Creating a standby database (using Rman) – Linux/Unix

Oracle's Rman can be used to create a standby database without shutting down the primary database. You do not need to create an Rman catalog.

The easiest way to use Rman is to create the backup on disk. However, this requires significant free space if your database is very large. It is also possible to save the backup to tape. In this case you would need a tape or media manager as part of your Rman configuration.

Example:

This example backs up the database to disk.

Preliminary steps:

1. Setup the same Linux or Unix user accounts (Oracle software owner) and groups (dba) on the standby server (preferably with the same uid and the same gid), as on the primary server.
2. Recreate the same Oracle directory structure on the standby server as on the primary server.
3. Install Oracle software on the standby server preferably in the same location as on the primary server.
4. Copy the init.ora file for the production database on the production server to the standby server.
5. Copy the /var/opt/oracle/oratab or /etc/oratab from the production server to the standby server.
6. Copy the password file $ORACLE_HOME/dbs/orapw<$ORACLE_SID> from the production server to the standby server. You may have to create a new password file with the *orapwd* utility.
7. Copy the spfile from the production server to the standby server if it is used.
8. Create a directory on the primary and standby server where the full backup of the database can be kept. In this example /oracle/orabase/backupfile.

On the standby server:

9. Listener must be running on the standby server and you must be able to connect from primary server to the standby database (the SQL*Net port must be open on any firewall between the two servers).
10. Start the standby database in nomount mode.

```
$ sqlplus "/ as sysdba"
SQL> startup nomount
```

On the primary server:

11. Set the Oracle environment to the production database (with the oraenv command in Linux/UNIX) and start Rman:

```
$ rman
connect target /
run{
change archivelog all crosscheck;
allocate channel ch1 type disk;
backup incremental level 0 database format
'/oracle/orabase/backupfile/bk_inc0_%s_%p' setsize=8250000 include current
controlfile for standby ;
sql "alter system archive log current";
backup archivelog all format '/oracle/orabase/backupfile/al_%s_%p';
release channel ch1;
}
```

12. Copy files from /oracle/orabase/backupfile to same location on standby server.
13. Create an alias entry for the standby database in the tnsnames.ora file on the production server. In this example the alias is called standbydb.

14. Start Rman and create the standby database. The password for sys must be the password from the password file.

```
$ rman
connect auxiliary sys/password@standbydb
connect target /
run {
allocate auxiliary channel ch1 type disk;
duplicate target database for standby dorecover nofilenamecheck;
release channel ch1;
}
```

15. Standby database is now complete. Rman has started the standby database in mount standby mode which is correct for Dbvisit to continue. For reference the commands to start a standby database are:

```
SQL> startup nomount
SQL> alter database mount standby database;
```

NOTE:

If the following message is received from Rman:

```
RMAN>  connect auxiliary sys/password@standbydb
RMAN-00571: ===========================================================
RMAN-00569: =============== ERROR MESSAGE STACK FOLLOWS ===============
RMAN-00571: ===========================================================
RMAN-04006: error from auxiliary database: ORA-12528: TNS:listener: all
appropriate instances are blocking new connections
```

Then the following should be added (in red) to the listener.ora on the standby server:

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = /U01/app/oracle/product/10.2.0/db_1)
      (PROGRAM = extproc)
     )
     (SID_DESC =
          (SID_NAME = dbvisitp)
          (ORACLE_HOME = /U01/app/oracle/product/10.2.0/db_1)
     )
   )
```

Where dbvisitp is the name of your database

Creating a standby database (traditional method) - Linux/Unix

The process involves making a full backup of your production database on the primary server. This can be either a hot or a cold backup. If a hot backup is used, then the archive log files created during the backup will also need to be copied to the standby database.

Preliminary steps:

1. Setup the same Linux or Unix user accounts (Oracle software owner) and groups (dba) on the standby server (preferably with the same uid and the same gid) as on the primary server.
2. Recreate the same Oracle directory structure on the standby server as on the primary server.
3. Install Oracle software on the standby server preferably in the same location as on the primary server.
4. Copy the init.ora file for the production database on the production server to the standby server.
5. Copy the /var/opt/oracle/oratab or /etc/oratab from the production server to the standby server (this is not necessary for Windows).

6. Copy the password file $ORACLE_HOME/dbs/orapw<$ORACLE_SID> from the production server to the standby server. You may have to create a new password file with the *orapwd* utility.
7. Copy the spfile file from the production server to the standby server if it is used.

On the primary server:

8. Make a standby controlfile on the production database using the following command:

```
SQL> ALTER DATABASE CREATE STANDBY CONTROLFILE AS
'/oracle/orabase/admin/dbvisitp/create/STANDBY_DBVISIT_control01.ctl' REUSE;
```

   Where *dbvisitp* is the name of the database

9. Copy this standby controlfile to a temporary location on the standby server.
10. Make a cold or a hot backup of the database to either disk or tape.
11. Restore the full production database backup (including any new archive logs) to the standby server.

On the standby server:

12. Replace the existing controlfile(s) (if any) with this new standby controlfile from the temporary location. Make sure the names of the controlfile(s) stay the same.

    Example: if the existing controlfiles were named control01.ctl and control02.ctl, replace the existing controlfiles with the new standby controlfile (created in step 8 above) and make sure that new standby controlfiles are named control01.ctl and control02.ctl.

13. Set the correct Oracle environment (on Linux and Unix with the oraenv command)
14. Start sqlplus:

```
sqlplus /nolog
SQL> connect / as sysdba ;
SQL> startup nomount
SQL> alter database mount standby database ;
SQL> recover standby database ;
Specify log: {<RET>=suggested | filename | AUTO | CANCEL}
```

15. Apply archives until there are no more archives to be applied.
16. Standby database is now complete. The standby database is in mount standby mode which is correct for Dbvisit to continue. For reference the commands to start a standby database are:

```
SQL> startup nomount
SQL> alter database mount standby database;
```