DATA GUARD PHYSICAL STANDBY CONFIGURATION IN 11g

CONFIGURE PHYSICAL STANDBY

In this article I present step by step to configure physical standby database in 11g. I am going to use RMAN Active Database Duplicate command to create a **PHYSICAL STANDBY DATABASE** on the auxiliary server exactly same structure of the source database. We do not need any backup for this process.

DISASTER RECOVERY ENVIRONMENT

```
OPERATING SYSTEM  : RHEL
DATABASE SOFTWARE : 11.2.0.1

PRIMARY DB_NAME   : CRMS
STANDBY DB_NAME   : CRMS

PRIMARY DATABASE UNIQUE NAME : CRMS
STANDBY DATABASE UNIQUE NAME : STBYCRMS

PRIMARY SERVER  → 192.168.1.130  → SERVER1.ORACLE.COM  → SERVER1  → PRODUCTION DATABASE
STANDBY SERVER  → 192.168.1.131  → SERVER2.ORACLE.COM  → SERVER2  → STANDBY DATABASE
```

DATABASE ENVIRONMENT

DATABASE	ROLE	$ORACLE_HOME	DATA_FILES	CTRL_FILES	FLASH_RECOVERY_AREA
CRMS	PRIMARY	/u01/.../db_home1	/u02,/u03	/u01,/u04	/u01/app/oracle/flash_recovery_area
STBYCRMS	STANDBY	/u01/.../db_home1	/u02,/u03	/u01,/u04	/u01/app/oracle/flash_recovery_area

DATABASE	ROLE	TNS ALIAS NAME	REDO LOG FILES
CRMS	PRIMARY	CRMSDB	/u04, /u05
STBYCRMS	STANDBY	STBY_CRMSDB	/u04 (SRL on CRMS DATABASE)

GENERAL CONSIDERATION

Initialization parameter **db_name** should be same name on both primary and standby database.
Initialization parameter **db_unique_name** should be different name on primary & standby databases.

FOR PRIMARY SITE

Archivelog mode should be enabled.
Database Force logging should be enabled.
Create a Password file for Standby Server.
Configure Data Guard specific initialization parameters.
Create a new init.ora file for Standby database.
Copy newly created init.ora file and password file to standby server.
Create SRL (Standby Redo Logs) on Primary Server. (Recommended but NOT mandatory).
Verify space for archivelogs on the primary site. Keep sufficient space for archive destination.
Configure **tnsnames.ora & listener.ora** for connectivity between Primary and Standby database.

FOR STANDBY SITE

Create Directory structure for standby database.
Create remote login password file (If it is not copied from Primary site to Standby site).

DATA GUARD PHYSICAL STANDBY CONFIGURATION IN 11g

```
Configure tnsnames.ora & listener.ora for connectivity between Primary and Standby database.
Startup standby instance in NOMOUNT phase using newly created initialization parameter file.
Execute Duplicate target database command on Primary/Standby site.
IMO, execute Duplicate command from the Standby. Finally Start Recover process.
```

ENSURE DATABASE IN ARCHIVELOG

```
SYS> select name, log_mode from v$database;
...
```

If the database log_mode is NOARCHIVELOG, then change that the database log_mode to ARCHIVELOG.

```
# VERIFY ARCHIVE DETINATION

SYS> archive log list;

Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     14367
Next log sequence to archive   14369
Current log sequence           14369
```

VERIFY SPACE FOR ARCHIVE LOGS

```
SYS> select name, value from v$parameter where name LIKE 'db_recovery%';
...

SYS> alter system set db_recovery_file_dest_size=12000m scope=both;
...
```

ENSURE DATABASE IN FORCE LOGGING MODE

```
SYS> select force_logging from v$database;

FOR
----
NO

SYS> alter database force logging;
...

SYS> select name, log_mode, open_mode, force_logging from v$database;
...
```

Oracle Data Guard recommends the primary database to be run in **FORCE LOGGING** mode. This option helps to avoid problems with unrecoverable transactions executed on primary database. Sometimes SQL statement can be executed in NOLOGGING option, Oracle still generate redo information for standby databases; i.e. force logging takes primacy & all operations are logged into the redo logs.

WHY STANDBY REDO LOG FILES ?

```
The STANDBY REDO LOGS (SRL) is similar to ONLINE REDO LOGS.
SRL files are required to store redo received from the primary database.
```

WHEN STANDBY REDO LOGS ARE REQUIRED?

```
If you use Cascade Destination you need them.
If you are using Real Time Apply on standby database you need them.
If your standby database is in MAX PROCTECTION/MAX AVAILABILITY modes you need them.
```

Oracle recommends that you create a standby redo log on the primary database so that the primary database can switch over quickly to the standby role without any extra step. It is good to create SRL on both sides (Primary and Standby) you switchover quickly.

SRL is not mandatory for Primary Database but its good and useful in role conversion from Primary to Standby Database. It is important to configure the Standby Redo Logs **(SRL)** with the same size as the online redo logs.

Ex (If primary database Online redo log groups whose log files are 200K,
then the Standby Redo Log Groups should have log file sizes of 200K.

FORMULA TO CONFIGURE STANDBY REDOLOG FILES

= (Number of redo log files) **+** 1 ***** (Number of threads)
= 3 + 1 ***1**

Consider your primary database has 3 Online redolog groups and 2 log member per group, It is best performance to configure 1 member per group, additionally create an extra group.

ONLINE REDO LOGS	PRIMARY	STANDBY
Member per group	2	1
Number of log group	3	4

Do NOT multiplex the standby redo logs.

ADD STANDBY LOGFILES ON PRIMARY DATABASE

```
SYS> select b.group#, b.member, a.bytes from v$logfile b, v$log a
where a.group#=b.group# order by group#;

GROUP#     MEMBER                                                      BYTES
---------- ---------------------------------------------------- ----------
    1      /u04/app/oracle/oradata/REDOLOG/crms/redo1a.log        52428800
    1      /u05/app/oracle/oradata/REDOLOG/crms/redo1b.log        52428800
    2      /u05/app/oracle/oradata/REDOLOG/crms/redo2b.log        52428800
    2      /u04/app/oracle/oradata/REDOLOG/crms/redo2a.log        52428800
    3      /u05/app/oracle/oradata/REDOLOG/crms/redo3b.log        52428800
    3      /u04/app/oracle/oradata/REDOLOG/crms/redo3a.log        52428800

6 rows selected.
```

```
# IDENTIFY STANDBY REDO LOGS

SYS> SELECT GROUP#, BYTES FROM V$STANDBY_LOG;
no rows selected.
```

```
# CREATE STANDBY REDOLOG FOR GROUP 4

SYS> ALTER DATABASE ADD STANDBY LOGFILE GROUP 4
('/u04/app/oracle/oradata/REDOLOG/crms/stby_redo04.log') SIZE 50m;
...

# CREATE STANDBY REDOLOG FOR GROUP 5

SYS> ALTER DATABASE ADD STANDBY LOGFILE GROUP 5
('/u04/app/oracle/oradata/REDOLOG/crms/stby_redo05.log') SIZE 50m;
...

# CREATE STANDBY REDOLOG FOR GROUP 6

SYS> ALTER DATABASE ADD STANDBY LOGFILE GROUP 6
('/u04/app/oracle/oradata/REDOLOG/crms/stby_redo06.log') SIZE 50m;
...

# CREATE STANDBY REDOLOG FOR GROUP 7

SYS> ALTER DATABASE ADD STANDBY LOGFILE GROUP 7
('/u04/app/oracle/oradata/REDOLOG/crms/stby_redo07.log') SIZE 50m;
...

SYS> SELECT GROUP#, member from V$LOGFILE where type='STANDBY';
...

SYS> select b.group#, b.member, a.bytes from v$logfile b, v$standby_log a
WHERE a.group#=b.group#;

    GROUP# MEMBER                                                    BYTES
---------- ---------------------------------------------------- ----------
         4 /u04/app/oracle/oradata/REDOLOG/crms/stby_redo4.log    52428800
         5 /u04/app/oracle/oradata/REDOLOG/crms/stby_redo5.log    52428800
         6 /u04/app/oracle/oradata/REDOLOG/crms/stby_redo6.log    52428800
         7 /u04/app/oracle/oradata/REDOLOG/crms/stby_redo7.log    52428800
```

CONFIGURE NETWORK CONNECTIVITY ON PRIMARY SERVER

```
# tnsnames.ora Network Configuration File:
/u01/app/oracle/product/11.2.0/db_home1/network/admin/tnsnames.ora
# Generated by Oracle configuration tools.

CRMSDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.130)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = crms)
      (UR=A)
    )
  )
```

```
STBY_CRMSDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.131)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = stbycrms)
      (UR=A)
    )
  )
```

```
$ vi listener.ora

# listener.ora Network Configuration File:
/u01/app/oracle/product/11.2.0/db_home1/network/admin/listener.ora
# Generated by Oracle configuration tools.

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.130)(PORT = 1521))
    )
  )


ADR_BASE_LISTENER = /u01/app/oracle

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME =crms)
      (ORACLE_HOME = /u01/app/oracle/product/11.2.0/db_home1)
    )
 )
```

```
$ lsnrctl start
...
```

CONFIGURE NETWORK FILES ON STANDBY SERVER

```
$ vi listener.ora

# listener.ora Network Configuration File:
/u01/app/oracle/product/11.2.0/db_home1/network/admin/listener.ora
# Generated by Oracle configuration tools.

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.131)(PORT = 1521))
    )
  )
```

```
ADR_BASE_LISTENER = /u01/app/oracle

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = stbycrms)
      (ORACLE_HOME = /u01/app/oracle/product/11.2.0/db_home1)
    )
 )
```

```
$ vi tnsnames.ora

# tnsnames.ora Network Configuration File:
/u01/app/oracle/product/11.2.0/db_home/network/admin/tnsnames.ora
# Generated by Oracle configuration tools.

STBY_CRMSDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.131 )(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = stbycrms)
      (UR=A)
    )
  )

CRMSDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.130)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = crms)
      (UR=A)
    )
  )
```

```
$ lsnrctl start
...

$ tnsping CRMSDB
...

$ tnsping STBY_CRMSDB
...
```

ORACLE 11g BINARY INSTALLED LOCATION

```
/u01/app/oracle/product/11.2.0/db_home1  →  11g RDBMS BINARY INSTALLED
```

I have installed 11g oracle binary on **/u01** mount point.

Data files are located at /u02 & /u03. Control files are located at /u01 & /u04.

CREATE DIRECTORY STRUCTURE FOR STANDBY SITE

```
$ mkdir -p /u02/app/oracle/oradata/stbycrms
$ mkdir -p /u03/app/oracle/oradata/stbycrms

$ mkdir -p /u01/app/oracle/oradata/CTRL/stbycrms
$ mkdir -p /u04/app/oracle/oradata/CTRL/stbycrms
$ mkdir -p /u01/app/oracle/flash_recovery_area/stbycrms

$ mkdir -p /u01/app/oracle/admin/stbycrms/adump
$ mkdir -p /u01/app/oracle/admin/stbycrms/dpdump

$ mkdir -p /u04/app/oracle/oradata/REDOLOG/stbycrms
$ mkdir -p /u05/app/oracle/oradata/REDOLOG/stbycrms
Alert log location : /u01/app/oracle/diag/rdbms/stbycrms/stbycrms/trace/
```

SET DATAGUARD SPECIFIC INSTANCE PARAMETERS

In this section, we will discuss initialization parameters used in data guard configuration.

PRIMARY ROLE INITIALIZATION PARAMETERS	
DB_NAME	Must be same on primary site and on standby site
DB_UNIQUE_NAME	Must be different on primary site and on standby site
LOG_ARCHIVE_CONFIG	Specify all db_unique_name separated by comma in DR configuration
LOG_ARCHIVE_DEST_n	Specify local and remote archivelog file location
LOG_ARCHIVE_DEST_STATE_n	Specify state of archiving (ENABLE or DIFFER)
REMOTE_LOGIN_PASSWORDFILE	Must be to set EXCLUSIVE mode.
STANDBY ROLE INITIALIZATION PARAMETERS	
FAL_SERVER	For archivelog gap - Where to request missing archive logs
DB_FILE_NAME_CONVERT	Required when directory structure is different for datafiles
LOG_FILE_NAME_CONVERT	Required when directory structure is different for logfiles
STANDBY_FILE_MANAGEMENT	Primary site file additions/deletions are reflected in standby site.

The **DB_NAME** parameter will be **crms** for the both the primary database and the standby database. The **DB_UNIQUE_NAME** will be **crms** for the primary database and **stbycrms** for the standby database.

```
# VERIFY DB_NAME & DB_UNIQUE_NAME OF PRIMARY DATABASE

SYS> show parameter db_name;

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------
db_name                              string      crms

SYS> show parameter db_unique_name;

NAME                                 TYPE        VALUE
------------------------------------ ----------- -----------
db_unique_name                       string      crms
```

```
# SET REMOTE LOGIN PASSWORD TO EXCLUSIVE

SYS> show parameter remote_login; (If it is NOT exclusive, set as 'EXCLUSIVE')
...

SYS> alter system set remote_login_passwordfile='EXCLUSIVE' scope=spfile;
...
```

```
# CREATE PASSWORD FILE ON THE PRIMARY DATABASE SERVER

$ cd /u01/app/oracle/product/11.2.0/db_home1/dbs
$ export ORACLE_SID=crms
$ orapwd file=orapwcrms  password=****** force=y entries=3

# SCP ORAPWCRMS FILE TO THE STANDBY SERVER

$ cd $ORACLE_HOME/dbs
$ scp orapwcrms  orace@192.168.1.131:$ORACLE_HOME/dbs/orapwstbycrms
oracle@192.168.1.131's password:
```

A password file must be created on the primary and copied over to the standby side. **SYS password must be identical on both sites**.

```
SYS> select * from v$pwfile_users;

USERNAME                       SYSDB SYSOP SYSAS
------------------------------ ----- ----- -----
SYS                            TRUE  TRUE  FALSE
```

The view **V$PWFILE USERS** to see the users who have been granted SYSDBA and SYSOPER system privileges.

```
SYS> SELECT name, value FROM v$parameter WHERE
name IN ('db_name','db_unique_name','log_archive_format','remote_login_passwordfile')
ORDER BY NAME;

NAME                                                 VALUE
---------------------------------------------------- ----------------------------------------
db_name                                              crms
db_unique_name                                       crms
log_archive_format                                   %t_%s_%r.dbf
remote_login_passwordfile                            EXCLUSIVE
```

LOG_ARCHIVE_CONFIG – SYNTAX

```
LOG_ARCHIVE_CONFIG =
{
 [SEND | NOSEND]
 [RECEIVE | NORECEIVE]
 [DG_CONFIG=(remote_db_unique_name1 [, ... remote_db_unique_name9) | NODG_CONFIG]
}
```

Its DEFAULT value is: '**SEND, RECEIVE, NODG_CONFIG**'. You need to update only **DG_CONFIG**.

```
SYS> alter system set LOG_ARCHIVE_CONFIG='DG_CONFIG=(crms,stbycrms)' scope=both;
System altered.
```

LOG_ARCHIVE_CONFIG: Enables/Disables the sending of redo logs to remote destinations and the receipt of remote redologs. The DB_UNIQUE_NAME is used in the LOG_ARCHIVE_CONFIG parameter. It lists valid ***db_unique_name*** separated by comma for data guard configuration.

DG_CONFIG attribute to list the **DB_UNIQUE_NAME** for the primary database and participated standby databases. You can get unique database names from the view **V$DATAGUARD_CONFIG**.

```
SYS> select * from V$DATAGUARD_CONFIG;
...

SYS> show parameter LOG_ARCHIVE_CONFIG;

NAME                                 TYPE        VALUE
------------------------------------ ----------- ---------------------------
log_archive_config                   string      DG_CONFIG=(crms,stbycrms)
```

CONFIGURE DESTINATIONS WITH LOG_ARCHIVE_DEST_n

Using **LOG_ARCHIVE_DEST_n** initialization parameter, you can define up to 31 (where n=1,2,3,...31) destinations in Oracle 11g, which specifies either LOCATION or SERVICE attribute to identify either a local disk or remote database destination where redo transport services to transmit redo data.

There are many attributes of the **LOG_ARCHIVE_DEST_n** parameter. Most important parameters attributes are **LOCATION** AND **SERVICE**. Other attributes are optional.

LOG_ARCHIVE_DEST_1='LOCATION

Mention the local destination to store redo data (archive logs) locally on disk.

LOCATION = USE_DB_RECOVERY_FILE_DEST (FRA) *or*

LOCATION = LOCAL_DISK_DIRECTORY (/u05/crms/archives)

LOG_ARCHIVE_DEST_2='SERVICE

Mention the Oracle **net service name** of the standby database that identifies the remote oracle database instance to ship redo data to the standby destination via Oracle net.

SERVICE = STBY_CRMSDB (NET_SERVICE_NAME OF THE STANDBY DATABASE).

```
# SPECIFIES LOCAL DESTINATION FOR ARCHIVING (FRA)

SYS> alter system set LOG_ARCHIVE_DEST_1='location=USE_DB_RECOVERY_FILE_DEST
valid_for=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=crms' scope=both;

System altered.

# SPECIFIES NET_SERVICE_NAME TO REMOTE DATABASE DESTINATION FOR ARCHIVING

SYS> alter system set LOG_ARCHIVE_DEST_2='service=stby_crmsdb LGWR ASYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=stbycrms' scope=both;

System altered.
```

Query **V$ARCHIVE_DEST** to know current settings of ***log_archive_dest_n*** initialization parameter.

```
SYS> show parameter log_archive_dest_1;

NAME                             TYPE        VALUE
-------------------------------- ----------- -----------------------------------------
log_archive_dest_1               string      location=USE_DB_RECOVERY_FILE_DEST
                                             valid_for=(ALL_LOGFILES,ALL_ROLES)
                                              DB_UNIQUE_NAME=crms

SYS> show parameter log_archive_dest_2;

NAME                            TYPE        VALUE
------------------------------ ----------- --------------------------------------------
log_archive_dest_2              string      service=stbycrms LGWR ASYNC
                                            VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
                                            DB_UNIQUE_NAME=stbycrms
```

The destinations from **LOG_ARCHIVE_DEST_1** through **LOG_ARCHIVE_DEST_10** can contain either the **LOCATION** or **SERVICE** attribute to specify local disk directory or remotely accessed database. Destinations from **LOG_ARCHIVE_DEST_11** through **LOG_ARCHIVE_DEST_31** can contain only the **SERVICE** attribute, which does not support the LOCAL destination.

ASYNC/NOAFFIRM - LGWR

Network Transmission Mode is **ASYNC** when using **LGWR**. **SYNC** when using **ARCH** process.
NOAFFIRM is default when **ASYNC** is specified. **AFFIRM** is default when **SYNC** is specified.
ASYNC is default and Protection mode would be **MAXIMUM PERFORMANCE** by setting **ASYNC NOAFFIRM**.

VALID_FOR - ROLE BASED DESTINATIONS

We supply 2 Values for **VALID_FOR** attribute: **REDO_LOG_TYPE**, **DATABASE_ROLE**.
Specified by **VALID_FOR** attribute for **log_archive_dest_n** parameter.
Default value is **VALID_FOR=(ALL_LOGFILES, ALL_ROLES)**.

```
# DATABASE_ROLE CAN BE SET TO THE FOLLOWING VALUES.

PRIMARY_ROLE : This is valid when the database runs only in the Primary role.
STANDBY_ROLE : This is valid when the database runs only in the Standby role.
ALL_ROLES    : This is valid when the database runs in either Primary/Standby role.

# REDO_LOG_TYPE CAN BE SET TO THE FOLLOWING VALUES.

ONLINE_LOGFILE : This is valid only when archiving online redolog files.
STANDBY_LOGFILE: This is valid only when archiving standby redolog files.
ALL_LOGFILES   : This is valid when archiving either Online Redolog files/Standby Redolog files.
```

More over **valid_for** attribute is an optional. But Oracle recommends that you have to define the **valid_for** attribute for each redo transport destination in Data Guard configuration, so that redo transport continues after a role transition (switch over) to any standby database.

```
SYS> alter system set log_archive_dest_state_1=enable scope=both;
System altered.

# ENABLE REMOTE ARCHIVING - ENABLES LOG SHIPPING FROM PRIMARY TO STANDBY

SYS> alter system set log_archive_dest_state_2=enable scope=both;
System altered.
```

Possible values are ENABLE & DEFER. The default is **ENABLE**.
ENABLE specifies that valid log archive destination can be used for subsequent archiving operation.
DEFER specifies destination is excluded from archiving operations until re-enabled.

You can set **LOG_ARCHIVE_DEST_STATE_2=DEFER** optionally.
You can set **LOG_ARCHIVE_DEST_STATE_2=ENABLE,** after you done standby setup completely.

```
# SET STANDBY_FILE_MANAGEMENT=AUTO

SYS> alter system SET STANDBY_FILE_MANAGEMENT=AUTO scope=both;
System altered.
```

STANDBY_FILE_MANAGEMENT: Once you set to **AUTO**, whenever files added or dropped on primary database are reflected automatically to the standby database. The default value is MANUAL.

CONFIGURE FAL_SERVER ON PRIMARY SITE

If we want to make **Primary database CRMS would become Standby database**, and **Standby database STBYCRMS would become Primary database**, additionally we need to configure following parameters.

```
# ON PRIMARY DATABASE (CRMS) SIDE

SYS> alter system set FAL_SERVER=STBY_CRMSDB scope=both;
System altered.

SYS> alter system set FAL_CLIENT=CRMSDB scope=both;
System altered.
```

FAL_CLIENT and **FAL_SERVER** are initialization parameters used to configure log gap detection and resolution at the standby database side of a physical database configuration.

This functionality is provided by log apply services and is used by the physical standby database to manage the detection and resolution of archived redo logs. Refer **(Doc ID 1394472.1)**

FAL_SERVER: - FETCH ARCHIVE LOG (FAL) server for the standby database.
This initialization parameter specifies the Oracle **NET_SERVICE_NAME** for the standby database which is used to request missing archive logs.
FAL_SERVER = NET_SERVICE_NAME OF THE PRIMARY DATABASE

FAL_CLIENT: This is no longer required in 11g. 11g automatically detects it.
The primary database will obtain service name from related **LOG_ARCHIVE_DEST_n** parameter.
FAL_CLIENT = NET_SERVICE_NAME OF THE STANDBY DATABASE

In earlier releases, you set **FAL_CLIENT** parameter on the standby database, and the value is the Oracle Net Service name that the primary database uses to connect the standby database.

On Primary database side we need to set following parameters ***(FAL_SERVER, DB_FILE_NAME_CONVERT, LOG_FILE_NAME_CONVERT)*** to switch roles, the Primary database to become a Standby database.

If Primary database and Standby database directory structure is different, we must set following parameters **db_file_name_convert** and **log_file_name_convert**.

```
SYS> alter system set db_file_name_convert=
'/u02/app/oracle/oradata/stbycrms/','/u02/app/oracle/oradata/crms/',
'/u03/app/oracle/oradata/stbycrms/','/u03/app/oracle/oradata/crms/'
scope=spfile;

System altered.

SYS> alter system set log_file_name_convert=
'/u04/app/oracle/oradata/stbycrms/','/u04/app/oracle/oradata/crms/',
'/u05/app/oracle/oradata/stbycrms/','/u05/app/oracle/oradata/crms/'
scope=spfile;

System altered.
```

log_file_name_convert & db_file_name_convert parameters only need to be set if the directory Structure is different on the Standby site than the Primary site.

```
# SET DB_LOST_WRITE_PROTECT=TYPICAL
SYS> alter system set DB_LOST_WRITE_PROTECT=TYPICAL scope=both;
System altered.
```

The DB_LOST_WRITE_PROTECT parameter checks for data corruption on the primary database during redo transportations and on the standby database during redo apply. Its default values is NONE. **Now time for bounce primary database to bring parameters in effect.**

```
# BOUNCE THE PRIMARY DATABASE

SYS> startup force;
..
...
Database opened.

# CREATE PFILE FROM SPFILE

SYS> create pfile='/tmp/initcrms.ora' from spfile;
File created.

# SCP NEWLY CREATED PFILE TO STANDBY SERVER

$ cd /tmp
$ scp initcrms.ora oracle@192.168.1.131:/tmp/initstbycrms.ora
Password:
```

Now we copied **initcrms.ora** file to standby server and also renamed as **initstbycrms.ora**; once we modify **initstbycrms.ora** file as per standby database, then move that file to **$ORACLE_HOME/dbs** location. Already I have copied **orapwcrms.ora** file from primary server to standby server and also renamed as **orapwstbycrms.ora** at **$ORACLE_HOME/dbs** location.

DATAGUARD PARAMETERS FOR PRIMARY DATABASE - CRMS

```
##################################################################
################# DATAGUARD SPECIFIC PARAMETERS ##################
##################################################################

log_archive_config='DG_CONFIG=(crms,stbycrms)'
log_archive_dest_1='location=USE_DB_RECOVERY_FILE_DEST
valid_for=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=crms'
log_archive_dest_2='service=stby_crmsdb LGWR ASYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=stbycrms'
db_name=crms
db_unique_name=crms
log_archive_dest_state_2='ENABLE'
log_archive_dest_state_2='ENABLE'
fal_server='STBY_CRMSDB'
fal_client=CRMSDB        # OPTIONAL
standby_file_management='AUTO'
*.db_lost_write_protect='TYPICAL'
db_file_name_convert=('/u02/app/oracle/oradata/stbycrms/','/u02/app/oracle/oradata/crms/',
'/u03/app/oracle/oradata/stbycrms/','/u03/app/oracle/oradata/crms/')
log_file_name_convert=
('/u04/app/oracle/oradata/REDOLOG/stbycrms/','/u04/app/oracle/oradata/REDOLOG/crms/',
 '/u05/app/oracle/oradata/REDOLOG/stbycrms/','/u05/app/oracle/oradata/REDOLOG/crms/')
```

DATAGUARD PARAMETERS FOR STANDBY DATABASE – STBYCRMS

```
##################################################################
################# DATAGUARD SPECIFIC PARAMETERS ##################
##################################################################

log_archive_config='DG_CONFIG=(crms,stbycrms)'
log_archive_dest_1='location=USE_DB_RECOVERY_FILE_DEST
valid_for=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=stbycrms'
log_archive_dest_2='service=crmsdb LGWR ASYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=crms'
db_name=crms
db_unique_name=stbycrms
log_archive_dest_state_2='ENABLE'
log_archive_dest_state_2='ENABLE'
fal_server='CRMSDB'
fal_client='STBY_CRMSDB'    # OPTIONAL
standby_file_management='AUTO'
*.db_lost_write_protect='TYPICAL'
db_file_name_convert=('/u02/app/oracle/oradata/crms/','/u02/app/oracle/oradata/stbycrms/',
'/u03/app/oracle/oradata/crms/','/u03/app/oracle/oradata/stbycrms/')
log_file_name_convert=
('/u04/app/oracle/oradata/REDOLOG/crms/','/u04/app/oracle/oradata/REDOLOG/stbycrms/',
'/u05/app/oracle/oradata/REDOLOG/crms/','/u05/app/oracle/oradata/REDOLOG/stbycrms/')
```

```
# PRIMARY DATABASE PFILE INITCRMS.ORA

$ vi /tmp/initcrms.ora

crms.__oracle_base='/u01/app/oracle' # ORACLE_BASE set from environment
*.audit_file_dest='/u01/app/oracle/admin/crms/adump'
*.audit_trail='DB'
*.compatible='11.2.0.0.0'
*.control_files='/u01/app/oracle/oradata/CTRL/crms/control01.ctl',
'/u01/app/oracle/flash_recovery_area/crms/control02.ctl',
'/u04/app/oracle/oradata/CTRL/crms/control03.ctl'
*.db_block_size=8192
*.db_domain=''
*.db_file_name_convert=('/u02/app/oracle/oradata/stbycrms/','/u02/app/oracle/oradata/crms/',
'/u03/app/oracle/oradata/stbycrms/','/u03/app/oracle/oradata/crms/')
*.db_lost_write_protect='TYPICAL'
*.db_name='crms'
*.db_unique_name='crms'
*.db_recovery_file_dest='/u01/app/oracle/flash_recovery_area'
*.db_recovery_file_dest_size=10000M
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=crmsXDB)'
*.fal_client='CRMSDB'  # OPTIONAL
*.fal_server='STBY_CRMSDB'
*.log_archive_config='DG_CONFIG=(crms,stbycrms)'
*.log_archive_dest_1='location=USE_DB_RECOVERY_FILE_DEST
valid_for=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=crms'
*.log_archive_dest_2='service=stby_crmsdb LGWR ASYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=stbycrms'
*.log_archive_dest_state_1='ENABLE'
*.log_archive_dest_state_2='ENABLE'
*.log_archive_format='%t_%s_%r.dbf'
*.log_archive_max_processes=20
*.log_file_name_convert=('/u04/app/oracle/oradata/REDOLOG/stbycrms',
'/u04/app/oracle/oradata/REDOLOG/crms/',
'/u05/app/oracle/oradata/REDOLOG/stbycrms/','/u05/app/oracle/oradata/REDOLOG/crms/')
*.memory_target=1460M
*.nls_date_format='DD-MON-YYYY hh24:MI:SS'
*.open_cursors=300
*.processes=150
*.remote_login_passwordfile='EXCLUSIVE'
*.service_names='crms'
*.standby_file_management='AUTO'
*.undo_retention=2800
*.undo_tablespace='UNDOTBS'
```

I won’t use Primary database pfile **initcrms.ora** file; but I will start standby database instance using **initstbycrms.ora** file. Compare b**oth** files. You can understand clearly how initialization parameters are configured for primary database and standby database.

```
# STANDBY DATABASE PFILE INITSTBYCRMS.ORA

$ vi /tmp/initstbycrms.ora

crms.__oracle_base='/u01/app/oracle' # ORACLE_BASE set from environment
*.audit_file_dest='/u01/app/oracle/admin/stbycrms/adump'
*.audit_trail='DB'
*.compatible='11.2.0.0.0'
*.control_files='/u01/app/oracle/oradata/CTRL/stbycrms/control01.ctl',
'/u01/app/oracle/flash_recovery_area/stbycrms/control02.ctl',
'/u04/app/oracle/oradata/CTRL/stbycrms/control03.ctl'
*.db_block_size=8192
*.db_domain=''
*.db_file_name_convert=('/u02/app/oracle/oradata/crms/','/u02/app/oracle/oradata/stbycrms/',
'/u03/app/oracle/oradata/crms/','/u03/app/oracle/oradata/stbycrms/')
*.db_lost_write_protect='TYPICAL'
*.db_name='crms'
*.db_unique_name='stbycrms'
*.db_recovery_file_dest='/u01/app/oracle/flash_recovery_area'
*.db_recovery_file_dest_size=10000M
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=stbycrmsXDB)'
*.fal_client='STBY_CRMSDB'  # OPTIONAL
*.fal_server='CRMSDB'
*.log_archive_config='DG_CONFIG=(crms,stbycrms)'
*.log_archive_dest_1='location=USE_DB_RECOVERY_FILE_DEST
valid_for=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=stbycrms'
*.log_archive_dest_2='service=crmsdb LGWR ASYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=crms'
*.log_archive_dest_state_1='ENABLE'
*.log_archive_dest_state_2='ENABLE'
*.log_archive_format='%t_%s_%r.dbf'
*.log_file_name_convert=('/u04/app/oracle/oradata/REDOLOG/crms/',
'/u04/app/oracle/oradata/REDOLOG/stbycrms/',
'/u05/app/oracle/oradata/REDOLOG/crms/','/u05/app/oracle/oradata/REDOLOG/stbycrms/')
*.memory_target=1460M
*.nls_date_format='DD-MON-YYYY hh24:MI:SS'
*.open_cursors=300
*.processes=150
*.remote_login_passwordfile='EXCLUSIVE'
*.standby_file_management='AUTO'
*.undo_retention=2800
*.undo_tablespace='UNDOTBS'
```

```
# COPY MODIFIED PFILE TO $ORACLE_HOME/dbs LOCATION ON THE STABNDBY SERVER

$ cp /tmp/initstbycrms.ora $ORACLE_HOME/dbs
$ ls -l $ORACLE_HOME/dbs/init*
...
```

UPDATE ORATAB FILE

```
# ADD ENTRIES IN /etc/oratab FILE

$ vi /etc/oratab

stbycrms:/u01/app/oracle/product/11.2.0/db_home1:N
```

STARTUP STANDBY DATABASE INSTANCE

```
# STARTUP COMMAND SHOULD BE WITH NOMOUNT.

$ export ORACLE_SID=stbycrms
$ sqlplus "/as sysdba"
..
...
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SYS> startup nomount;
ORACLE instance started.

..
...
SYS> exit
...
```

Connect Primary database as Target database & standby database as auxiliary instance through RMAN. Already Primary database must be up and running and ensure Standby database is in nomount stage. It’s time to clone Primary database to Standby database.

```
# ON STANDBY SERVER CONNECT TO BOTH THE TARGET (CRMS) AND THE AUXILIARY (STBYCRMS) THROUGH RMAN

$ hostname -i
192.168.1.131

$ rman target sys/passwd@CRMSDB auxiliary sys/passwd@STBY_CRMSDB

Recovery Manager: Release 11.2.0.1.0 - Production on Fri Oct 30 21:07:00 2015

Copyright (c) 1982, 2009, Oracle and/or its affiliates.  All rights reserved.

connected to target database: CRMS (DBID=1570419964)
connected to auxiliary database: CRMS (not mounted)


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

Starting Duplicate Db at 31-OCT-15
using target database control file instead of recovery catalog
allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: SID=18 device type=DISK
```

```
contents of Memory Script:
{
   backup as copy reuse
   targetfile  '/u01/app/oracle/product/11.2.0/db_home1//dbs/orapwcrms' auxiliary format
 '/u01/app/oracle/product/11.2.0/db_home/dbs/orapwstbycrms'   ;
}
executing Memory Script

Starting backup at 31-OCT-15
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=1 device type=DISK
Finished backup at 31-OCT-15

contents of Memory Script:
{
   backup as copy current controlfile for standby auxiliary format
'/u01/app/oracle/oradata/CTRL/stbycrms/control01.ctl';
   restore clone controlfile to  '/u01/app/oracle/flash_recovery_area/stbycrms/control02.ctl'
from '/u01/app/oracle/oradata/CTRL/stbycrms/control01.ctl';
   restore clone controlfile to  '/u04/app/oracle/oradata/CTRL/stbycrms/control03.ctl' from
 '/u01/app/oracle/oradata/CTRL/stbycrms/control01.ctl';
}
executing Memory Script

Starting backup at 31-OCT-15
using channel ORA_DISK_1
channel ORA_DISK_1: starting datafile copy
copying standby control file
..
...
[Trimmed]
```

Once duplication is completed, connect to the standby database through SQL.

```
# CONNECTING TO THE STANDY DATABASE

$ rlsqlplus sys/crms@STBY_CRMSDB as sysdba
..
...

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SYS>
```

CHECKING THE STANDBY DATABASE

```
SYS> select NAME, DB_UNIQUE_NAME, OPEN_MODE, DATABASE_ROLE, PROTECTION_MODE from v$database;

NAME      DB_UNIQUE_NAME  OPEN_MODE  DATABASE_ROLE          PROTECTION_MODE
--------- --------------- ---------- ---------------------- ----------------------
CRMS      stbycrms        MOUNTED    PHYSICAL STANDBY       MAXIMUM PERFORMANCE
```

```
SYS> select INSTANCE_NAME, STATUS, SWITCHOVER_STATUS from v$instance, v$database;

INSTANCE_NAME    STATUS       SWITCHOVER_STATUS
---------------- ------------ --------------
stbycrms         MOUNTED       NOT ALLOWED
```

ON PRIMARY DATABASE

```
SYS> select NAME, DB_UNIQUE_NAME, OPEN_MODE, DATABASE_ROLE, PROTECTION_MODE from v$database;

NAME     DB_UNIQUE_NAME   OPEN_MODE    DATABASE_ROLE     PROTECTION_MODE
-------- ---------------- ------------ ----------------- -----------------------
CRMS     crms             READ WRITE    PRIMARY           MAXIMUM PERFORMANCE

SYS> select INSTANCE_NAME, STATUS, SWITCHOVER_STATUS from v$instance, v$database;

INSTANCE_NAME    STATUS       SWITCHOVER_STATUS
---------------- ------------ -----------------
crms             OPEN         TO_STANDBY

SYS> archive log list;
Database log mode              Archive Mode
Automatic archival             Enabled
Archive destination            USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence     14433
Next log sequence to archive   14435
Current log sequence           14435

SYS> select SEQUENCE#, FIRST_TIME, NEXT_TIME, APPLIED FROM V$ARCHIVED_LOG order by sequence#;

SEQUENCE# FIRST_TIME           NEXT_TIME            APPLIED
---------- -------------------- -------------------- ---------
     ..     ...         ...     ...         ...      ..
     14432  31-OCT-2015 11:30:38  31-OCT-2015  12:07:24  NO
     14432  31-OCT-2015 11:30:38  31-OCT-2015  12:07:24  NO
     14433  31-OCT-2015 12:07:24  31-OCT-2015  12:40:31  NO
     14433  31-OCT-2015 12:07:24  31-OCT-2015  12:40:31  NO
     14434  31-OCT-2015 12:40:31  31-OCT-2015  12:43:22  NO
     14434  31-OCT-2015 12:40:31  31-OCT-2015  12:43:22  NO
```

ON STANDBY SERVER

```
SYS> select SEQUENCE#, FIRST_TIME, NEXT_TIME, APPLIED FROM V$ARCHIVED_LOG order by sequence#;

SEQUENCE# FIRST_TIME           NEXT_TIME            APPLIED
---------- -------------------- -------------------- ---------
     14432  31-OCT-2015 11:30:38  31-OCT-2015 12:07:24  NO
     14433  31-OCT-2015 12:07:24  31-OCT-2015 12:38:31  NO
     14434  31-OCT-2015 12:38:31  31-OCT-2015 12:43:22  NO
```

START MANAGED RECOVERY PROCESS (MRP) ON STANDBY DATABASE

```
$ ps -ef | grep mrp
oracle   20051 19907  0 12:30 pts/1    00:00:00 grep mrp

# START REDO APPLY PROCESS ON THE STANDBY DATABASE.

SYS> alter database recover managed standby database; or
SYS> alter database recover managed standby database disconnect;  or
SYS> alter database recover managed standby database disconnect from session;

$ ps -ef | grep mrp
oracle   19950     1  1 12:40 ?        00:00:00 ora_mrp0_stbycrms
```

```
SYS> select SEQUENCE#, FIRST_TIME, NEXT_TIME, APPLIED FROM V$ARCHIVED_LOG order by sequence#;

SEQUENCE#  FIRST_TIME           NEXT_TIME            APPLIED
---------- -------------------- -------------------- ---------
     14432  31-OCT-2015 11:30:38  31-OCT-2015 12:07:24  YES
     14433  31-OCT-2015 12:07:24  31-OCT-2015 12:38:31  YES
     14434  31-OCT-2015 12:38:31  31-OCT-2015 12:43:22  IN-MEMORY
```

IS STANDBY IN SYNC WITH PRIMARY ?

```
# ON THE PRIMARY DATABASE

SYS> select thread#, max(sequence#) from v$archived_log group by thread#;

   THREAD#  MAX(SEQUENCE#)
---------- --------------
         1          14434


# ON STANDBY DATABASE

SYS> select thread#, max(sequence#) from v$archived_log where applied='YES' group by thread#;

  THREAD#  MAX(SEQUENCE#)
---------- --------------
         1          14434

SYS> select CLIENT_PROCESS, PROCESS, THREAD#, SEQUENCE#, STATUS   FROM v$managed_standby
WHERE client_process='LGWR' or process='MRP0' ORDER BY PROCESS;

CLIENT_P PROCESS      THREAD#  SEQUENCE# STATUS
-------- --------- ---------- ---------- ------------
N/A      MRP0               1      14435 WAIT_FOR_LOG
LGWR     RFS                1      14435 IDLE
```

```
Maximum SEQUENCE# generated on the Primary database : 14434
Maximum SEQUENCE# applied   on the Primary database : 14434
Now Standby database is in SYNC with Primary.
```

TEST LOG TRANSPORT FROM PRIMARY TO STANDBY

```
SYS> alter system switch logfile;
System altered.

SYS> /
System altered.

SYS> /
System altered.

SYS> /
System altered.

SYS> /
System altered.

SYS> /
System altered.


SYS> select thread#, max(sequence#) from v$archived_log group by thread#;
...
# ON STANDBY
SYS> select thread#, max(sequence#) from v$archived_log where applied='YES' group by thread#;
...
```

TROUBLESHOOT REDO SHIPPING PRIMARY TO STANDBY

In case you are facing error related to archives are NOT shipping to standby site, use following query on Primary Site to diagnose it.

```
SYS> select dest_name, status, error from v$archive_dest where dest_name='LOG_ARCHIVE_DEST_2';
...

SYS> select message from v$dataguard_status;
...
```

You might face (Heartbeat failed or Network Hung), even if the standby is up and running and also MRP is active, the primary database hesitates to ship the redo to the standby and throws some network related errors like following errors.

```
ARC2: Error 16198 due to hung ARCH operation to  ...
PING[ARC1]: Heartbeat failed to connect to standby ...
NSA: Error 3135 archiving log 1 to ...
FAL[server, ARC3]: Error 12541 creating remote archivelog file ...
WARN: ARC3: Terminating ARCH (pid 32415) hung on a network operation

# CONNECT FROM THE PRIMARY TO STANDBY AND FROM THE STANDBY TO THE PRIMARY DATABASE.

$ sqlplus sys/passwd@standby_db_net_service_name as sysdba
$ sqlplus sys/passwd@STBY_CRMSDB as sysdba
$ sqlplus sys/passwd@primary_db_net_service_name as sysdba
$ sqlplus sys/passwd@CRMSDB as sysdba
```

If you are able to connect from primary to standby and also standby to primary, still Primary throws network related errors try following steps.

1. SET LOG_ARCHIVE_DEST_STATE_2=DEFER on the Primary to the Standby and ENABLE it back.
2. Cancel the MRP on the standby and restart the recovery.
3. Finally bounce the standby database.

```
# TO STOP REDO APPLY PROCESS

SYS> alter database recover managed standby database cancel;
...
```

USEFULE LINKS TO DIAGNOSE

Link for Monitor Data Guard Transport here.

Script-to-Collect-Diagnostic-Information-in-a-Dataguard-Environment here

Online redo logs will be created on the Standby side by RMAN DUPLICATE command based on the value of **log_file_name_convert** parameter.

```
SYS> select group#, member from v$logfile;

    GROUP# MEMBER
---------- ------------------------------------------------------------
         3 /u04/app/oracle/oradata/REDOLOG/stbycrms/redo3a.log
         2 /u04/app/oracle/oradata/REDOLOG/stbycrms/redo2a.log
         1 /u04/app/oracle/oradata/REDOLOG/stbycrms/redo1a.log
         3 /u05/app/oracle/oradata/REDOLOG/stbycrms/redo3b.log
         1 /u05/app/oracle/oradata/REDOLOG/stbycrms/redo1b.log
         2 /u05/app/oracle/oradata/REDOLOG/stbycrms/redo2b.log
         4 /u04/app/oracle/oradata/REDOLOG/stbycrms/stby_redo4.log
         5 /u04/app/oracle/oradata/REDOLOG/stbycrms/stby_redo5.log
         6 /u04/app/oracle/oradata/REDOLOG/stbycrms/stby_redo6.log
         7 /u04/app/oracle/oradata/REDOLOG/stbycrms/stby_redo7.log

10 rows selected.
```

```
SYS> select member from v$logfile where type='STANDBY';

MEMBER
------------------------------------------------------------
/u04/app/oracle/oradata/REDOLOG/stbycrms/stby_redo4.log
/u04/app/oracle/oradata/REDOLOG/stbycrms/stby_redo5.log
/u04/app/oracle/oradata/REDOLOG/stbycrms/stby_redo6.log
/u04/app/oracle/oradata/REDOLOG/stbycrms/stby_redo7.log
```

On Primary Site, Standby Redo Logs will not be used unless you do switchover.

On Standby Site, Online Redo logs will not be used, instead Standby Redo Logs(SRL) will be used.

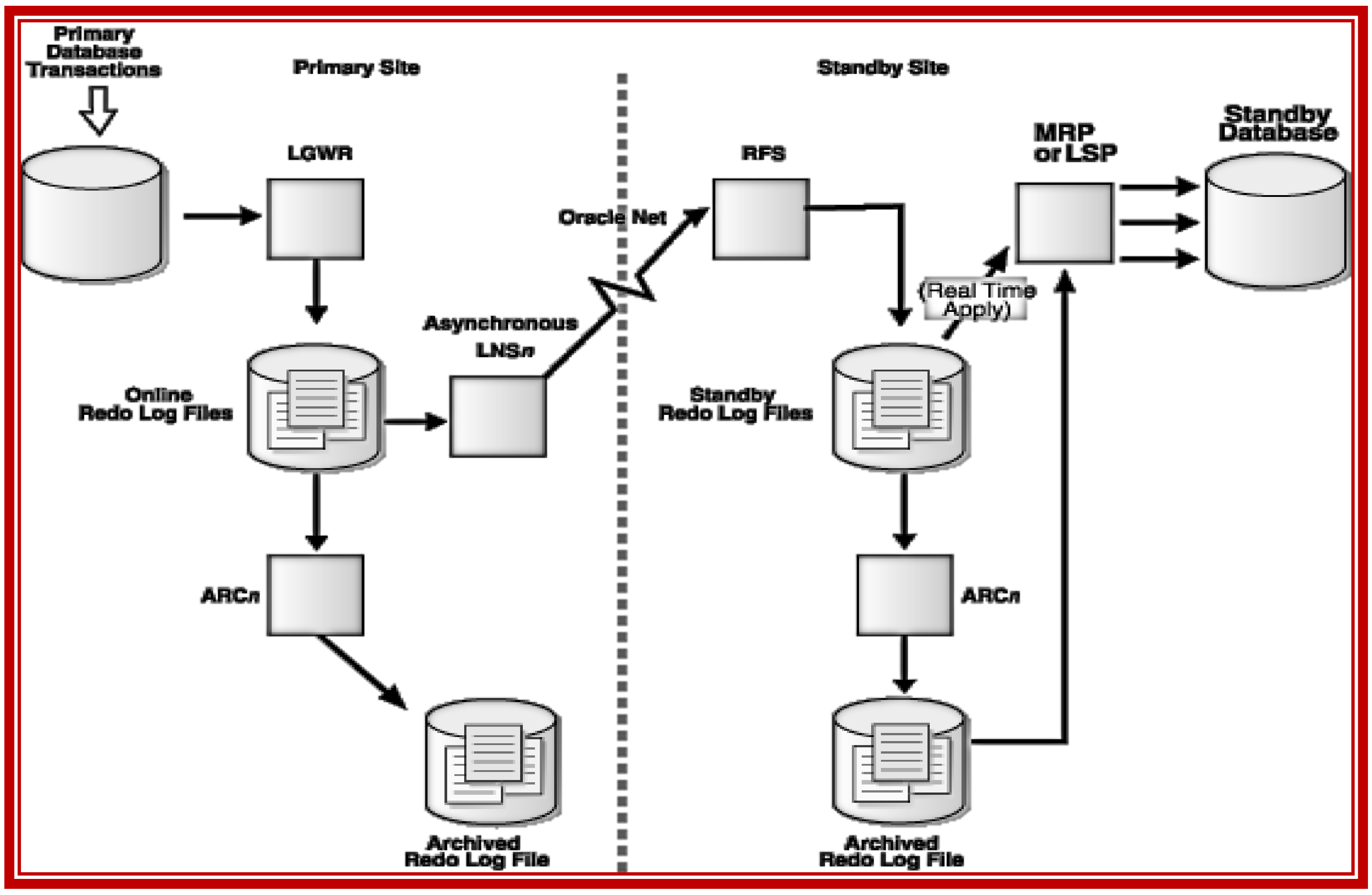
```
# QUERYING STANDBY REDOLOG FILES ON STANDBY SITE

SYS> select group#, sequence#, bytes, used,status from v$standby_log;

    GROUP#  SEQUENCE#      BYTES       USED STATUS
---------- ---------- ---------- ---------- ----------
         4          0   52428800        512 UNASSIGNED
         5      14445   52428800   26890240 ACTIVE
         6          0   52428800        512 UNASSIGNED
         7          0   52428800        512 UNASSIGNED
```

Suppose Standby Redolog files are not configured, then the primary can only transport redo once the Online Redo Logs fills up and archived to Primary archive destination. Then the archived redo will be sent to the standby. Finally redo is transported via archivelog files.

LGWR ASYNC ARCHIVAL PROCESSING WITH LNS

```
*.log_archive_dest_1='location=USE_DB_RECOVERY_FILE_DEST
*.log_archive_dest_2='service=stby_crmsdb LGWR ASYNC
```

10g & 11g Redo Shipping Flow is same



LGWR ASYNC ARCHIVAL WITH NETWORK SERVER LNSn PROCESSESS

If we want to use of **REAL TIME APPLY,** first we have to configure **SRL** on the Standby side. Let's assume Standby Redo Logs are configured on standby side,

LNS ships redo to RFS and RFS writes redo on Standby Redolog files.
Redo is applied directly through the **SRL (Real Time Apply)** and does NOT have to wait for the SRL's to be archived. Finally ARCn process archives the standby redo logs to archive destination.

CHECK REAL TIME APPLY IS ENABLED OR NOT

```
# ON STANDBY DATBASE

SYS> select DATABASE_ROLE, DB_UNIQUE_NAME, PROTECTION_LEVEL, OPEN_MODE from v$database;

DATABASE_ROLE    DB_UNIQUE_NAME                 PROTECTION_LEVEL     OPEN_MODE
---------------- ------------------------------ -------------------- --------------------
PHYSICAL STANDBY stbycrms                       MAXIMUM PERFORMANCE  MOUNTED

SYS> select DEST_ID, DEST_NAME, STATUS, TYPE, SRL, RECOVERY_MODE
from v$archive_dest_status where dest_id=1;

   DEST_ID DEST_NAME              STATUS    TYPE           SRL RECOVERY_MODE
---------- ---------------------- --------- -------------- --- -----------------------
         1 LOG_ARCHIVE_DEST_1     VALID     LOCAL          NO  MANAGED
```

If it's not a REAL TIME APPLY one can see **RECOVERY_MODE is MANAGED**

ENABLE REAL TIME APPLY

```
# STOP REDO APPLY

SYS> alter database recover managed standby database cancel;
...

SYS> alter database open;
...
```

Even though I did not specify read only option explicitly, open_mode would be read only.

```
SYS> select database_role,db_unique_name, protection_level, open_mode from v$database;

DATABASE_ROLE    DB_UNIQUE_NAME                 PROTECTION_LEVEL     OPEN_MODE
---------------- ------------------------------ -------------------- --------------------
PHYSICAL STANDBY stbycrms                       MAXIMUM PERFORMANCE  READ ONLY

# ENABLE REAL TIME APPLY {USING CURRENT LOGFILE} CLAUSE

SYS> alter database recover managed standby database using current logfile disconnect; or
SYS> alter database recover managed standby database using current logfile
disconnect from session;

Database altered.


SYS> select DATABASE_ROLE,DB_UNIQUE_NAME, PROTECTION_LEVEL, OPEN_MODE from v$database;

DATABASE_ROLE    DB_UNIQUE_NAME                 PROTECTION_LEVEL     OPEN_MODE
---------------- ------------------------------ -------------------- --------------------
PHYSICAL STANDBY stbycrms                       MAXIMUM PERFORMANCE  READ ONLY WITH APPLY
```

READ ONLY WITH APPLY - A Physical Standby database is open in real-time query mode.

```
# REAL TIME APPLY ENABLED

SYS> select DEST_ID,DEST_NAME,STATUS,TYPE,SRL,RECOVERY_MODE
from v$archive_dest_status where dest_id=1;

   DEST_ID DEST_NAME              STATUS    TYPE            SRL RECOVERY_MODE
---------- ---------------------- --------- --------------- --- -----------------------
         1 LOG_ARCHIVE_DEST_1     VALID     LOCAL           NO  MANAGED REAL TIME APPLY

$ ps -ef | grep mrp
...
```

MANAGED REAL TIME APPLY - Log apply services recover redo data from standby redo logs at the same time the logs are being written to, as opposed to recovering redo from archived redo logs when a log switch occurs.

HOW TO CHECK DATAGUARD PROCESSES ?

We can query the **PID** column of **V$MANAGED_STANDBY** view to get the details of the data guard related processes at OS level.

```
SYS> select process, status, sequence#, pid, client_process from v$managed_standby;
...
```

DATA DICTIONARY VIEWS

PHYSICAL STANDBY DATABASE	
V$archive_dest	V$dataguard_status
V$archive_gap	Vlog, vlogfile
V$archived_log	V$log_history
V$database	V$managed_standby
V$dataguard_config	V$Standby_log