# **17 Database Refreshing**

**Refresh**:

* Refresh is nothing but a copy of data from one database server to another database server.
* It may be a partial copy or full database copy.
* Copy the latest production data to a non-production environment (e.g., development, testing, or staging).

**Types of Refresh**:

1. table refresh
2. Schema refresh
3. Full database refresh

**Traditional exp / imp or datapump**:

**Table refresh using exp / imp**

**exp**

* Take the table backup
* Copy dumpfile to target server using **scp table\_t1.dmp oracle@10.100.24.51:/SSD**
* exp system/manager file=table\_t1.dmp log=table\_t1.log tables=u1.t1

**imp**

* imp \"/as sysdba\" file=dumpfile.dmp log=imp\_t1.log fromuser=u1 touser=u1 tables=t1

**Schema refresh using exp / imp**

**exp**

* Take the schema backup.
* Copy dumpfile to target server.
* exp system/manager file=schema\_u1.dmp log=u1.log owner=u1

**imp**

* imp system/system file=dumpfile.dmp log=imp\_schema.log fromuser=u1 touser=u2

**Table refresh using** **expdp / impdp:**

**expdp**

* expdb system/system directory=expdp dumpfile=u1\_emp.dmp logfile=emp.log tables=u1.emp

**impdp (**remap\_schem , rempa\_table**)**

* impdp system/manager directory=expdp dumpfile=u1.dmp logfile=u1.log tables=u1.emp **remap\_table**=emp:emp\_copy
* impdp system/system directory=impdp dumpfile=u1.dmp logfile=emp.log tables=u1.emp **remap\_schema**=u1:mouli **remap\_table**=emp:emp\_copy
* impdp \"/as sysdba\" directory=IMPDPJP DUMPFILE=2025.dmp LOGFILE=tab.txt tables=JMSUATJP.t\_pr\_ot\_management\_det,JMSUATJP.t\_pr\_ot\_management\_seq

**Schema refresh using expdp / impdp:**

**expdp**

* expdp system/system directory=expdp dumpfile=u1.dmp logfile=u1.log schemas=u1

**impdp** (remap\_schema)

* In target schema no objects should be present.
* impdp system/system directory=impdp dumpfile=u1.dmp logfile=emp.log schemas=u1
* impdp system/system directory= impdp dumpfile=u1.dmp logfile=u2.log schemas=u1 **remap\_schema**=u1:u2

**Table\_exists\_action:**

* If the importing table is already present in the database and we didn’t pass **table\_exists\_action** then by-default import will skip the table.

1. Replace
2. Append
3. Truncate

**Replace:**

* It will drop the existing table and recreate table.
* table\_exists\_action=replace

**Append:**

* loads rows from the source and leaves existing rows unchanged.
* In this if the table has primary key it will fail.
* table\_exists\_action=append

**Truncate:**

* TRUNCATE deletes existing rows from table and then loads rows from the source.
* table\_exists\_action=truncate

**Full Database Refresh:**

* In full db refresh source and target must be same version.
* Source and target must have same operating system.
* Source db name and target db name is different
* Copy pfile from source to target in every full db refresh.

**Conventional Cold Refresh:**

**Source db:**

1. check **C R D** files location.
2. shut down database
3. Take cold backup & copy to target server
4. Copy pfile to target server
5. Put db in mount state take controlfile trace

* Alter database backup controlfile to trace as ‘/SSD/testdb/ctl\_01’;

**Target db:**

1. Create required directories.
2. Set env.
3. Edit the pfile (mv inithyd.ora inithydtst.ora)

* In pfile change db name and files location.

1. Put db in nomount state
2. Edit controlfile trace and if db name is different Recreate the controlfile with resetlogs.
3. Open the database with resetlogs

* Alter database open resetlogs;

**Conventional Hot Backup:**

* copy pfile from source to target in every full db refresh.

**source db:**

* directory structure /prod/hyd/oradata

1. db must be enable with archivelog mode
2. Check the max sequence
3. Put the db in begin backup mode
4. Take backup of only datafiles and copy to target server
5. End the begin backup mode
6. do some log switches
7. Check max sequence
8. Once again check the archives count and copy to the backup location
9. Copy pfile to target server

**Target db:**

* directory structure/prod/hydtst/oradata

1. Create required directories
2. Set env file.
3. Edit pfile
4. Put db in nomount state
5. Edit controlfile trace and if db name is different Recreate the controlfile with resetlogs.

* Don’t use **reuse** optine while creating controlfile. if don’t have controlfile.

1. Recover database using backup controlfile until cancel;

* This will apply archives.

1. Alter database open resetlogs;

**RMAN Database Refresh:**

1. Rman cold backup
2. Rman hot backup
3. Incremental backup
4. Auxiliary duplicate method (backup is required)
5. active duplicate method (backup is not required)

**Rman Cold Database Refresh:**

**Source side:**

1. Put the db in mount state.
2. Connect to Rman.

* RMAN > backup database tag=coldbkp;
* RMAN > backup current controlfile;(tag)

1. open the database (startup)
2. Copy backup to target server
3. Copy pfile

**Target server:**

1. create required directories
2. Set env
3. Edit pfile

* Initially db name is same as prod db name (hyd) change remaining names

1. put the db in nomount state
2. Restore the controlfile. Rman**> restore controlfile from '/data/rman/UATDB\_3.rman';**
3. Put db in mount
4. Register backup in target database. Rman**> catalog start with ‘backup file location’;**
5. Restore database

* When the db datafiles locations are different we have to use this script.

run

{

set newname for datafile 1 to '/data/testdb/oradata/system01.dbf';

set newname for datafile 2 to '/data/testdb/oradata/sysaux01.dbf';

set newname for datafile 3 to '/data/testdb/oradata/undotbs01.dbf';

set newname for datafile 4 to '/data/testdb/oradata/users01.dbf';

set newname for datafile 5 to '/data/testdb/oradata/mouli01.dbf';

restore database;

}

1. Update datafiles location in controlfile

* RMAN>switch database to copy;

1. Rename redo logfiles

* SQL>alter database rename file ‘/prod/hyd/redo’ to ‘/prod/hydtst/redo’;

1. open the database with resetlogs

* alter database open resetlogs;

1. Add temp file

* alter tablespace temp add tempfile ‘/prod/hydtst/oradata/temp\_01.dbf’ size 100m;
* drop old temp file

**Rman Incremental Backup:**

**Source:**

1. Incremental backup as level 0,1.

* Backup incremental level 0 database;
* Backup incremental level 1 database;

1. Backup archivelog all;
2. Controlfile backup at last.

* Backup current controlfile;

1. Copy pfile from source db to target db.
2. Copy all the backup files to target server.

**Target: If directory structure same as source**

1. Create directories structure as source.
2. Edit pfile according to requirement. (sga,pga,directory path,etc)
3. Initially db name is same as source db name.
4. Start the db in nomount state.
5. Restore controlfile from backup.

* Connect to rman target /
* Restore controlfile from ‘/location/filename/’;

1. Put the database in mount state. RMAN>alter database mount;
2. Register backup’s in target database.

* RMAN>Catalog start with ‘/backup/location’;

1. Restore database;
2. Recover database;(**archives will apply by this**)
3. Check the db state.

* Select name,open\_mode from v$database;

1. Open database with resetlogs.

* Alter database open resetlogs;

**Target: If directory structure different from source**

1. Create directories structure as source.
2. Edit pfile according to requirement. (sga,pga,directory path,etc)
3. Initially db name is same as source db name.
4. Start the db in nomount state.
5. Restore controlfile from backup.

* Connect to rman target /
* Restore controlfile from ‘/location/filename/’;

1. Put the database in mount state. RMAN>alter database mount;
2. Register backup’s in target database.

* RMAN>Catalog start with ‘/backup/location’;

1. Create script for datafiles new location as directories are different from source.

* Run

{

Set new name for datafile 1 to ‘/home/oracle/oradata/system01.dbf’;

Set new name for datafile 2 to ‘/home/oracle/oradata/sysaux01.dbf’;

Set new name for datafile 3 to ‘/home/oracle/oradata/undo01.dbf’;

Set new name for datafile 4 to ‘/home/oracle/oradata/users01.dbf’;

Restore database;

}

1. Rman>Recover database;
2. Update datafiles location in controlfile.

* Rman> switch database to copy;

1. Rename redo log files also.

* Alter database rename file ‘/prod/oracle/redo01’ to /home/oracle/redo01’;
* Alter database rename file ‘/prod/oracle/redo02’ to /home/oracle/redo02’;
* Alter database rename file ‘/prod/oracle/redo03’ to /home/oracle/redo03’;

1. Open database with resetlogs.

* Alter database open resetlogs;

1. Add new temp tablespace and drop old temp files.

* Alter tablespace temp add tempfile ‘/home/oracle/oradata/temp01.dbf’ size 100m;

**Auxiliary duplicate database refresh:**

**Source:**

1. Incremental backup as level 0,1.

* Backup incremental level 0 database;
* Backup incremental level 1 database;
* Backup archivelog all;

1. Controlfile backup at last.

* Backup current controlfile;

**Target:**

1. create required directories
2. Set env
3. Edit pfile

* db\_name=testdb
* db\_file\_name\_convert=‘/SSD/11.2.0.4/‘,’/data/testdb/oradata/‘
* Log\_file\_name\_convert=‘/SSD/11.2.0.4/‘,’/data/testdb/oradata/‘

1. put db in nomount state
2. Connect to rman

* Rman auxiliary /
* Rman>duplicate target database to **testdb** backup location ‘/prod/’ nofilenamecheck;

**Active duplicate database refresh:**

**Source:**

1. Add tnsnames of target db.
2. copy pfile to target

**Target:**

1. Create required directories
2. Listener
3. Copy Password file from source
4. Set env file
5. Edit pfile

* db\_name= ramtst
* db\_file\_name\_convert=‘/prod/ram/oradata/‘,’/prod/ramtst/oradata/‘
* Log\_file\_name\_convert=‘/prod/ram/oradata/‘,’/prod/ramtst/oradata/‘

1. put db in nomount state

**Source:**

connect to Rman

* Rman target / auxiliary=system/system@tns
* Rman>duplicate target database to ramtst backup location ‘/prod/abc‘ nofilenamecheck;

**Export & import in multitenant database’s:**

**TABLE REFRESH:**

* expdp u5/u5@priya directory=BACKUP dumpfile=emp.dmp logfile=emp.log tables=u5.emp
* impdp u5/u5@priya directory=BACKUP dumpfile=emp.dmp logfile=tab.log Tables=u5.emp remap\_table=u5.emp:test

**SCHEMA REFRESH:**

* expdp system/teameis@eis directory=BACKUP dumpfile=eis.dmp logfile=eis.log SCHEMAS=U1
* impdp system/teameis@eis directory=BACKUP dumpfile=eis.dmp logfile=eis.log SCHEMAS=U1 remap\_schema=U1:U5

**Pluggable Database Refresh from another PDB:**

* create pluggable database ABC from priya file\_name\_convert=('priya','abc');