxxx>' TO '<originalpath/name
of the datafile added since the last backup>';

- RECOVER DATABASE USING BACKUP
CONTROLFILE;

- Apply the logs

- Are you prompted for a non-existent log?

Not sure

Yes

No

Query V\$ARCHIVED_LOG to find out
if the log exists. If it exists, apply that log

- Note the seq# against ORA-280

Media Recovery Complete

- SELECT * FROM v\$log;
Note the seq# of online redolog

- If this seq# matches the one against ORA-280,
enter full path/name of one of the members of redolog
Query V\$logfile to find out the location of redo member

- Keep repeating until you get "Media Recovery Complete"

- ALTER DATABASE OPEN RESETLOGS;

B: If a datafile has been added since the last backup and you don't want to restore the damaged files to the original location:

————— 2 options —————

*Using current controlfile
(if available)*

- Restore only the damaged datafiles to new location (NOT controlfile, redo) (*)

- STARTUP MOUNT

- ALTER DATABASE RENAME FILE
'<oldpath/name>' TO '<newpath/name>';

- ALTER DATABASE CREATE DATAFILE
'<path/name of the datafile added since the last backup>';

OR – if this has to go a new location

ALTER DATABASE CREATE DATAFILE
'oldpath/name>' AS '<newpath/name>'

- RECOVER DATABASE

- Apply the logs

- ALTER DATABASE OPEN;

Using backup controlfile

- Restore the damaged datafile and controlfile to new location (NOT redo) (*)

- Edit CONTROL_FILES in init.ora to show the new location of controlfile

- STARTUP MOUNT

- RECOVER DATABASE USING BACKUP CONTROLFILE;

- Apply logs until redo to be applied to new datafile (You'll get errors ORA-1244, ORA-1110)

- SELECT NAME FROM VSDATAFILE WHERE
FILE# = (get the file# from ORA-1110)
(You should get a file named something like UNNAMEDxxx)

- ALTER DATABASE CREATE DATAFILE
'<UNNAMEDxxx>' AS 'path/name of datafile added since last backup';

-ALTER DATABASE RENAME DATAFILE
'<UNNAMEDxxx>' TO '<path/name of datafile added since last backup>';

- Apply the logs.

- Are you prompted for a non-existent log?

Not sure

|

Query V\$ARCHIVED_LOG to find out if the log exists. If it exists, apply that log

Yes

|

- Note the seq# against ORA-280

- SELECT * FROM v\$log;
Note the seq# of online redolog

- If this seq# matches the one against ORA-280, enter full path/name of one of the members of redolog
Query V\$logfile to find out the location of redo member

- Keep repeating until you get "Media Recovery Complete"

- ALTER DATABASE OPEN RESETLOGS;

No

|

Media Recovery Complete

(*) NOTE:

RESTORE FILES FROM THE BACKUP OF THE PROBLEM DATABASE ONLY. RESTORING FROM A DIFFERENT DATABASE BACKUP (EVEN IF THE DATABASE HAVE THE SAME NAME AND THE SAME DIRECTORY STRUCTURES) WILL NOT WORK, SINCE THEY ARE TWO DIFFERENT ENTITIES

ARCHIVELOG MODE REDOLOG LOSS

Are redolog files there at the os level?

Yes

No
(REFER E)
Page 13

Is it just a missing/damaged member of the group?
(Check alert log for any abnormalities If not sure, refer C first and then come back to this point if unsuccessful))

No

Yes
(Refer D)
Page 13

SELECT * FROM VSLOG;
(Check STATUS of group listed against the error.

Current
(REFER A)
Page 11

Inactive
(REFER B)
Page 12

Active
(REFER C)
Page 12

A: IF STATUS IS CURRENT:

Is database currently open or shut?

Open

Shut

SELECT * FROM VSLOG;
Note which group is current

**ALTER DATABASE CLEAR
UNARCHIVED LOGFILE
GROUP <current group #)**

SHUTDOWN IMMEDIATE

Backup immediately , since recovery information has just been cleared

- STARTUP MOUNT
- SELECT * FROM VSLOG;
Check which group is current and note current log sequence no.

- Restore ALL datafiles and redolog files from backup (*)

- RECOVER DATABASE UNTIL CANCEL;
- When it prompts for current log sequence# , type "Cancel"

- ALTER DATABASE OPEN RESETLOGS;

- SHUTDOWN IMMEDIATE

- Backup the entire database

B: IF STATUS IS INACTIVE:

What does ARC column from VSLOG show?

Yes

This group is already archived
Hence no recovery information is lost

**ALTER DATABASE CLEAR LOGFILE
GROUP <GROUP#>**

No

**- SHUTDOWN ABORT
- STARTUP MOUNT
- SELECT * FROM V\$DATAFILE;**
Is there any datafile OFFLINE?

Yes

ALTER DATABASE DATAFILE '<path/name> ONLINE;
If this datafile is looking for information from the online log, it'll
Give error when bringing the file online.

Has the datafile come online?

Yes

**- ALTER DATABASE CLEAR UNARCHIVED
LOGFILE GROUP <GROUP#>
UNRECOVERABLE DATAFILE;**

NOTE: The redo necessary to bring the datafile
is being cleared, so one would have to drop this
datafile and the associated tablespace

- ALTER DATABASE OPEN;

**- DROP TABLESPACE <NAME> INCLUDING
CONTENTS; PLEASE NOTE THAT ALL
DATA BELONGING TO THIS TABLESPACE
WILL BE LOST**

- Recreate this tablespace

No

**- ALTER DATABASE CLEAR UNARCHIVED
LOGFILE GROUP <GROUP#>**

- ARCHIVE LOG ALL

- If there're only 2 log groups, then add a group
and drop the above damaged group.

**-ALTER DATABASE ADD LOGFILE GROUP <#>
'<path/name>' SIZE <.> (You may want to query VSLOG to note
the size of other log groups)**

**- ALTER DATABASE DROP LOGFILE
GROUP <damaged GROUP#>;**

- ALTER SYSTEM SWITCH LOGFILE;
(Do this a couple of times)

C: IF STATUS IS ACTIVE:

- ALTER SYSTEM CHECKPOINT;

-SELECT * FROM VSLOG;

STATUS: INACTIVE?

Problem taken care of

STATUS != INACTIVE OR DATABASE HUNG??

- SHUTDOWN ABORT

- Restore full database from last backup (*)

- STARTUP MOUNT

- RECOVER DATABASE;

- Apply the logs (Assumes you've all the logs needed for recovery)

- ALTER DATABASE OPEN RESETLOGS;

- SHUTDOWN IMMEDIATE

- Take a full backup

D: IF ITS JUST A MISSING/DAMAGED MEMBER OF THE GROUP:

(Check alert log for any abnormalities)

- SELECT * FROM V\$LOGFILE;

Is STATUS shown as INVALID for any of the logfile members?

Yes

No

(This may not be a case of damaged/missing member)

ALTER SYSTEM SWITCH LOGFILE;

Is STATUS of that logfile member still shown as INVALID?

Yes

No

(this may not be a case of damaged/missing member)

ALTER SYSTEM SWITCH LOGFILE;

Now drop this member and add it again:

ALTER DATABASE DROP LOGFILE MEMBER '<path/name>;'

ALTER DATABASE ADD LOGFILE MEMBER '<path/name>' TO GROUP <#>;

ALTER SYSTEM SWITCH LOGFILE;

E: IF THE CURRENT REDOLOGS ARE NOT THERE AT THE OS LEVEL:

Corrupt the database and force it open, using hidden parameters.

THERE'S NO 100% GUARANTEE THAT BY USING THESE HIDDEN PARAMETERS, THE DATABASE WOULD COME UP. BY USING THESE PARAMETERS, ORACLE SKIPS CERTAIN MANDATORY CHECKS TO OPEN THE DATABASE AND HENCE A VERY HIGH POSSIBILITY OF INCONSISTENCY. PLEASE NOTE THAT ORACLE DOES NOT SUPPORT A DATABASE AFTER USING THESE PARAMETERS FOR RECOVERY. YOU HAVE TO EXPORT THE ENTIRE DATABASE, RECREATE AND IMPORT. AGAIN, THERE'S NO GUARANTEE THAT THE EXPORT WILL SUCCEED.

• **SHUTDOWN IMMEDIATE**

- Take a full backup. We may have to fall back on this backup, if we're unable to force the database open

- In init.ora, add:

```
_ALLOW_RESETLOGS_CORRUPTION=TRUE
_CORRUPTED_ROLLBACK_SEGMENTS=<string of rollback segments>
Comment out the ROLLBACK_SEGMENTS parameter
```

• **STARTUP MOUNT**

• **SELECT * FROM V\$DATAFILE**

(All datafiles should be online... If not, then ALTER DATABASE DATAFILE <'fullpath/name'> ONLINE;

• **RECOVER DATABASE UNTIL CANCEL USING BACKUP CONTROLFILE;**

When prompted for a log, type "Cancel"

• **ALTER DATABASE OPEN RESETLOGS;**

- Export the entire database

- Now scratch out the database....remove all the files from the OS level.

- Go to ORACLE_HOME/bin

- If there're any SGADEF<SID>.DBF and LK<SID> files, remove them

- Check if there's any stray shared memory and semaphores owned by oracle for this instance. If they exist, remove them

- Recreate the database

- Do a full import

- Make sure that the _CORRUPTED_ROLLBACK_SEGMENTS no longer appears in init.ora

(*) NOTE:

RESTORE FILES FROM THE BACKUP OF THE PROBLEM DATABASE ONLY. RESTORING FROM A DIFFERENT DATABASE BACKUP (EVEN IF THE DATABASE HAVE THE SAME NAME AND THE SAME DIRECTORY STURCTURES) WILL NOT WORK, SINCE THEY ARE TWO DIFFERENT ENTITIES

ARCHIVELOG MODE LOSS OF ROLLBACK DATAFILE

Is database currently open or shut?

SHUT

OPEN

Was it a clean shutdown?
(Alert log should show:
ALTER DATABASE CLOSE NORMAL
ALTER DATABASE DISMOUNT)

DO NOT SHUTDOWN DB
(Refer to A: Page 16)

Yes

No

- Comment out ROLLBACK_SEGMENTS in init.ora

- Restore only the lost file (*)

- STARTUP MOUNT RESTRICT

- STARTUP MOUNT

- ALTER DATABASE DATAFILE <lost rbs datafile> OFFLINE;
- Query V\$DATAFILE to confirm that the datafile is offlined.

- SELECT FILE#, NAME, STATUS FROM V\$DATAFILE;

- ALTER DATABASE OPEN;

Online

Offline

Errors (ORA-604, ORA-376)

Stmt processed

- SHUTDOWN IMMEDIATE

- Add _CORRUPTED_ROLLBACK_SEGMENTS = <string of rbs> in init.ora
(There's a high chance of corrupting the dictionary in this case)

- STARTUP RESTRICT

**ALTER DATABASE DATAFILE
'<PATH/NAME>' ONLINE;**

- RECOVER DATAFILE '<PATH/NAME>'

- DROP TABLESPACE <name> INCLUDING CONTENTS

- Recreate the tablespace

- Recreate the rollback segments

- Apply the archivelogs.

Are you being prompted for a non-existent log?

Not sure

Yes

No

- Remove _CORRUPTED_ROLLBACK_SEGMENTS
from init.ora
- Uncomment ROLLBACK_SEGMENTS from init.ora

Query V\$ARCHIVED_LOG
to find out if the log exists. If it
exists, then apply the log

- Note the seq# against ORA-280 Media Recovery Comple

- ALTER SYSTEM DISABLE RESTRICTED SESSION

- SELECT * FROM VSLOG;
Note the seq# of online redolog

- If this seq# matches the one against
ORA-280, enter the full path/name of
one of the members of redolog
Query V\$LOGFILE to find out the
location of the redo member

- Keep repeating until you get "Media
Recovery Complete"

- ALTER DATABASE OPEN ;

A: DATABASE IS CURRENTLY OPEN:

- ALTER DATABASE DATAFILE '<PATH/NAME>' OFFLINE;

- Restore only that datafile from backup (*)

- RECOVER DATAFILE '<PATH/NAME>'

- Apply the archive logs

- Are you being prompted for a non-existent archivelog?

Not sure

Query V\$ARCHIVED_LOG to find out if the log exists. If it exists, apply that log

Yes

- Note the seq# against ORA-280

- SELECT * FROM v\$log;
Note the seq# of online redolog

- If this seq# matches the one against ORA-280, enter full path/name of one of the members of redolog
Query V\$logfile to find out the location of redo member

- Keep repeating until you get "Media Recovery Complete"

No

Media Recovery Complete

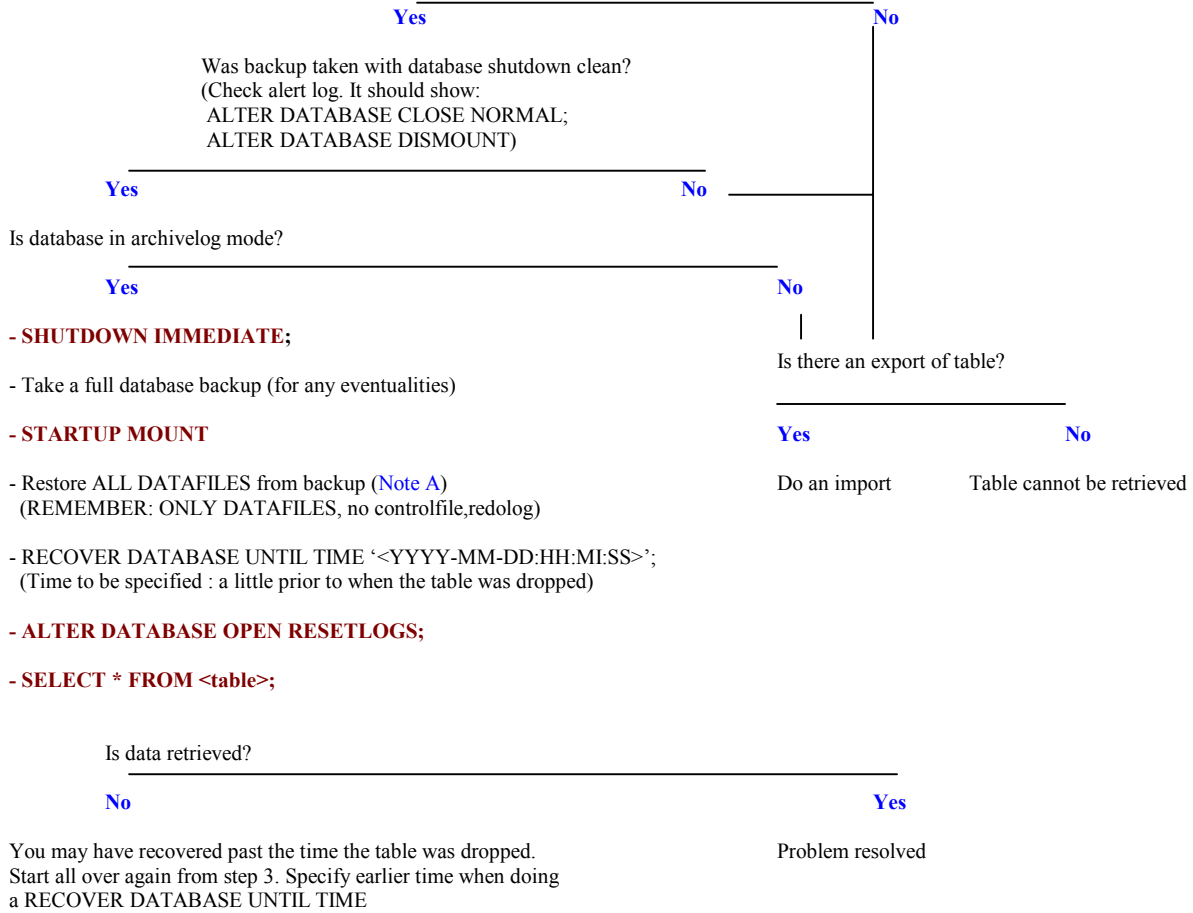
- ALTER DATABASE DATAFILE '<PATH/NAME>' ONLINE;

(*) NOTE:

RESTORE FILES FROM THE BACKUP OF THE PROBLEM DATABASE ONLY. RESTORING FROM A DIFFERENT DATABASE BACKUP (EVEN IF THE DATABASE HAVE THE SAME NAME AND THE SAME DIRECTORY STRUCTURES) WILL NOT WORK, SINCE THEY ARE TWO DIFFERENT ENTITIES

TABLE DROPPED/TRUNCATED

Do you have a good backup?



NOTE A:
RESTORE FILES FROM THE BACKUP OF THE PROBLEM DATABASE ONLY. RESTORING FROM A DIFFERENT DATABASE BACKUP (EVEN IF THE DATABASE HAVE THE SAME NAME AND THE SAME DIRECTORY STRUCTURES) WILL NOT WORK, SINCE THEY ARE TWO DIFFERENT ENTITIES

**ARCHIVELOG MODE
DISK CRASH
LOSS OF CONTROLFILE, REDOLOG, DATAFILES**

Has the disk been rectified?

No

Yes

- Restore ALL the files from coldbackup for the damaged disk to a new location (REFER A: Page 18)

- IF BACKUP OF REDOLOGS NOT AVAILABLE, REFER B (Page 19)

- Edit CONTROL_FILES in init.ora to show the new location of the controlfile

- SVRMGR> CONNECT INTERNAL

- SHUTDOWN ABORT

- STARTUP MOUNT

- Rename all the datafiles that were moved to new location: and open the database

ALTER DATABASE RENAME FILE '<oldpath/name>' TO '<newpath/name>';

RECOVER DATAFILE '<newpath/name>';
Apply the archivelogs (assumes you've all the archives needed for recovery)

ALTER DATABASE OPEN;

-Then drop and recreate logfile members of non-current group:

SELECT A.GROUP#, A.MEMBER, A.STATUS MEMBER_STATUS, B.STATUS GROUP_STATUS, B.BYTES FROM VSLOGFILE A, VSLOG B WHERE A.GROUP# = B.GROUP#;

For GROUP_STATUS not showing CURRENT:

**ALTER DATABASE DROP LOGFILE MEMBER '<oldpath/name>';
ALTER DATABASE ADD LOGFILE MEMBER '<newpath/name>' TO GROUP <#>;**

- ALTER SYSTEM SWITCH LOGFILE;

- Repeat above step if there's any other member that's been moved to a new location and is showing as INVALID in VSLOGFILE

- SHUTDOWN IMMEDIATE

- Backup

NOTE A:

RESTORE FILES FROM THE BACKUP OF THE PROBLEM DATABASE ONLY. RESTORING FROM A DIFFERENT DATABASE BACKUP (EVEN IF THE DATABASE HAVE THE SAME NAME AND THE SAME DIRECTORY STRUCTURES) WILL NOT WORK, SINCE THEY ARE TWO DIFFERENT ENTITIES

NOTE:B

IF BACKUP OF REDOLOGS NOT AVAILABLE:

- Restore all the datafiles and the controlfiles

- STARTUP MOUNT

- ALTER DATABASE BACKUP CONTROLFILE TO TRACE;

- Collect the trace file from udump directory, which would have been generated for this statement and vi to it. This trace file will have the CREATE CONTROLFILE statement.

- Modify the controlfile statement by changing RESETLOGS to NORESETLOGS

- Keep only the CREATE CONTROLFILE statement...upto the ';' . Everything else prior to the CREATE CONTROLFILE and after the ';'delete

- Save and exit.....remember the name of this trace file. We'll need to use this file to recreate the controlfile

- STARTUP NOMOUNT

- **@<trace file name>** This will recreate the controlfile in the same destination as specified in init.ora

- ALTER DATABASE OPEN RESETLOGS;

(This will recreate all the redologs in their respective destination...The completion of this statement may take some time depending upon the number of redologs that need to be recreated)

-Then drop and recreate logfile members of non-current group:

**- SELECT A.GROUP#, A.MEMBER, A.STATUS MEMBER STATUS,
B.STATUS GROUP_STATUS, B.BYTES FROM V\$LOGFILE A,
V\$LOG B WHERE A.GROUP# = B.GROUP#;**

- For GROUP_STATUS not showing CURRENT:

**ALTER DATABASE DROP LOGFILE MEMBER '<oldpath/name>';
ALTER DATABASE ADD LOGFILE MEMBER '<newpath/name>' TO GROUP <#>;**

- ALTER SYSTEM SWITCH LOGFILE;

- Repeat above step if there's any other member that's been moved to a new location and is showing as INVALID in V\$LOGFILE

- SHUTDOWN IMMEDIATE

- Backup

ARCHIVELOG MODE DATABASE CRASHED WHILE IN HOTBACKUP MODE

STARTUP MOUNT
SELECT V1.FILE#, NAME FROM V\$BACKUP V1, V\$DATAFILE V2
WHERE V1.STATUS='ACTIVE' AND V1.FILE#=V2.FILE#;

If any rows returned

If no rows returned

ALTER DATABASE DATAFILE '<path/name>' END BACKUP;
 (for every datafile listed against the query)

(Refer to the section from “no rows returned”
 Section -Loss of non-system datafile)

ALTER DATABASE OPEN;

If any errors ,

- Restore all the files from previous backup (*)

- **SELECT * FROM V\$BACKUP;**

- If STATUS is shown as ACTIVE for any files,

ALTER DATABASE DATAFILE '<path/name>' END BACKUP;

- **RECOVER DATABASE UNTIL CANCEL USING BACKUP CONTROLFILE;**

- Apply the logs.

Are you prompted for a non-existent log?

Not sure

Yes

No

SELECT * FROM VSARCHIVED_LOG;

- Note the seq# against ORA-280

Media Recovery Complete

Log exists
No break in seq#

Log does not exist
Break in seq#

- **SELECT * FROM v\$log;**
 Note the seq# of online redolog

Apply that log

- **SHUTDOWN**
 - Restore ALL datafiles
 (NOT Controlfile) (*)
 - **RECOVER DATABASE UNTIL**
CANCEL;
 - Type “cancel” when Oracle prompts
 for the missing archive log

- If this seq# matches the one against ORA-280,
 enter full path/name of one of the members of redolog
 Query V\$LOGFILE to find out the location of redo member
 - Keep repeating until you get “Media Recovery Complete”

- ALTER DATABASE OPEN RESETLOGS;

ALTER DATABASE OPEN;

ALTER DATABASE OPEN;

***PLEASE NOTE THAT IF YOU TRY TO “RECOVER DATABASE UNTIL TIME...”
 A POINT-IN-TIME RECOVERY HAS TO GO PAST THE END-BACKUP MARKER.....SO YOU COULD RECOVER UNTIL TIME AFTER THE END
 BACKUP IS ISSUED. NOT UNTIL TIME THE FILES ARE STILL IN THE BACKUP MODE.***

If still error during open, then dump the file header

SVRMGR> **ALTER SESSION SET EVENTS 'IMMEDIATE TRACE NAME FILE_HDRS LEVEL 10';**

Collect the trace file from udump directory

Refer to RJOBIN's “Reading File header Dumps” brown bag.

(*) NOTE:

**RESTORE FILES FROM THE BACKUP OF THE PROBLEM DATABASE ONLY. RESTORING FROM A DIFFERENT DATABASE
 BACKUP (EVEN IF THE DATABASE HAVE THE SAME NAME AND THE SAME DIRECTORY STURCTURES) WILL NOT WORK,
 SINCE THEY ARE TWO DIFFERENT ENTITIES**