# 



Atos for internal use

Oracle\_DB\_Recovery\_Datafile\_Deletion

author(s) : Kabilan Tamilvanan

document number : INW-DBA-0007

version : 1.0

status : Final

source : Atos

document date : 10 Nov 2022

number of pages : 7

| Role | Name | Signature | Date |
| --- | --- | --- | --- |
| Reviewer 1 | Bala Muthusami | B.M | 10 Nov 2022 |
| Quality Assurance Function | Santosh Kudtarkar | S.K. | 8 Nov 2022 |
| Document Owner | Kabilan Tamilvanan | K.T | 8 Nov 2022 |

© Copyright 2019, Atos Global IT Solutions & Services Private Limited All rights reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner. For any questions or remarks on this document, please contact Atos, .

­­­Contents

[1 Purpose 4](#_Toc118991868)

[2 Scope 5](#_Toc118991869)

[3 Recovering database while datafiles were deleted from db filesytems 6](#_Toc118991870)

[4 HOW TO RECOVER DELETED ORACLE DATAFILES WITH NO DOWNTIME 8](#_Toc118991871)

List of changes

| version | Date | Description | Author(s) |
| --- | --- | --- | --- |
| 1.0 | 08.11.2022 | Initial | Kabilan Tamilvanan |
| 1.0 | 10.11.2022 | Review | Bala Muthusami |

­­­­

# Purpose

This document is prepared to recover database in case of database files deleted at OS level physically by DBA

# Scope

The scope of the document is limited to recovering database in the case of database files deleted from db filesystems mistakenly by DBA

# Recovering database while datafiles were deleted from db filesytems

**Recovering the primary database's datafile using the physical standby, and vice versa**

**GOAL**

How to recover the primary database's datafile using a copy of a standby database's datafile.

and

How to recover the standby database's datafile using a copy of a primary database's datafile.

**Important**

Before replacing a datafile with a copy from either production or standby, please confirm that all archivelogs are available for full recovery of this datafile. It is also important to ensure that the source is corruption-free. Run DBV or RMAN validate to check for corruption.

a) dbv must return with zero corrupted pages

$ dbv file=<full path filename> blocksize=<tablespace blocksize> logfile=<output log>

 If using ASM, you need to also supply the userid for dbv:

$ dbv userid=system/<password> file=<full path filename> blocksize=<tablespace blocksize> logfile=<output log>

b) rman validate:

RMAN> backup validate check logical datafile n;

Once RMAN is completed, this view must return zero rows:

SQL> select \* from v$database\_block\_corruption;

**SOLUTION**

These procedure will work for all file systems - cooked, raw or ASM.

Throughout this example we will be using datafile 5.

**Recovering the Primary's Datafile**

1) in the standby database, backup the datafile to a cooked file system:

RMAN> backup datafile 5 format '/tmp/df5\_st.bk';

2) transfer the backup piece from the standby to the primary host using scp, ftp, nfs etc

3) in the primary database, do the following:

a) catalog this backup piece and confirm that it is available for use:

RMAN> catalog backuppiece '/tmp/df5\_st.bk';  
RMAN> list backuppiece '/tmp/df5\_st.bk'  
RMAN> list backup of datafile 5;

b) restore the datafile:

SQL> alter database datafile 5 offline;  
  
Now make an operating system copy of the primary datafile before overlaying with the restored copy as a precaution  
  
RMAN> restore datafile 5;

c) recover the datafile:

RMAN> recover datafile 5;

d) place the datafile online:

SQL> alter database datafile 5 online;

R**ecovering the Standby's Datafile**

When recovering the standby, reverse the steps.

1) at the primary site, take a backup of the datafile:

RMAN> backup datafile 5 format '/tmp/df5\_pr.bk' tag 'PRIMARY\_5';

 2) transfer the file to the standby site using an operating system utility such as scp, NFS, ftp etc

3) at the standby site, catalog the backuppiece and confirm it's available for use:

RMAN> catalog backuppiece'/tmp/df5\_pr.bk';  
RMAN> list backuppiece'/tmp/df5\_pr.bk';  
RMAN> list backup of datafile 5;

 4) stop redo apply on the physical standby database. For an active dataguard you will need to restart the standby database in MOUNT mode first before stopping managed recovery.

SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;

 5) on the standby site restore the datafile:

RMAN> restore datafile 5;

 6) restart redo apply on the physical standby database. For an active dataguard you can go ahead and restart the active dataguard process.

SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT;

NOTE:  prior to 12.1, for real time apply, you must use:

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

# HOW TO RECOVER DELETED ORACLE DATAFILES WITH NO DOWNTIME

**Phase 1:** instant recovery to prevent Oracle errors (like ORA-27041 “unable to open file”, ORA-01110, ORA-01116)

1. Find the PID of DBWRITER for the right database.

ps -ef | grep dbw0\_SID\_OF\_YOUR\_DB

oracle 12272 1 0 10:55 ? *00*:*00*:*00* ora\_dbw0\_test

oracle 12941 11501 0 12:36 pts/0 *00*:*00*:*00* grep dbw0\_test

1. List the deleted file handles for that DBWRITER process.

ls -l /proc/\_PID\_OF\_DBWRITER/fd | grep deleted

lrwx------ 1 oracle oinstall 64 Oct 15 11:24 10 -> /home/oracle/product/10.2.0.2/dbs/lkinsttest (deleted)

lrwx------ 1 oracle oinstall 64 Oct 15 11:24 23 -> /ra5a/orabkp/test/TEST/datafile/o1\_mf\_lost\_3k6xzjpm\_.dbf (deleted)

1. Create a symbolic link to your datafile with the original name.

ln -s /proc/PID\_OF\_DBWRITER/fd/23 /ra5a/orabkp/test/TEST/datafile/o1\_mf\_lost\_3k6xzjpm\_.dbf

Now you are no longer going to get errors. However, if your database goes down now, you will lose that datafile for good.

**Phase 2:** restore the file from backup

ARCHIVELOG database

1. (Optional.) Issue a checkpoint. This is to reduce the recovery time when bringing the file online, depending on activity for that datafile. Unfortunately, you can’t checkpoint a single file, so the checkpoint may take some time.

alter system checkpoint;

1. Backup the datafile with rman. Why rman? It’s much easier then you think. Total downtime is about one second for inactive datafiles, and more for active ones (with writes).

rman target /

report schema;

backup as copy datafile YOUR\_DATAFILE\_NUMBER format '/location\_of\_your\_database/new\_name\_for\_File.dbf';

sql 'alter database datafile YOUR\_DATAFILE\_NUMBER offline';

switch datafile YOUR\_DATAFILE\_NUMBER to copy;

restore datafile YOUR\_DATAFILE\_NUMBER;

recover datafile YOUR\_DATAFILE\_NUMBER;

sql 'alter database datafile YOUR\_DATAFILE\_NUMBER online';

exit;