

Recover Gaps in Standby Databases from primary Using incremental backups

- A standby database that has fallen behind from prod can be refreshed using incremental backups taken in production
- A refresh of the controlfile for standby is also needed
- This scenario applies also when a tablespace is transported in production and the standby needs to be updated

Method 1 - using production incremental backups (uses tape)

- **primary**, schedule an incremental backup (either differential or cumulative) or use the existing backup from previous night.
- **primary**, connected to the catalog backup current controlfile for standby;
- **standby**, restore the refreshed controlfile (connected to the catalog and standby db in nomount mode)

```
run{
allocate channel t1 device type sbt_tape;
restore standby controlfile;
}
exit
SQL> alter database mount;
```

- **standby**, clean up the standby controlfile and rename all the files as with the standby names using the 'trick' of cataloging them to the controlfile as datafilecopies
 - ◆ **note:** must not be connected to the primary catalog for this step

```
rman target /
crosscheck copy of database;
delete expired copy of database; -> note can be dangerous, it *can delete read only
list copy of database;
catalog start with '+<DBNAME>_DATADG1'; (note, files may be already in catalog from
```

- switch datafiles to copy (manually as the rman command may fail)

```
set lines 300
spool /tmp/rename.sql
select 'alter database rename file '''||f.name||''' to '''||c.name||''';' SQL from v$backup
spool off
vi /tmp/rename.sql (edit as appropriate to remove non-executable lines)
SQL> alter system set standby_file_management=manual;
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP OFF;

SQL> @/tmp/rename.sql run the script as generated above

SQL> alter system set standby_file_management=auto;
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP ON; (if relevant)
```

- restore files and recover as appropriate

```
run{
allocate channel t1 device type sbt_tape;
restore datafile 229; # probably restore database would also do
}

run{
allocate channel t1 device type sbt_tape;
recover database noredo;
}
```

RecoveryStandbyGaps < PSSGroup < TWiki

- clear redologs before restarting standby recover

```
SQL> select 'alter database clear logfile group '||group#||';' code_to_run from v$log order by group#;
SQL> run the script as generated above
```

- **note:** check with asmcmd and clean up orphaned versions of redo logs
- restart standby recover as appropriate, for example

```
alter database recover managed standby database disconnect from session;
```

Method 2 - using an ad-hoc backup from SCN (disk only)

- ♦ On the standby database, find the SCN from which to start the recovery using these two queries:

```
STANDBY> SELECT CURRENT_SCN FROM V$DATABASE;
STANDBY> select min(change#),min(time) from v$recover_file r, v$datafile f where f.file#=#r
```

- Stop Redo Apply on the standby database:

```
♦ STANDBY> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;
```

- Connect to the primary database as the RMAN target and create an incremental backup from the current SCN of the standby database that was recorded in step 1:

```
RMAN_PRIMARY> RUN {
    BACKUP as compressed backupset device type disk INCREMENTAL
    FROM SCN xxxxxx DATABASE FORMAT '/ORA/dbs00/oracle/backup/ForStandby_%U
    BACKUP device type disk CURRENT CONTROLFILE FOR STANDBY FORMAT '/ORA/dbs0
}
```

- Transfer all backup sets created on the primary system to the standby system (note that there may be more than one backup file created):

```
♦ make sure destination directory exists and there is enough space
```

```
♦ SCP /ORA/dbs00/oracle/backup/ForStandby* standby:/ORA/dbs00/oracle/backup/
```

- Connect to the standby database as the RMAN target, and catalog all incremental backup pieces:

```
♦ RMAN_STANDBY> CATALOG START WITH '/ORA/dbs00/oracle/backup/ForStandby_';
```

- Connect to the standby database as the RMAN target and apply incremental backups and restore controlfile

```
♦ RMAN_STANDBY> RECOVER device type disk DATABASE NOREDO;
```

```
♦ RMAN_STANDBY> SHUTDOWN;
```

```
♦ RMAN_STANDBY> STARTUP NOMOUNT;
```

```
♦ note with 10g must restore a new controlfile for standby or procedure does not work (SCN is not incremented)
```

```
♦ RMAN_STANDBY> RESTORE STANDBY CONTROLFILE FROM
'/ORA/dbs00/oracle/backup/ForStandbyCTRL.bck';
```

```
♦ RMAN_STANDBY> SHUTDOWN;
```

```
♦ RMAN_STANDBY> STARTUP MOUNT;
```

```
♦ RMAN_STANDBY> crosscheck copy of database;
```

```
♦ RMAN_STANDBY> delete expired copy of database;
```

```
♦ note the following is a 'trick' to catalog the actual files as copies so we can then switch. redolog and other ASM files may throw errors, ignore
```

```
♦ RMAN_STANDBY> catalog start with '+<DBNAME>_DATADG1'; (edit)
```

```
♦ in the general case use the query below, switch database to copy only works if all datafiles are oracle-managed (i.e. does not work on compass)
```

```
♦ SQL> select 'alter database rename file '''||f.name||''' to
'''||c.name||''';' SQL from v$backup_copy_details c, v$datafile f where
```

RecoveryStandbyGaps < PSSGroup < TWiki

```
c.file#=f.file# and c.name <> f.name order by c.file#;
```

- ◆ SQL> run the script as generated above
- ◆ alternative if db has all oracle-managed files: RMAN_STANDBY> SWITCH DATABASE TO COPY;

- Clear all standby redo log groups:

- ◆ STANDBY> select 'alter database clear logfile group '||group#||';'
code_to_run from v\$log order by group#; ## run the output

This topic: PSSGroup > RecoveryStandbyGaps

Topic revision: r3 - 2008-12-17 - LucaCanali



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback