Creating a Physical Standby Database (Doc ID 1475344.1)

In this Document

APPLIES TO:

Oracle Database - Ent Edition - Version 10.1.0.2 to 12.1.0.2 [Release 10.1 to 12.1]
Information in this document applies to any platform.

PURPOSE

** checked for relevance '7-Jul-2015' **

This Document shows the various Possibilities to create a Physical Standby Database on Oracle Database 11.2.0.x.

It should be seen as an Addition and Clarification to the Documentation. You may also review

Oracle® Data Guard, Concepts and Administration, 11g R lease 2 (11.2), E25608-0x

Chapter 3: Creating a Physical Standby Database

Appendix E: Creating a Standby Database with Recovery Manager

which is the Base for this Document. We take an Example Database and show the various Methods to create the Standby Database. Note that only the basic Concepts are shown. Further more complex Scenarios are possible as well, of course. Take a Look into the corresponding Documentation to get the deeper Insight.

Would you like to explore this Topic further with other Oracle Customers, Oracle Employees and Industry Experts ??

You can discuss this Note, show your Experiences or ask Questions about it directly right at the Bottom of this Note in the Discussion Thread about this Document.

If you want to discover Discussions about other Articles und Subjects or even post new Discussions you can access the My Oracle Support Community Page for High Availability Data Guard

SCOPE

This Document is intended for System Adminstrators planning to setup a Standby Database. Depending on the Needs and Prerequisites on the existing Environment you can choose any of the mentioned Ways to create the Standby Database and Data Guard Environment.

Note that this Note does not cover Setting up Log Transport and Log Apply Services in Detail - we take the basic default Setup here. The same is true for the RMAN Environment in Case where different Backup Types and Scenarios are possible. We just try to show the basic Concept here.

DETAILS

Prerequisites and Assumptions to get started

- There already exists a Primary Database you want to create your Standby Database for
- Primary Database is in ARCHIVELOG-Mode and local Archiving is using Fast Recovery Area
- FORCE L ING is enabled on the Primary Database
- The Primary Database is using a SPFILE
- On the Standby Site ORACLE_HOME is installed using the same Oracle Release and Patchlevel
- The Platform of the Standby Database is the same as the Primary Site or a supported Combination as per Note 413484.1
- There is Network Connectivity between the Primary and Standby System
- Listener is configured and running on the Primary and Standby Site
- We are creating a Standby Database using default Log Transport Services in Maximum Performance Mode

Environment Information

For all mentioned Possibilities we are using this Environment

OS:	Linux x64-64
Hostname (Primary):	TSTPRIM
Hostname (Standby):	TSTSTBY
Database Version:	11.2.0.x

DB_NAME (Primary and Standby):	prim_db
SID/DB_UNIQUE_NAME (Primary):	prim_db
SID/DB_UNIQUE_NAME (Standby):	stby_db
Listener Port (Primary and Standby):	1521
Primary Database Files Location:	/oracle/oradata/prim_db
Standby Database Files Location:	/oracle/oradata/stby_db

Preparing the Environment and Standby Instance

- Create TNS-Alias to resolv later. The TNS-Alias to resolv TNSNAMES.ORA. Here it looks like that:

```
PRIM_DB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = TSTPRIM)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = prim_db.world)
    )
  )

STBY_DB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = TSTSTBY)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = stby.world)
    )
  )
```

- Create Folders in ADMIN- and ORADATA-Folders to host the Database Files and Dumpfiles (can use $ORACLE_BASE/admin of the Primary as a Reference which Folders to create – typically those are the ‘adump’- and ‘dpdump’-Folders)

```
$ cd $ORACLE_BASE/admin

$ mkdir stby_db

$ cd stby_db

$ mkdir adump

$ mkdir dpdump
```

- Create a PFILE from the Primary PFILE in the Format ‘init<Standby SID>.ora’

```
SQL> create pfile=’/tmp/initstby_db.ora’ from spfile;
```

- Set the following Initialization Parameters on the Primary Database:

- log_archive_dest_1 = ‘location=USE_DB_RECOVERY_FILE_DEST’
- log_archive_dest_2 = ’service=stby_db async valid_for=(ONLINE_LOGFILES,PRIMARY_ROLE) db_unique_name=stby_db’
- log_archive_dest_state_2 = ’defer’
- log_archive_config= ’dg_config=(prim_db,stby_db)’
- log_archive_max_processes = 8

-> Those Parameters prepare the Primary Database to support a Standby Database ’stby_db’. log_archive_dest_2 is responsible for transferring Redo to the Standby Site later – currently it is deferred, we will enable later once the Standby Database exists. log_archive_config records all db_unique_name’s participating in this Configuration.

- Modify the following Initialization Parameters in the PFILE (initstby_db.ora) for the Standby Database we created before:

- log_archive_dest_1 = ‘location=USE_DB_RECOVERY_FILE_DEST’
- log_archive_config= ’dg_config=(prim_db,stby_db)’
- log_archive_max_processes = 8
- fal_server = ‘prim_db’
- log_file_name_convert = ‘<absolute path or asm diskgroup name of primary online redo log files>’,’<absolute path or asm diskgroup name of standby online redo log files>’
- db_file_name_convert = ‘<absolute path or asm diskgroup name of primary data files>’,’<absolute path or asm diskgroup name of standby data files>’
- db_unique_name = ‘stby_db’

-> Those Parameters are required to be adjusted for the Standby Database. log_archive_dest_1 here defines the local Archive Destination where we will put ArchiveLogs arriving from the Primary Database later. log_archive_config and db_unique_name are required for the Data Guard Configuration (like on the Primary) and fal_server is required for Gap Resolution to be able to automatically fetch Archive Logs from the Primary again in

Case there is a Gap. To automate the Substitution of Database File Locations we set db_file_name_convert and log_file_name_convert. Those Pairs of Values are a simple String Substitution. See Note 1367014.1 for further Details.

-> You may also adjust audit_file_dest and control_files to match with your Environment and Locations.

- Copy prepared 'initstby_db.ora' together with the Passwordfile of the Primary Database (orapwprim_db in $ORACLE_HOME/dbs) to the Standby Site and place both into '$ORACLE_HOME/dbs'-Folder. Then you have to rename the Passwordfile to match with the Standby SID:

```
$ mv orapwprim_db orapwstby_db
```

Create the Physical Standby Database

In the previous Section we prepared the Environments and the Primary Database to support a Standby Database. Now we can proceed and create the Physical Standby Database itself. There are 3 common Ways to perform this which will be shown in the next Section. You can choose any Way most suitable for you:

1. Creating manually via User-Managed Backups
2. Using RMAN Backup-based Duplication
3. Creating a Standby Database from the active Primary Database without a Backup using RMAN Duplicate

Creating a Standby Database via User Managed Backups

We can use a Hot or Cold Database Backup to create the Standby Database. It is only important to create the Standby Controlfile from the Primary after the Backup has completed.

For a Cold Backup we simply perform a clean shutdown of the Primary Database and copy all Datafiles to a temporary Location, then startup the Primary Database again:

```
SQL> shutdown immediate;

$ cp /oracle/oradata/prim_db/*.dbf /backup/

SQL> startup
```

- If you want to use a Host Backup, we have to put the Primary Database into Backup Mode and copy the Datafiles:

```
SQL> alter database begin backup;

$ cp /oracle/oradata/prim_db/*.dbf /backup/

SQL> alter database end backup;
```

- We can create the Standby Controlfile since the Backup is complete (either Hot or Cold Backup)

```
SQL> alter database create standby controlfile as '/backup/control01.ctl';
```

- Now the required Files for the Standby Database are complete and we can transfer those to the Standby Database and place them into the Standby Database File-Folder (in our Ex le to '/oracle/oradata/stby_db/')

- Once the Datafiles are there we can set the Environment and first mount the Standby Database

```
$ export ORACLE_SID = stby_db

SQL> connect / as sysdba

SQL> startup mount
```

- When the Database is mounted proceed with the Post-Creation Steps below

Creating a Standby Database using RMAN (Backup based)

Instead of creating a User-Managed Backup we can also use a RMAN Backup to create the Standby Database.

- In this Case we have to create a RMAN Backup of the Primary Database first:

```
RMAN> connect target /

RMAN> backup database plus archivelog format '/backup/STBY_%U';
```

- Since the Backup already includes the Controlfile, there is no Need to backup or create a Standby Controlfile separately

- Transfer all Backuppieces created by the RMAN Backup to the exact same Folder (in our Case '/backup/') on the Standby Database

- Startup nomount the Standby Database

```
$ export ORACLE_SID = stby_db

SQL> startup nomount
```

- At this Point we can now start the RMAN Duplicate to create the Standby Database

```
$ export ORACLE_SID = stby_db

RMAN> connect target sys/<Password>@prim_db

RMAN> connect auxiliary /

RMAN> duplicate target database for standby nofilenamecheck;
```

-> The RMAN Duplicate now extracts the Controlfile and Datafiles from the Backup to build the Standby Database. Once done it mounts the Standby Database. So we can now proceed with the Post-Creation Steps below.

Creating a Standby Database using RMAN without Backup (from active Database)

It is now possible to create a Physical Standby Database from the active Primary Database, ie. It is not necessary to create a Backup first. The Blocks are transferred to the Standby Database via the Network during the RMAN Duplicate. Before you decide to use this Option to create your Standby Database you should ensure you sufficient Bandwith available to transfer all Database Blocks of the Primary to the Standby. Depending on the Size of your Primary Database and the Bandwith available it might take long Time to complete this Task. If you have a large Database or a slow Network Connection you may consider to use another Option to create your Standby Database.

The Steps to create the Standby Database that Way are similar to a Backup-based RMAN Duplicate, but we don't have to take the Backup, we can directly start the RMAN Duplicate from the Standby Site:

```
$ export ORACLE_SID = stby_db

SQL> startup nomount

RMAN> connect target sys/<Password>@prim_db

RMAN> connect auxiliary /

RMAN> duplicate target database for standby from active database nofilenamecheck;
```

-> The RMAN Duplicate first copies the Controlfile from the Primary Database as a Standby Controlfile, then mounts the Standby Database with this Conrolfile and creates the Datafiles/copies the Database Blocks from the Primary. Once done the Duplicate finishes and leaves the Database in mount-Status. So we can now proceed with the Post-Creation Steps below.

NOTE : If primary and standby sharing same storage do not use NOFILENAMECHECK clause unless you are sure about the directories are different.

ref,

<NOTE 789370.1 > - Creating Physical Standby using RMAN Duplicate Without Shutting down The Primary

Post-Creation Steps to make the Data Guard Environment operational

In the previous Chapters we prepared and created the Physical Standby Database. Now we can start the Log Transport and Log Apply Services to have the Data Guard Environment completely operational.

- First of all we should now add Standby RedoLogs to the new created Standby Database to collect the current Redo arriving from the Primary Database. We can add those using

```
SQL> alter database add standby logfile <group #> (‘<Member>’) size <size>;
```

You should add at least one more Standby RedoLogs Group than corresponding Online RedoLog Group. See Note 219344.1 for further Details and Examples.

- Next we can enable Log Transport Services on the Primary Database which have been prepared initially – performing a Logfile Switch afterwards will finally enable it (all changes to log_archive_dest_n and log_archive_dest_state_n become active after the next Log Switch once set).

```
SQL> alter system set log_archive_dest_state_2 = ‘enable’ scope=both;

SQL> alter system switch logfile;
```

To verify if Standby RedoLogs get allocated on the Standby Database and so Log Transport Services are running query v$standby_log on the Standby Database:

```
SQL> select * from v$standby_log;
```

- Assuming that Log Transport Services are running now, we can finally start Log Apply Services (Managed Recovery):

```
SQL> alter database recover managed standby database using current logfile disconnect;
```

Query v$managed_standby to monitor the Progress of the Managed Recovery:

```
SQL> select * from v$managed_standby where process = ‘MRP0’;
```

Most likely there will now be a FAL Gap Request first to ship and apply all ArchiveLogs created since we took the Backup. Depending on the Time and Amount of Redo generated on the Primary meanwhile it may take a while.

Note that starting Managed Recovery will also attempt to clear/create the Online RedoLogs on the Standby Database proactive to reduce Switchover/Failover Time.

Optional additional Steps

Basically we have now an active Data Guard Configuration. Depending on your Requirements you can proceed performing those Steps:

- Create an SPFILE from the current PFILE on the Standby Database
- Enable Flashback to be able to Flashback this Database (eg. instead of the Primary Database to get historical Data or avoid having to recreate the Standby Data in Case of a RESETLOGS-Operation on the Primary Database)
- Setup and enable a Data Guard Broker Configuration – the Data Guard Broker is an Offset to Data Guard in order to make monitoring and administering of a Data Guard Configuration much easier. Note that using a Data Guard Configuration with Cloud Control (formerly Grid Control) requires the Data Guard Broker
- Raise Protection Mode to Maximum Availability or Maximum Protection
- Proactive setup Log Transport Services from Standby Database to the Primary and create Standby RedoLogs on the Primary for Switchover Purposes is used
- Open the Standby Database READ ONLY and restart Managed Recovery (Active Data Guard). Note that this requires Licensing of the 'Active Data Guard'-Option. However you can open the Standby Database open READ ONLY without Apply Services; this does not require this Option to be licensed
- Setup Fast-Start Failover (requires the Data Guard Broker, too)
- Add additional Standby Database(s)
- Turn the just created Physical Standby Database into a Logical Standby Database

You can directly participate in the Discussion about this Note below. The Frame is the interactive live Discussion - not a Screenshot ;-)

EFERENCES

NOTE:1367014.1 - Usage and Limitation of db_file_name_convert and log_file_name_convert
NOTE:219344.1 - Usage, Benefits and Limitations of Standby Redo Logs (SRL)

Didn't find what you are looking for?