

Data Guard 11g
Installation and Configuration
On Oracle RAC Systems

An Oracle White Paper
October 2008

Maximum
Availability
Architecture

Oracle Best Practices For High Availability

Contents

1	INTRODUCTION	4
1.1	PURPOSE	4
1.2	BACKGROUND	4
1.3	SCOPE & APPLICATION	4
1.4	RELATED DOCUMENTS.....	5
1.5	ASSUMPTIONS.....	5
2	ARCHITECTURE AND CONCEPTS	6
2.1	GLOSSARY.....	6
2.2	RAC ARCHITECTURE.....	6
2.3	DATA GUARD ARCHITECTURE.....	6
2.4	NETWORK.....	7
2.5	DATA GUARD ARCHITECTURE.....	8
2.5.1	Log Transport Services.....	8
2.5.2	Log Apply Services.....	8
2.5.3	Data Guard Broker.....	9
2.6	HOW IT WORKS	9
2.6.1	Archived Log Shipping.....	9
2.6.2	Standby Redo logs	9
2.7	LEVELS OF PROTECTION	10
2.7.1	Maximum Protection.....	10
2.7.2	Maximum Availability	10
2.7.3	Maximum Performance	10
2.7.4	Pros and Cons.....	10
3	PREREQUISITES.....	11
3.1	HARDWARE	11
3.2	NETWORK.....	11
3.3	SOFTWARE	11
3.4	REAL APPLICATION CLUSTERS	11
3.4.1	ASM.....	11
3.4.2	Raw Devices	11
4	CREATING A DATA GUARD ENVIRONMENT.....	13
4.1	ASSUMPTIONS.....	13
4.2	PROCEDURE - SUMMARY.....	13
4.3	CREATING THE RAC STANDBY DATABASE.....	14
4.3.1	Configure Primary and Standby sites.....	14
4.3.2	Install Oracle Software on each site.....	14
4.3.3	Server Names / VIPs.....	14
4.3.4	Configure Oracle Networking	15
4.3.5	Configure ASM on each Site	17

4.3.6	Prepare Primary Database for Duplication	18
4.3.7	Duplicate the Primary database.....	20
4.3.8	Create an SPFILE for the Standby Database	20
4.3.9	Create secondary control files	21
4.3.10	Cluster-enable the Standby Database	22
4.3.11	Temporary Files	23
4.3.12	Create Standby Redo Logs.....	23
4.4	CONFIGURING DATA GUARD USING SQL PLUS	24
4.4.1	Introduction.....	24
4.4.2	Configure the Standby Database	24
4.4.3	Configure the Primary Database	25
4.4.4	Set the Protection Mode	25
4.4.5	Enable Redo Transport & Redo Apply	26
4.5	CONFIGURING DATA GUARD USING THE DATA GUARD BROKER.....	26
4.5.1	Introduction.....	26
4.5.2	Broker Configuration Files	27
4.5.3	Enabling the Broker	27
4.5.4	Creating a Broker Configuration	27
4.5.5	Enable the Broker Configuration.....	27
4.5.6	Broker Customisation.....	28
5	MONITORING	29
5.1.1	Introduction.....	29
5.1.2	Log Files	29
5.1.3	Fixed Views	29
6	MANAGEMENT.....	30
6.1	SWITCHOVER	30
6.1.1	Switchover using SQL Plus	30
6.1.2	Switchover using Data Guard Broker	30
6.2	FAILOVER.....	31
6.2.1	Failover using SQL Plus.....	31
6.2.2	Failover using Data Guard Broker	31
6.3	FORCED FAILOVER	31
6.3.1	Forced Failover using Data Guard Broker	32
6.4	OPENING A STANDBY DATABASE READ ONLY	32
6.5	REAL TIME APPLY / REAL TIME QUERY	32
7	APPENDIX A - USING RMAN TO CREATE THE STANDBY DATABASE (TRADITIONAL METHOD	33
7.1	ASSUMPTIONS.....	33
7.2	RMAN BACKUP.....	33
7.2.1	New Backup	33
7.2.2	Existing Backup.....	34
7.3	CREATING THE STANDBY DATABASE	34
7.3.1	Prerequisites	34
7.3.2	Procedure	34
8	APPENDIX B - FURTHER READING	36
8.1	ORACLE MANUALS	36
8.2	METALINK	36

1 Introduction

1.1 Purpose

This document will describe an end-to-end process for creating a High Availability environment which utilises both Real Application Clusters (Oracle RAC) 11g and Data Guard 11g. This paper complements the standard documentation by the way that it steps through the whole process of building a complete environment where both the primary and physical standby systems are based on Oracle RAC. The described process has been tested and applied many times over by Oracle and customers in workshops or during consulting engagements and reflects the practical experiences gained from these.

1.2 Background

Organisations are using Oracle databases to store mission critical information. This information must be kept safe even in the event of a major disaster.

The traditional methods used to achieve this have centred on Oracle RAC and hardware-orientated solutions such as remote disk mirroring. Whilst Oracle RAC provides a high level of availability, RAC nodes are generally situated in the same computer room (with the exception so-called Extended Clusters). Oracle RAC clusters provide a high level of redundancy on the hardware layer to ensure that there are no single points of failure. That said, there are however still single points of failure in the architecture – the database alone (which is still single) or the whole computer room itself. If the room is destroyed or becomes isolated then it is likely that access to the cluster is impossible. If the database itself becomes unusable (partly or completely destroyed), redundancy on the server level doesn't help much.

This is where a disaster recovery plan comes into operation. This plan could consist of restoring the database to another machine. However, this process is likely to take a number of hours to complete. A better solution is to have a shadow database already restored to a separate machine, which is continuously being kept up to date. In the event of a disaster this database could be brought on line very quickly. This technology provided by Oracle is called Data Guard and the database itself is called a Standby database. While Data Guard is in fact the conceptual framework around two types of standby databases (physical and logical) the discussions in this document focus solely on the type of Physical Standby database.

1.3 Scope & Application

This document will examine the steps undertaken to create a Data Guard installation in general and in the context of Oracle RAC in particular. It will also cover the procedures, which need to be undertaken should one wish to switchover or failover from the primary site to the site where the standby database is located.

1.4 Related Documents

- Oracle Data Guard Concepts and Administration 11g Release 1 Sep 2007
- Oracle Data Guard Broker 11g Release 1 Sep 2007

1.5 Assumptions

The following assumptions have been made in this document:

- Network Connectivity between production and standby systems matches the requirements for the desired level of protection and for the transport of production amounts of database redo information
- The reader is familiar with the configuration of Oracle Net
- The reader is familiar with Recovery Manager (RMAN)
- The reader is familiar with the installation and configuration of Oracle Real Application Clusters

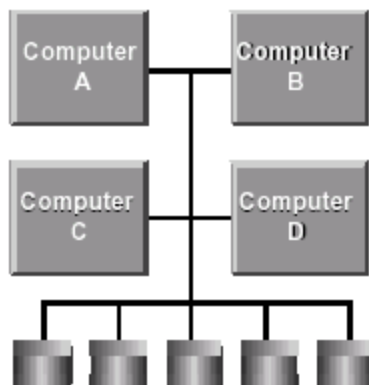
2 Architecture and Concepts

2.1 Glossary

Primary Site	This is where the users connect to access the production database.
Standby Site	This is where the standby database is maintained. It is also known as the disaster recovery (DR) site. Users will only connect to this site in the event of a failover or switchover.
Disaster	Non-availability of the primary site.

2.2 RAC Architecture

Oracle Real Application Clusters (Oracle RAC) consists of a number of separate computer systems joined together via a high speed interconnect, all of which have access to the same set of disks.



Oracle RAC creates a database on the shared disks using ASM, RAW partitions or Clustered Filesystems. The most common implementation in Oracle 11g is using ASM. Each cluster node runs an Oracle Instance that is used to interact with the database.

Users can connect to any one of the Oracle instances to run their applications.

2.3 Data Guard Architecture

Ideally the architecture of the node(s) located at the standby site will be the same as that of the primary, although this is not mandatory.

When planning the architecture of the standby system, the following will need to be considered (especially if that system does not have the same architecture as the primary system)

- If a failover is required (unplanned outage of the primary site), can the standby site handle the expected workload?
- If the standby site is going to be used whilst maintenance is being performed on the primary site (planned outage), can it handle the expected workload?

Assuming that capacities have been catered for, then the following is true:

It is not necessary for:

- the standby site to be clustered.
- the standby site to use RAW devices (unless it is a cluster itself).

NOTE: The requirement for having both sites (primary and standby) equipped with identical software versions and/or operating systems has been relaxed. Customers can now have flexible configurations. The status of what is currently allowed is reflected in

Metalink [Note 413484.1](#):

“Data Guard Support for Heterogeneous Primary and Standby Systems in Same Data Guard Configuration”

This note will always have the latest support information.

NOTE: If the standby system is a cluster, only one node of that cluster will be able to synchronize the standby database. The remaining nodes will stay idle until the database on the standby system is opened for normal (productive) operation. The Data Guard Broker, discussed later in this document, enables automatic failover of the “apply node” to a surviving node of the standby cluster in the event that the original apply node fails for whatever reason.

2.4 Network

The primary and standby sites will be connected together via a Network link. This network link must be reliable and have suitable bandwidth and latency.

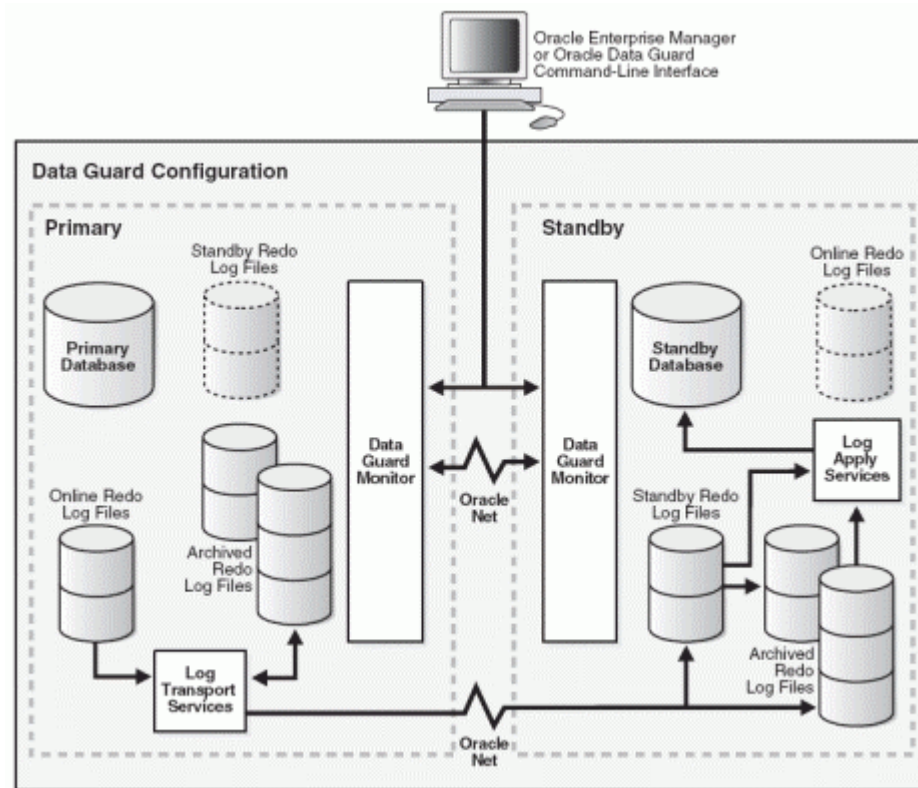
Data Guard provides 3 different modes:

- Maximum Protection
- Maximum Availability
- Maximum Performance

These are discussed later; however, the first 2 require synchronous writes to the standby site. Note that there is often a performance tradeoff using synchronous writes. While providing zero data loss protection, these modes may impact primary database performance if bandwidth is limited or if the round-trip network latency (RTT) between the primary and the standby database is high.

2.5 Data Guard Architecture

The Diagram below shows the Data Guard architectural components. These are explained in more detail below.



2.5.1 Log Transport Services

The Log Transport Services (aka Redo Transport Services) are designed to propagate changes from the primary database to the standby database in two ways – either by shipping archivelogs (ARCH) or by transmitting redo data continuously as it is processed by the Logwriter process (LNS).

2.5.2 Log Apply Services

Log Apply Services (aka 'Redo Apply' with Physical and 'SQL Apply' with Logical standby databases) are responsible for applying the redo information to the standby database from the archived or the standby redo log files.

2.5.3 Data Guard Broker

Data Guard Broker is the management and monitoring component that helps create, control, and monitor a primary database protected by one or more physical standby databases. Usage of the Broker is supported in RAC environments.

2.6 How it Works

As information is written to the primary database this information is also transmitted and applied to the standby database. These are basically two phases that are not directly coupled to each other.

2.6.1 Archived Log Shipping

In this scenario, the primary database instance(s) will generate archive logs. As soon as the archive logs are created they are shipped over to the standby site (by ARCH processes) where they will be applied to the database. This could be done immediately or after a configured delay.

2.6.2 Standby Redo logs

The preferred method of transferring redo information is utilizing the LNS process which transmits redo data as the Logwriter background process is flushing the redo buffer and writing to an online redo log file. This results in a continuous redo stream across the network (no peaks here). Best practices require that the receiving standby site have Standby redo Logs (SRLs) configured. The incoming redo is then written directly to these logs.

The number of SRLs should be equal to the sum of all online redo logs in the primary database + 1, e.g. 4 RAC instances each with 4 redo logs would give in 17 SRLs.

Whenever an entry gets written to an online redo log in any primary instance then that entry is simultaneously written to one of the standby redo logs. When a logswitch operation is performed on a primary online redo log, then a log switch will also occur on the standby database which means that the current SRL will be archived to a local directory on the standby system.

The feature 'Real Time Apply' was introduced in Oracle 10g. When Real Time Apply is used, redo is applied to the standby database as it is written to the standby redo log, rather than waiting for a log switch before applying the redo.

2.7 Levels of Protection

Data Guard can be configured to offer differing degrees of protection. These are summarised below:

2.7.1 Maximum Protection

This solution *guarantees* Zero Data Loss between the primary and the standby sites for multiple failure scenarios (e.g. network failure and primary site failure). If the last standby site becomes unavailable for any reason then the primary database is shutdown.

2.7.2 Maximum Availability

This solution provides zero data loss for single-failure scenarios. If however the standby site becomes unavailable (single failure), this protection mode places the emphasis on “availability” and work is allowed to continue on the primary database. If a second failure, for example the loss of the primary database, occurs during this period, then there will be unprotected transactions on the primary database that will be lost. Once available again, the standby database will automatically retrieve any data from the primary databases archive log files that have not been transferred and will resynchronize again without requiring manual intervention.

2.7.3 Maximum Performance

This solution works either by asynchronously shipping redo through the LNS process or by shipping archived redo logs through the ARCH process from the primary to the standby database as they are generated. This means that a degree of data loss can potentially be experienced if there is a failure, since there is never a guarantee that the primary and standby databases are in sync.

2.7.4 Pros and Cons

Maximum Protection and Maximum Availability require high specification network links between the primary and the standby sites. As data is written on the primary database this information is simultaneously written on the standby site. This means that the added benefit of zero data loss protection must be weighed against the potential impact to primary database performance should your network not have the required bandwidth or have high RTT latency typically found in WAN environments.

Maximum Performance has no impact on the primary database performance, but the asynchronous nature of redo transport can allow for data loss in case of a failover.

3 Prerequisites

3.1 Hardware

The standby site will ideally be hosted on hardware able to support the primary site's workload. Whilst this is not mandatory, it ensures that if a failover to the standby site is required, similar service levels will be attained.

3.2 Network

In order to facilitate a Maximum Protection/Availability standby site, the primary database will synchronously write redo log information to both the primary and standby site(s). It is therefore essential that the network link between these sites:

- Is reliable (no single point of failure).
- Has suitable bandwidth (depends on the amount of expected redo).
- Has very low latency.

3.3 Software

In previous Oracle releases both the primary and standby sites have been required to run the same version of the database software. This requirement has been substantially relaxed. For current support of mixed environments please see Metalink [Note 413484.1](#).

3.4 Real Application Clusters

3.4.1 ASM

If the primary database utilises ASM for disk storage it is strongly recommended to have ASM configured also on the standby site.

NOTE: ASM is the preferred storage management for Oracle RAC databases and is also being increasingly used for non-RAC databases as well.

3.4.2 Raw Devices

If the primary database is an Oracle RAC database and utilises raw devices and the standby database is a cluster database as well, then raw devices also need to be configured on the standby site.

If however the standby site is NOT a cluster, then there is no need for the standby database to use raw devices. In this case the following will make management simpler:

3.4.2.1 Non Raw Devices

1. Create a directory on the standby site to hold the data files.
2. Create the same directories on each of the primary RAC nodes.
3. On the primary RAC nodes create symbolic links from this directory to the raw devices.
4. When creating the RAC database use these symbolic links rather than the raw device names.

Advantages

Data Guard can automatically manage database files. I.e. when a file is added to the primary database it automatically gets also added to the standby database. If the directory structure is not exactly the same on both sites then a filename conversion has to be configured by using the init.ora-Parameters *db_file_name_convert* and *log_file_name_convert*.

E.g. `/u01/oradata/dbname/system.dbf` is the same regardless of the site.

This is also true if Oracle Managed Files (OMF) are being used.

4 Creating a Data Guard Environment

4.1 Assumptions

For the purpose of the instructions below the following have been assumed:

- Primary Host Names are **europa** and **callisto**
- Standby Host Names are **dione** and **hyperion**
- The primary database will be referred to as **MOON**
- Virtual Names are **europa-vip**, **callisto-vip**, **dione-vip** and **hyperion-vip**
- Both the primary and standby databases use ASM for storage
- The following ASM disk groups are being used **+DATA** (for data) and **+FRA** for Recovery/Flashback
- The standby database will be referred to as **SUN**
- Oracle Managed Files will be used.
- ORACLE_BASE is set to **/u01/app/oracle**

Where these names are used below they will be highlighted as above.

4.2 Procedure - Summary

The procedure to create a Data Guard environment is summarised below. Further sections go into detail about how to perform each of the following tasks:

1. Configure PRIMARY and STANDBY Sites
2. Install Oracle Software on each site.
3. Configure Oracle Networking on each site
4. Configure ASM on both sites
5. Configure listeners on each site
6. Configure Oracle Networking on each site.
7. Create initialisation files (Primary/Standby).
8. Duplicate the primary database to the standby site
9. Create a server parameter file for the standby site
10. Create extra Standby Control Files
11. Create Standby Redo Log files
12. Register standby database with cluster
13. Configure the Data Guard Broker
14. Place standby database into appropriate protection mode
15. Monitor.

4.3 Creating the RAC Standby Database

4.3.1 Configure Primary and Standby sites

To make management of the environment (and the configuration of Data Guard) simpler, it is recommended that the Primary and Standby machines have exactly the same structure, i.e.

- ORACLE_HOME points to the same mount point on both sites.
- ORACLE_BASE/admin points to the same mount point on both sites.
- ASM Disk Groups are the same on both sites

4.3.2 Install Oracle Software on each site.

The Oracle software will be installed from the Oracle Media on both sites. This will generally include:

- Oracle Clusterware
- Oracle database executables for use by ASM
- Oracle database executables for use by the RDBMS

4.3.3 Server Names / VIPs

In Oracle Real Application Clusters 11g virtual server names and IP addresses are used and maintained by Oracle Cluster Ready Services (CRS). Examples of a cluster naming is as follows:

Note: Both short and fully qualified names will exist.

Server Name/Alias/Host Entry	Purpose
europa.local	Public Host Name (PRIMARY Node 1)
callisto.local	Public Host Name (PRIMARY Node 2)
dione.local	Public Host Name (STANDBY Node 1)
hyperion.local	Public Host Name (STANDBY Node 2)
europa-vip.local	Public Virtual Name (PRIMARY Node 1)
callisto-vip.local	Public Virtual Name (PRIMARY Node 2)

dione-vip.local	Public Virtual Name (STANDBY Node 1)
hyperion-vip.local	Public Virtual Name (STANDBY Node 2)

4.3.4 Configure Oracle Networking

4.3.4.1 Configure Listener on Each Site

Each site will have a listener defined which will be running from the ASM Oracle Home. The following listeners have been defined in this example configuration.

Primary Role
Listener_europa
Listener_callisto
Listener_dione
Listener_hyperion

4.3.4.2 Static Registration

Oracle must be able to access all instances of both databases whether they are in an open, mounted or closed state. This means that these must be statically registered with the listener.

These entries will have a special name which will be used to facilitate the use of the Data Guard Broker, discussed later.

4.3.4.3 Sample Listener.ora

```

LISTENER_dione =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = dione-vip)(PORT = 1521)
          (IP = FIRST))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = dione)(PORT = 1521)
          (IP = FIRST))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC))
      )
    )
  )
SID_LIST_LISTENER_dione =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME=SUN_dgmgrl.local)
      (SID_NAME = SUN1)
      (ORACLE_HOME = $ORACLE_HOME)
    )
  )

```

4.3.4.4 Configure TNS entries on each site.

In order to make things simpler the same network service names will be generated on each site. These service names will be called:

Alias	Comments
MOON1_DGMGRL.local	Points to the MOON instance on europa using the service name MOON_DGMGRL.local. This can be used for creating the standby database.
MOON1.local	Points to the MOON instance on europa, using the service name MOON.local
MOON2.local	Points to the MOON instance on callisto using the service name MOON.local
MOON.local	Points to the MOON database i.e. Contains all database instances.
SUN1_DGMGRL.local	Points to the SUN instance on dione using the service name SUN1_DGMGRL ** This will be used for the database duplication.
SUN1.local	Points to the SUN instance on dione using the service name SUN.local
SUN2.local	Points to the SUN instance on hyperion using the service name SUN.local
SUN.local	Points to the SUN database i.e. Contains all the database instances
listener_DB_UNIQUE_NAME.local	This will be a tns alias entry consisting of two address lines. The first address line will be the address of the listener on Node1 and the second will be the address of the listener on Node 2. Placing both of the above listeners in the address list will ensure that the database automatically registers with both nodes. There must be two sets of entries. One for the standby nodes call listener_SUN and one for the primary nodes called listener_MOON

4.3.4.4.1 Sample tnsnames.ora (europa)

```

MOON1_DGMGRL.local =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = europa-vip)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = MOON_DGMGRL.local)
    )
  )

MOON1.local =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = europa-vip)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = MOON.local)
      (INSTANCE_NAME = MOON1)
    )
  )

MOON2.local =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = callisto-vip)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = MOON.local)
    )
  )

```

```

        (INSTANCE_NAME = MOON2)
    )
)

MOON.local =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = europa-vip)(PORT = 1521))
  (ADDRESS = (PROTOCOL = TCP)(HOST = callisto-vip)(PORT = 1521))
  (LOAD_BALANCE = yes)
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = MOON.local)
  )
)

SUN1_DGMGRL.local =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = dione-vip)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = SUN_DGMGRL.local)
  )
)

SUN2.local=
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = hyperion-vip)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = SUN.local)
    (INSTANCE_NAME=SUN2)
  )
)

SUN1.local=
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = dione-vip)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = SUN.local)
    (INSTANCE_NAME=SUN1)
  )
)

SUN.local=
(DESCRIPTION =
  (ADDRESS_LIST=
    (ADDRESS = (PROTOCOL = TCP)(HOST = dione-vip)(PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = hyperion-vip)(PORT = 1521)))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = SUN.local)
  )
)

LISTENERS_MOON.local=
(ADDRESS_LIST =
  (ADDRESS = (PROTOCOL = TCP)(HOST = europa-vip)(PORT = 1521))
  (ADDRESS = (PROTOCOL = TCP)(HOST = callisto-vip)(PORT = 1521))
)

```

4.3.5 Configure ASM on each Site

As this is an Oracle RAC database using ASM it is strongly recommended that ASM is also configured on the standby site before continuing. To keep things simple it is assumed that the disk groups created on the standby site have the same names as those on the primary.

4.3.6 Prepare Primary Database for Duplication.

Certain initialisation parameters are only applicable when a database is running in either a standby or primary database role. Defining ALL of the parameters on BOTH sites will ensure that, if the roles are switched (Primary becomes Standby and Standby becomes the new Primary), then no further configuration will be necessary.

Some of the parameters will however be node-specific; therefore there will be one set of parameters for the Primary site nodes and one for the Standby site nodes.

4.3.6.1 Primary Site Preparation

The following initialisation parameters should be set on the primary site prior to duplication. Whilst they are only applicable to the primary site, they will be equally configured on the standby site.

Dg_broker_config_file1	Point this to a file within the ASM disk group - Note File need not exist.
Dg_broker_config_file2	Point this to a file within the ASM disk group - Note File need not exist.
db_block_checksum	To enable datablock integrity checking (OPTIONAL)
db_block_checking	To enable datablock consistency checking (OPTIONAL)

As long as performance implications allow and do not violate existing SLAs it should be mandatory to have db_block_checksum and db_block_checking enabled.

Additionally, the following must also be configured:

Archive Log Mode

The primary database must be placed into archive log mode.

Forced Logging

The standby database is kept up to date by applying transactions on the standby site, which have been recorded in the online redo logs. In some environments that have not previously utilized Data Guard, the NOLOGGING option may have been utilized to enhance database performance. Usage of this feature in a Data Guard protected environment is strongly undesirable.

From Oracle version 9.2, Oracle introduced a method to prevent NOLOGGING transactions from occurring. This is known as *forced logging mode* of the database. To enable forced logging, issue the following command on the primary database:

```
alter database force logging;
```

Password File

The primary database must be configured to use an external password file. This is generally done at the time of installation. If not, then a password file can be created using the following command:

```
orapwd file=$ORACLE_HOME/dbs/orapwMOON1 password=myspasswd
```

Before issuing the command ensure that the ORACLE_SID is set to the appropriate instance – in this case MOON1.

Repeat this for each node of the cluster.

Also ensure that the initialisation parameter `remote_login_passwordfile` is set to 'exclusive'.

As with Oracle11.1 the Oracle Net sessions for Redo Transport can alternatively be authenticated through SSL (see also section 6.2.1 in the Data Guard Concepts manual).

4.3.6.2 Standby Site Preparation

Initialisation File

As part of the duplication process a temporary initialisation file will be used. For the purposes of this document this file will be called /tmp/initMOON.ora have one line:

```
db_name=MOON
```

Password File

The standby database must be configured to use a password file. This must be created by copying the password file from the primary site to the standby site and renaming it to reflect the standby instances.

Repeat this for each node of the cluster.

Additionally ensure that the initialisation parameter `remote_login_passwordfile` is set to exclusive.

Create Audit File Destination

Create a directory on each node of the standby system to hold audit files.

```
mkdir /u01/app/oracle/admin/SUN/adump
```

Start Standby Instance

Now that everything is in place the standby instance needs to be started ready for duplication to commence:

```
export ORACLE_SID=SUN1
sqlplus / as sysdba
startup nomount pfile='/tmp/initMOON.ora'
```

Test Connection

From the primary database test the connection to the standby database using the command:

```
sqlplus sys/myspasswd@SUN_dgmgr1 as sysdba
```

This should successfully connect.

4.3.7 Duplicate the Primary database.

The standby database is created from the primary database. In order to achieve this, up to Oracle10g a backup of the primary database needs to be made and transferred to the standby and restored. Oracle RMAN 11g simplifies this process by providing a new method which allows an 'on the fly'-duplicate to take place. This will be the method used here (the pre-11g method is described in the Appendices).

From the primary database invoke RMAN using the following command:

```
export ORACLE_SID=MOON1
rman target / auxiliary sys/mypasswd@SUN1_dgmg1
```

NOTE: If RMAN returns the error "rman: can't open target" then ensure that 'ORACLE_HOME/bin' appears first in the PATH because there exists a Linux utility also named RMAN.

Next, issue the following duplicate command:

```
duplicate target database for standby from active database
spfile
set db_unique_name='SUN'
set control_files='+DATA/SUN/controlfile/control01.dbf'
set instance_number='1'
set audit_file_dest='/u01/app/oracle/admin/SUN/adump'
set remote_listener='LISTENERS_SUN'
nofilenamecheck;
```

4.3.8 Create an SPFILE for the Standby Database

By default the RMAN duplicate command will have created an spfile for the instance located in \$ORACLE_HOME/dbs.

This file will contain entries that refer to the instance names on the primary database. As part of this creation process the database name is being changed to reflect the DB_UNIQUE_NAME for the standby database, and as such the spfile created is essentially worthless. A new spfile will now be created using the contents of the primary database's spfile.

4.3.8.1 Get location of the Control File

Before starting this process, note down the value of the control_files parameter from the currently running standby database.

4.3.8.2 Create a textual initialisation file

The first stage in the process requires that the primary databases initialisation parameters be dumped to a text file:

```
set ORACLE_SID=MOON1
```

```
sqlplus "/ as sysdba"
create pfile='/tmp/initSUN.ora' from spfile;
```

Copy the created file '/tmp/initSUN.ora' to the standby server.

4.3.8.3 Edit the init.ora

On the standby server, edit the /tmp/initSUN.ora file:

NOTE: Change every occurrence of **MOON** with **SUN** with the exception of the parameter **DB_NAME** which must **NOT** change.

Set the control_files parameter to reflect the value obtained in 4.3.8.1 above. This will most likely be +DATA/SUN/controlfile/control01.dbf.

Save the changes.

4.3.8.4 Create SPFILE

Having created the textual initialisation file it now needs to be converted to a spfile and stored within ASM by issuing:

```
export ORACLE_SID=SUN1
sqlplus "/ as sysdba"
create spfile='+DATA/SUN/spfileSUN.ora' from pfile=
'/tmp/initSUN.ora'
```

4.3.8.5 Create Pointer File

With the spfile now being in ASM, the RDBMS instances need to be told where to find it.

Create a file in the \$ORACLE_HOME/dbs directory of standby node 1 (dione) called initSUN1.ora . This file will contain one line:

```
spfile='+DATA/SUN/spfileSUN.ora'
```

Create a file in the \$ORACLE_HOME/dbs directory of standby node 2 (hyperion) called initSUN2.ora . This file will also contain one line:

```
spfile=' +DATA/SUN/spfileSUN.ora'
```

Additionally remove the RMAN created spfile from \$ORACLE_HOME/dbs located on standby node 1 (dione)

4.3.9 Create secondary control files

When the RMAN duplicate completed, it created a standby database with only one control file. This is not good practice, so the next step in the process is to create extra control files.

This is a two-stage process:

1. Shutdown and startup the database using nomount :

```
shutdown immediate;
```

```
startup nomount;
```

2. Change the value of the control_files parameter to '+DATA',' +FRA'

```
alter system set control_files='+DATA',' +FRA' scope=spfile;
```

3. Shutdown and startup the database again :

```
shutdown immediate;
```

```
startup nomount;
```

3. Use RMAN to duplicate the control file already present:

```
export ORACLE_SID=SUN1
```

```
rman target /
```

```
restore controlfile from '+DATA/SUN/controlfile/control01.dbf'
```

This will create a control file in both the ASM Disk group's +DATA and +FRA. It will also update the control file parameter in the spfile.

If you wish 3 to have control files simply update the control_files parameter to include the original controlfile as well as the ones just created.

4.3.10 Cluster-enable the Standby Database

The standby database now needs to be brought under clusterware control, i.e. registered with Cluster Ready Services.

Before commencing, check that it is possible to start the instance on the second standby node (hyperion):

```
export ORACLE_SID=SUN2
```

```
sqlplus "/ as sysdba"
```

```
startup mount;
```

NOTE: Resolve any issues before moving on to the next steps.

4.3.10.1 Ensure Server Side Load Balancing is configured

Check whether the init.ora parameter remote_listener is defined in the standby instances.

If the parameter is not present then create an entry in the tnsnames.ora files (of all standby nodes) which has the following format:

```
LISTENERS_SUN.local =  
  (DESCRIPTION =  
    (ADDRESS_LIST =  
      (ADDRESS = (PROTOCOL = TCP)(HOST = dione-vip.local)(PORT = 1521))  
      (ADDRESS = (PROTOCOL = TCP)(HOST = hyperion-vip.local)(PORT =  
1521))  
    )  
  )
```

Then set the value of the parameter `remote_listener` to `LISTENERS_SUN.local`.

4.3.10.2 Register the Database with CRS

Issue the following commands to register the database with Oracle Cluster Ready Services:

```
srvctl add database -d SUN -o $ORACLE_HOME -m local -p
"+DATA/SUN/spfileSUN.ora" -n MOON -r physical_standby -s mount
srvctl add instance -d SUN -i SUN1 -n dione
srvctl add instance -d SUN -i SUN2 -n hyperion
```

4.3.10.3 Test

Test that the above has worked by stopping any running standby instances and then starting the database (all instances) using the command:

```
srvctl start database -d SUN
```

Once started check that the associated instances are running by using the command:

```
srvctl status database -d SUN
```

4.3.11 Temporary Files

Temporary files associated with a temporary tablespace are automatically created with a standby database.

4.3.12 Create Standby Redo Logs

Standby Redo Logs (SRL) are used to store redo data from the primary databases when the transport is configured using the Logwriter (LGWR), which is the default.

Each standby redo log file must be at least as large as the largest redo log file in the primary database. It is recommended that all redo log files in the primary database and the standby redo logs in the respective standby database(s) be of the same size.

The recommended number of SRLs is :

$(\# \text{ of online redo logs per primary instance} + 1) * \# \text{ of instances}$.

Whilst standby redo logs are only used by the standby site, they should be defined on both the primary as well as the standby sites. This will ensure that if the two databases change their roles (primary-> standby and standby -> primary) then no extra configuration will be required.

The standby database must be mounted (mount as 'standby' is the default) before SRLs can be created.

SRLs are created as follows (the size given below is just an example and has to be adjusted to the current environment):

1. `sqlplus '/ a sysdba'`
2. `startup mount`

```
3. alter database add standby logfile SIZE 100M;
```

NOTE: Standby Redo Logs are also created in logfile groups. But be aware of the fact that group numbers then must be greater than the group numbers which are associated with the ORLs in the primary database. Wrt group numbering Oracle makes no difference between ORLs and SRLs.

NOTE: Standby Redo Logs need to be created on both databases.

The standby database is now created. The next stage in the process concerns enabling transaction synchronisation. There are two ways of doing this:

1. Using SQL Plus
2. Using the Data Guard Broker

4.4 Configuring Data Guard using SQL Plus

4.4.1 Introduction

This section of the document describes the process of setting up a physical standby database environment using SQLPlus and manually setting database initialisation parameters.

4.4.2 Configure the Standby Database

The following initialisation parameters need to be set on the standby database:

Parameter	Value (dione)	Value (hyperion)
db_unique_name	SUN	
db_block_checking	TRUE (OPTIONAL)	
db_block_checksum	TRUE (OPTIONAL)	
log_archive_config	dg_config=(MOON, SUN)	
log_archive_max_processes	5	
fal_client	SUN1.local	SUN2.local
fal_server	'MOON1.local', 'MOON2.local'	
Standby_file_management	Auto	
log_archive_dest_2	service=MOON LGWR SYNC AFFIRM db_unique_name=PRIMARY_MOON VALID_FOR=(ALL_LOGFILES,PRIMARY_ROLE)	

log_archive_dest_2 (Max. Performance Mode)	service=MOON ARCH db_unique_name=PRIMARY_MOON VALID_FOR=(ALL_LOGFILES,PRIMARY_ROLE)
---	---

4.4.3 Configure the Primary Database

The following initialisation parameters need to be set on the primary database:

Parameter	Value (europa)	Value (callisto)
db_unique_name	MOON	
db_block_checking	TRUE (OPTIONAL)	
db_block_checksum	TRUE (OPTIONAL)	
log_archive_config	dg_config=(MOON, SUN)	
log_archive_max_processes	5	
fal_client	MOON1.local	MOON2.local
fal_server	'SUN1.local', 'SUN2.local'	
standby_file_management	Auto	
Log_archive_dest_2	service=SUN LGWR SYNC AFFIRM db_unique_name=SUN VALID_FOR=(ALL_LOGFILES,PRIMARY_ROLE)	
Log_archive_dest_2 (Max. Performance Mode)	service=SUN ARCH db_unique_name=SUN VALID_FOR=(ALL_LOGFILES,PRIMARY_ROLE)	

4.4.4 Set the Protection Mode

In order to specify the protection mode, the primary database must be mounted but not opened.

NOTE: The database must be mounted in exclusive mode which effectively means that all RAC instances but one be shutdown and the remaining instance be started with a parameter setting of `cluster_database=false`.

Once this is the case then the following statement must be issued on the primary site:

If using Maximum Protection mode then use the command:

```
Alter database set standby database to maximize protection;
```

If using Maximum Availability mode then use the command:

```
Alter database set standby database to maximize availability;
```

If using Maximum Performance mode then use the command:

```
Alter database set standby database to maximize performance;
```

4.4.5 Enable Redo Transport & Redo Apply

Enabling the transport and application of redo to the standby database is achieved by the following:

4.4.5.1 Standby Site

The standby database needs to be placed into Managed Recovery mode. This is achieved by issuing the statement:

```
Alter database recover managed standby database disconnect;
```

Oracle 10gR2 introduced Real Time redo apply (SRLs required). Enabling real time apply is achieved by issuing the statement:

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

4.4.5.2 Primary Site:

Set:

```
log_archive_dest_state_2=enable
```

in the init.ora file or issue via SQLPlus :

```
alter system set log_archive_dest_state_2=enable
```

4.5 Configuring Data Guard using the Data Guard Broker

4.5.1 Introduction

The Data Guard Broker has a command line interface, which can be used to simplify management of whole Data Guard environments. When using the Broker, configuration information is stored within the Broker itself. When the Broker starts (enabled by a database initialisation parameter), it will use a series of ALTER SYSTEM statements to set up transaction synchronisation between the primary and standby sites. The parameters it sets are the same as those mentioned in the SQLPlus example earlier. **It is therefore imperative that database configuration changes are made only via the broker**, not by directly editing initialisation parameters. Failure to obey this rule will result in the Broker overwriting those values.

NOTE: If using Grid Control to manage a Data Guard environment the Broker must be configured.

4.5.2 Broker Configuration Files

The Data Guard Broker uses two files to hold its configuration information. By default these files are located in the \$ORACLE_HOME/dbs directory. In a RAC environment this is not appropriate as all database instances need to have access to the same configuration information.

Before continuing with the Broker configuration make sure that the Broker files are configured such that they point to shared storage (in this case ASM).

This can be checked by looking at the values of the parameters:

dg_broker_config_file1 and dg_broker_config_file2.

Appropriate values for these parameters can be found above (4.3.6.1).

4.5.3 Enabling the Broker

Before the Broker can be used it must first be enabled. This is done by changing the value of the database initialisation parameter dg_broker_start to true:

```
alter system set dg_broker_start=true;
```

NOTE: This needs to be performed on both the primary and standby site.

4.5.4 Creating a Broker Configuration

A Broker configuration is created either using Grid Control or the DGMGRL command line interface. This document uses the latter.

Start dgmgrl using the command

```
dgmgrl sys/mypasswd
```

** NOTE: Do not use "/" alone because this would cause problems later.

Enter the following to create the Data Guard configuration:

```
create configuration MOON_SUN as primary database is MOON
connect identifier is MOON.local;
```

```
add database SUN as connect identifier is SUN.local
maintained as physical;
```

4.5.5 Enable the Broker Configuration

Once the Broker configuration has been created then it needs to be enabled before it can be used. This is achieved using the following command:

```
enable configuration;
```

This will take some time to complete. Once it has completed issue the command:

```
show configuration;
```

If all is well the message 'SUCCESS' will be displayed.

If it isn't then the following log file needs to be checked and any issues resolved:

```
$ORACLE_BASE/diag/rdbms/<db_name>/<instance_name>/trace/  
drcINSTANCE_NAME.log
```

Another useful command is :

```
show database verbose DB_NAME
```

NOTE: This file appears on all nodes.

NOTE: Replace <db_name> with the value of DB_UNIQUE_NAME

NOTE: Replace <instance_name> with the value of ORACLE_SID.

NOTE: Secondary instances are not displayed until the configuration is enabled.

4.5.6 Broker Customisation

This will provide a basic configuration. Once the standby database starts, the Data Guard Broker will automatically put the standby database into managed recovery mode as detailed above.

This basic configuration is not enough however to sustain a complete environment. It must be further customised and this is done by setting Data Guard properties. The following properties should be defined:

```
dgmgrl sys/mypasswd
```

```
edit database MOON set property LogArchiveMaxProcesses=5;
```

```
edit database SUN set property LogArchiveMaxProcesses=5;
```

```
edit database MOON set property StandbyFileManagement=auto;
```

```
edit database SUN set property StandbyFileManagement=auto;
```

4.5.6.1 Maximum Availability/Protection

Additionally, if using Maximum Availability / Protection mode, the following values need to be set:

```
edit database MOON set property LogXptMode='SYNC';
```

```
edit database SUN set property LogXptMode='SYNC';
```

```
edit configuration                                     set  
protection mode as maxavailability | maxprotection;
```

5 Monitoring

5.1.1 Introduction

Once the configuration has been created, it is essential to check that everything is running smoothly. The following sections identify different ways of monitoring the environment.

5.1.2 Log Files

When an archive operation occurs it is entered into the alert log. Whenever a log switch occurs on the primary this will be registered in the alert log. When an archive log apply is performed on the standby database then this will be registered in the standby instance's alert log.

5.1.3 Fixed Views

The following fixed views can be used to monitor the Data Guard Broker:

Primary Site	
V\$ARCHIVE_DEST	Describes for the current instance all the archived redo log destinations, their current value, mode and status.
V\$ARCHIVE_DEST_STATUS	Displays runtime and configuration information for the archived redo log destinations.
V\$ARCHIVED_LOG	Displays archived redo log information from the control file, including archived log names.
V\$DATABASE	Provides database information from the control file. Including the status of the database.
V\$LOG	The view contains log file information from the online redo logs.
Standby Site	
V\$ARCHIVED_LOG	Displays archived redo log information from the control file, including archived log names.
V\$DATABASE	Provides database information from the control file, including the status of the database.
V\$LOGFILE	Contains information about the online/standby redo logs.
V\$MANAGED_STANDBY	Displays current and status information for some Oracle database server processes related to Data Guard.
V\$STANDBY_LOG	Displays information about the standby redo logs.

6 Management

6.1 Switchover

Upon occasion the primary site will become unavailable due to planned outages such as maintenance on the primary server. In this scenario it is advantageous to temporarily switchover to the standby site. This procedure is non-destructive and can be performed in reverse when the primary site becomes available, without having to rebuild either database.

6.1.1 Switchover using SQL Plus

Verify that each database is properly configured for the role it is about to assume and the standby database is in mounted state.

Before performing a switchover from a RAC primary shut down all but one primary instance (they can be restarted after the switchover has completed).

Before performing a switchover or a failover to a RAC standby shut down all but one standby instance (they can be restarted after the role transition has completed).

The following SQL statements perform the switchover to the standby (Note: if recovery is required follow the steps in section 7.1 Applying redo log changes).

On the primary database initiate the switchover:

```
alter database commit to switchover to physical standby
[with session shutdown];
```

NOTE: Use the 'with session shutdown' clause when connections to the primary exist.

Restart and mount the new standby database.

On the (old) standby database switch to new primary role:

```
alter database commit to switchover to primary;
```

Restart and open the new primary database.

Start Redo Apply on the new standby database:

```
alter database recover managed standby database
[using current logfile] disconnect;
```

NOTE: It is in general good practice to set `log_archive_dest_state_2` to 'defer' if the primary site is going to be unavailable for a longer period of time.

6.1.2 Switchover using Data Guard Broker

Switching over to the standby database in this scenario is accomplished by entering the following commands into the Broker's command line interface (this can also be achieved from Grid Control) :

```
dgmgrl sys/mypasswd
switchover to SUN;
```

** Substitute <SUN> with the name of the database you wish to switch over to.

NOTE: If the primary site is going to be unavailable for a longer period of time disable the database :

```
disable database MOON;
```

6.2 Failover

6.2.1 Failover using SQL Plus

If the primary database is not available because of a disaster then a failover will be required to the standby site.

Before opening the standby database for normal operation all outstanding changes need to be applied to the database. In order to facilitate this the following steps need to be accomplished:

Terminate managed recovery mode by

```
alter database recover managed standby database finish;
```

Switch the standby database to primary role and open it in normal mode:

```
alter database commit to switchover to primary;
alter database open;
```

6.2.2 Failover using Data Guard Broker

The same can be accomplished using the Data Guard Broker. The command is:

```
dgmgrl sys/mypasswd
failover to SUN;
```

6.3 Forced failover

In certain circumstances it may not be possible to perform a *standard* failover. If this is the case, a *forced* failover can be performed. A forced failover is destructive – once it is invoked, any remaining standby databases will have to be rebuilt.

If using SRLs then issue:

```
alter database recover managed standby database finish force;
```

Then follow these steps:

```
alter database activate standby database [skip standby logfile];
```

```
shutdown immediate;
startup mount;
alter database open resetlogs;
```

6.3.1 Forced Failover using Data Guard Broker

The same can be accomplished using the Data Guard Broker :

```
dgmgctl sys/mypasswd
failover to SUN immediate;
```

6.4 Opening a Standby Database Read Only

The standby database can be opened in a read only mode to run reports or to perform checks on the data. This is done in the following manner:

1. Stop managed recovery

```
alter database recover managed standby database cancel;
```
2. Open the database

```
alter database open read only;
```

Alternatively, performing a 'normal' startup on a standby database will open it in read-only mode.

NOTE: Whilst open in read only mode, changes are being received but not applied to the standby database.

6.5 Real Time Apply / Real Time Query

Oracle 11g introduces the ability of Data Guard to be open read-only and yet still be applying redo changes. This is known as Real Time Apply/Real Time Query. It is enabled via SQL Plus in the following manner:

1. Connect to the standby database:

```
sqlplus / as sysdba
```
2. Cancel Managed Recovery

```
alter database recover managed standby database cancel;
```
3. Open the database

```
alter database open;
```
4. Restart Managed Recovery

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

7 Appendix A - Using RMAN to create the Standby Database (Traditional Method)

Prior to Oracle 11g it was not possible to perform an “on the fly” standby database creation. The method below was widely used and can still be used with 11g if desired.

7.1 Assumptions

The process can be more complex and flexible, however the procedure below makes the following assumptions:

- The standby database structure is the same as the primary database.
- A Recovery Catalog exists.

NOTE: setting up a standby database as described here does not require usage of a Recovery Catalog

NOTE: Using RMAN in a running Data Guard environment generally does require a Recovery C.atalog.

7.2 RMAN Backup

There are two methods of creating an RMAN backup which is suitable for the creation of a standby database.

7.2.1 New Backup

If the database has never been backed up before using RMAN then a backup needs to be made:

From the primary site :

```
rman target /

Run {
  allocate channel d1 device type disk format '/backup/%U';
  allocate channel d2 device type disk format '/backup/%U';

  sql "alter system switch logfile"
  backup database include current controlfile for standby plus
  archivelog;
}
```

In RAC implementations issue:

```
Run {
  allocate channel d1 device type disk format '/backup/%U';
  allocate channel d2 device type disk format '/backup/%U';

  sql "alter system archive log current"
  backup database include current controlfile for standby;
  sql "alter system archive log current";
  backup archivelog all;
}
```

7.2.2 Existing Backup

If the database has previously been backed up then a standby control file needs to be created, this is done by:

```
rman target /
run {
    allocate channel d1 device type disk format '/backup/%U';
    allocate channel d2 device type disk format '/backup/%U';

    backup current controlfile for standby;
    sql 'alter system archive log current;';
    backup archive log all not backed up 1 times;
}
```

7.3 Creating the Standby Database

7.3.1 Prerequisites

Before building the standby database, ensure that:

- Oracle executables are installed on the standby machine
- Database directory structure exists on the standby machine
- Ensure that the listener is started on the standby machine
- Ensure that the standby database is statically registered with the listener
- Ensure that a tnsnames.ora entry exists on the primary, which points to the standby database
- Ensure that the primary initialisation parameter file is available on the standby host
- Ensure that remote_password_file is set to none
- Ensure that the standby instances have an external password file created
- Ensure that RMAN backup sets are available to the standby host

7.3.2 Procedure

1. Startup the standby database instance:

```
startup nomount pfile=init.ora
```

2. Connect to RMAN (On the primary site)

```
rman target / auxiliary sys/syspwd@<standby_alias>
```

3. Create the standby database:

```
duplicate target database for standby dorecover
[nofilenamecheck];
```

The standby database will now be created.

NOTE: It is not necessary to recover standby database before putting it into operation since it will be synchronised through its capability of Redo Log Gap Resolution when the managed recovery is started. So the 'dorecover' option might not be used.

8 Appendix B – Further reading

8.1 Oracle Manuals

- **Oracle Clusterware Installation Guide**
11g Release 1 for Linux
Oracle Corporation Part No. B28263-03 October 2007
- **Oracle Real Application Clusters Installation Guide**
11g Release 1 for Linux
Oracle Corporation Part No. B28264-02 September 2007
- **Oracle Data Guard Concepts and Administration**
11g Release 1
Oracle Corporation Part No. B28294-02 September 2007
- **Oracle Data Guard Broker**
11g Release 1
Oracle Corporation Part No. B28295-02 September 2007
- **Oracle Database Backup and Recovery User's Guide**
11g Release 1
Oracle Corporation Part No. B28270-02 September 2007
- **Oracle Database Backup and Recovery Reference**
11g Release 1
Oracle Corporation Part No. B28273-02 September 2007

8.2 Metalink

- Note 413484.1

Data Guard Support for Heterogeneous Primary and Standby Systems in the Same Data Guard Configuration

Data Guard 11g Installation and Configuration on Oracle RAC Systems
October, 2008

Author: Michael Rhys, Oracle

Contributing Author: Holger Kalinowski, Oracle

Oracle USA, Inc.
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Copyright © 2008, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.