

Backup & Recovery Check List

[Alejandro Vargas](#)
Oracle Support Israel.

User Managed Recovery Scenarios And Configuration

1. [Complete Closed Database Recovery. System tablespace is missing](#)
2. [Complete Open Database Recovery. Non system tablespace is missing](#)
3. [Complete Open Database Recovery \(when the database is initially closed\). Non system tablespace is missing](#)
4. [Recovery of a Missing Datafile that has no backups.](#)
5. [Restore and Recovery of a Datafile to a different location.](#)
6. [Control File Recovery](#)
7. [Incomplete Recovery, Until Time/Sequence/Cancel](#)

Rman Recovery Scenarios And Configuration

1. [Complete Closed Database Recovery. System tablespace is missing](#)
2. [Complete Open Database Recovery. Non system tablespace is missing](#)
3. [Complete Open Database Recovery \(when the database is initially closed\). Non system tablespace is missing](#)
4. [Recovery of a Datafile that has no backups.](#)
5. [Restore and Recovery of a Datafile to a different location.](#)
6. [Control File Recovery](#)
7. [Incomplete Recovery, Until Time/Sequence/Cancel](#)

User Managed Recovery Scenarios

User managed recovery scenarios do require that the database is in archive log mode, and that backups of all datafiles and control files are made with the tablespaces set to begin backup, if the database is open while the copy is made. At the end of the copy of each tablespace it is necessary to take it out of backup mode. Alternatively complete backups can be made with the database shutdown. Online redologs can be optionally backed up.

Files to be copied:

```
select name from v$datafile;
```

```
select member from v$logfile; # optional
select name from v$controlfile;
```

[go up](#)

Complete Closed Database Recovery. System tablespace is missing

If the system tablespace is missing or corrupted the database cannot be started up so a complete closed database recovery must be performed.

Pre requisites: A closed or open database backup and archived logs.

1. Use OS commands to restore the missing or corrupted system datafile to its original location, ie:
cp -p /user/backup/uman/system01.dbf
/user/oradata/u01/dbtst/system01.dbf
2. startup mount;
3. recover datafile 1;
4. alter database open;

[go up](#)

Complete Open Database Recovery. Non system tablespace is missing

If a non system tablespace is missing or corrupted while the database is open, recovery can be performed while the database remain open.

Pre requisites: A closed or open database backup and archived logs.

1. Use OS commands to restore the missing or corrupted datafile to its original location, ie:
cp -p /user/backup/uman/user01.dbf /user/oradata/u01/dbtst/user01.dbf
2. alter tablespace <tablespace_name> offline immediate;
3. recover tablespace <tablespace_name>;
4. alter tablespace <tablespace_name> online;

[go up](#)

Complete Open Database Recovery (when the database is initially closed). Non system tablespace is missing

If a non system tablespace is missing or corrupted and the database crashed, recovery can be performed after the database is open.

Pre requisites: A closed or open database backup and archived logs.

1. startup; (you will get ora-1157 ora-1110 and the name of the missing datafile, the database will remain mounted)
2. Use OS commands to restore the missing or corrupted datafile to its original location, ie:
`cp -p /user/backup/uman/user01.dbf /user/oradata/u01/dbtst/user01.dbf`
3. alter database datafile3 offline; (tablespace cannot be used because the database is not open)
4. alter database open;
5. recover datafile 3;
6. alter tablespace <tablespace_name> online;

[go up](#)

Recovery of a Missing Datafile that has no backups (database is open).

If a non system datafile that was not backed up since the last backup is missing, recovery can be performed if all archived logs since the creation of the missing datafile exist.

Pre requisites: All relevant archived logs.

1. alter tablespace <tablespace_name> offline immediate;
2. alter database create datafile '/user/oradata/u01/dbtst/newdata01.dbf';
3. recover tablespace <tablespace_name>;
4. alter tablespace <tablespace_name> online;

If the create datafile command needs to be executed to place the datafile on a location different than the original use:

```
alter database create datafile '/user/oradata/u01/dbtst/newdata01.dbf' as  
'/user/oradata/u02/dbtst/newdata01.dbf'
```

[go up](#)

Restore and Recovery of a Datafile to a different location.

If a non system datafile is missing and its original location not available, restore can be made to a different location and recovery performed.

Pre requisites: All relevant archived logs.

1. Use OS commands to restore the missing or corrupted datafile to the new location, ie:
`cp -p /user/backup/uman/user01.dbf /user/oradata/u02/dbtst/user01.dbf`
2. `alter tablespace <tablespace_name> offline immediate;`
3. `alter tablespace <tablespace_name> rename datafile
'/user/oradata/u01/dbtst/user01.dbf' to '/user/oradata/u02/dbtst/user01.dbf';`
4. `recover tablespace <tablespace_name>;`
5. `alter tablespace <tablespace_name> online;`

[go up](#)

Control File Recovery

Always multiplex your controlfiles. Controlfiles are missing, database crash.

Pre requisites: A backup of your controlfile and all relevant archived logs.

1. `startup;` (you get ora-205, missing controlfile, instance start but database is not mounted)
2. Use OS commands to restore the missing controlfile to its original location:
`cp -p /user/backup/uman/control01.dbf /user/oradata/u01/dbtst/control01.dbf`
`cp -p /user/backup/uman/control02.dbf /user/oradata/u01/dbtst/control02.dbf`
3. `alter database mount;`
4. `recover automatic database using backup controlfile;`
5. `alter database open resetlogs;`
6. make a new complete backup, as the database is open in a new incarnation and previous archived log are not relevant.

[go up](#)

Incomplete Recovery, Until Time/Sequence/Cancel

Incomplete recovery may be necessary when an archived log is missing, so recovery can only be made until the previous sequence, or when an important object was dropped, and recovery needs to be made until before the object was dropped.

Pre requisites: A closed or open database backup and archived logs, the time or sequence that the 'until' recovery needs to be performed.

1. If the database is open, shutdown abort
2. Use OS commands to restore all datafiles to its original locations:
`cp -p /user/backup/uman/u01/*.dbf /user/oradata/u01/dbtst/`
`cp -p /user/backup/uman/u02/*.dbf /user/oradata/u01/dbtst/`
`cp -p /user/backup/uman/u03/*.dbf /user/oradata/u01/dbtst/`
`cp -p /user/backup/uman/u04/*.dbf /user/oradata/u01/dbtst/`
etc...
3. startup mount;
4. recover automatic database until time '2004-03-31:14:40:45';
5. alter database open resetlogs;
6. make a new complete backup, as the database is open in a new incarnation and previous archived log are not relevant.

Alternatively you may use instead of until time, until sequence or until cancel:

```
recover automatic database until sequence 120 thread 1; OR
recover database until cancel;
```

[go up](#)

Rman Recovery Scenarios

Rman recovery scenarios require that the database is in archive log mode, and that backups of datafiles, control files and archived redolog files are made using Rman. Incremental Rman backups may be used also. Rman can be used with the repository installed on the archivelog, or with a recovery catalog that may be installed in the same or other database.

Configuration and operation recommendations:

Set the parameter `controlfile autobackup` to ON to have with each backup a controlfile backup also:

configure controlfile autobackup on;

Set the parameter retention policy to the recovery window you want to have, ie redundancy 2 will keep the last two backups available, after executing delete obsolete commands:

configure retention policy to redundancy 2;

Execute your full backups with the option '*plus archivelogs*' to include your archivelogs with every backup:

backup database plus archivelog;

Perform daily maintenance routines to maintain on your backup directory the number of backups you need only:

crosscheck backup;

crosscheck archivelog all;

delete noprompt obsolete backup;

To work with Rman and a database based catalog follow these steps:

1. `sqlplus /`
2. `create tablespace repcat;`
3. `create user rcuser identified by rcuser default tablespace repcat temporary tablespace temp;`
4. `grant connect, resource, recovery_catalog_owner to rcuser`
5. `exit`
6. `rman catalog rcuser/rcuser` `# connect to rman catalog as the rcuser`
7. `create catalog` `# create the catalog`
8. `connect target /` `#`

[go up](#)

Complete Closed Database Recovery. System tablespace is missing

In this case complete recovery is performed, only the system tablespace is missing, so the database can be opened without resetting the redologs.

1. `rman target /`
2. `startup mount;`
3. `restore database;`
4. `recover database;`
5. `alter database open;`

[go up](#)

Complete Open Database Recovery. Non system tablespace is missing, database is up

1. rman target /
2. sql 'alter tablespace <tablespace_name> offline immediate';
3. restore datafile 3;
4. recover datafile 3;
5. sql 'alter tablespace <tablespace_name> online';

[go up](#)

Complete Open Database Recovery (when the database is initially closed). Non system tablespace is missing

A user datafile is reported missing when tryin to startup the database. The datafile can be turned offline and the database started up. Restore and recovery are performed using Rman. After recovery is performed the datafile can be turned online again.

1. sqlplus /nolog
2. connect / as sysdba
3. startup mount
4. alter database datafile '<datafile_name>' offline;
5. alter database open;
6. exit;
7. rman target /
8. restore datafile '<datafile_name>';
9. recover datafile '<datafile_name>';
10. sql 'alter tablespace <tablespace_name> online';

[go up](#)

Recovery of a Datafile that has no backups (database is up).

If a non system datafile that was not backed up since the last backup is missing, recovery can be performed if all archived logs since the creation of the missing datafile exist. Since the database is up you can check the tablespace

name and put it offline. The option *offline immediate* is used to avoid that the update of the datafile header.

Pre requisites: All relevant archived logs.

1. sqlplus '/ as sysdba'
2. alter tablespace <tablespace_name> offline immediate;
3. alter database create datafile '/user/oradata/u01/dbtst/newdata01.dbf';
4. exit
5. rman target /
6. recover tablespace <tablespace_name>;
7. sql 'alter tablespace <tablespace_name> online';

If the create datafile command needs to be executed to place the datafile on a location different than the original use:

```
alter database create datafile '/user/oradata/u01/dbtst/newdata01.dbf' as  
'/user/oradata/u02/dbtst/newdata01.dbf'
```

[go up](#)

Restore and Recovery of a Datafile to a different location. Database is up.

If a non system datafile is missing and its original location not available, restore can be made to a different location and recovery performed.

Pre requisites: All relevant archived logs, complete cold or hot backup.

1. Use OS commands to restore the missing or corrupted datafile to the new location, ie:
cp -p /user/backup/uman/user01.dbf /user/oradata/u02/dbtst/user01.dbf
2. alter tablespace <tablespace_name> offline immediate;
3. alter tablespace <tablespace_name> rename datafile
'/user/oradata/u01/dbtst/user01.dbf' to '/user/oradata/u02/dbtst/user01.dbf';
4. rman target /
5. recover tablespace <tablespace_name>;
6. sql 'alter tablespace <tablespace_name> online';

[go up](#)

Control File Recovery

Always multiplex your controlfiles. If you loose only one controlfile you can replace it with the one you have in place, and startup the Database. If both controlfiles are missing, the database will crash.
Pre requisites: A backup of your controlfile and all relevant archived logs. When using Rman always set configuration parameter autobackup of controlfile to ON. You will need the dbid to restore the controlfile, get it from the name of the backed up controlfile.
It is the number following the 'c-' at the start of the name.

1. rman target /
2. set dbid <dbid#>
3. startup nomount;
4. restore controlfile from autobackup;
5. alter database mount;
6. recover database;
7. alter database open resetlogs;
8. make a new complete backup, as the database is open in a new incarnation and previous archived log are not relevant.

[go up](#)

Incomplete Recovery, Until Time/Sequence/Cancel

Incomplete recovery may be necessaire when the database crash and needs to be recovered, and in the recovery process you find that an archived log is missing. In this case recovery can only be made until the sequence before the one that is missing.

Another scenario for incomplete recovery occurs when an important object was dropped or incorrect data was committed on it.

In this case recovery needs to be performed until before the object was dropped.

Pre requisites: A full closed or open database backup and archived logs, the time or sequence that the 'until' recovery needs to be performed.

1. If the database is open, shutdown it to perform full restore.
2. rman target \
3. startup mount;
4. restore database;
5. recover database until sequence 8 thread 1; # you must pass the thread, if a single instance will always be 1.

6. alter database open resetlogs;
7. make a new complete backup, as the database is open in a new incarnation and previous archived log are not relevant.

Alternatively you may use instead of until sequence, until time, ie: '2004-12-28:01:01:10'.

[go up](#)