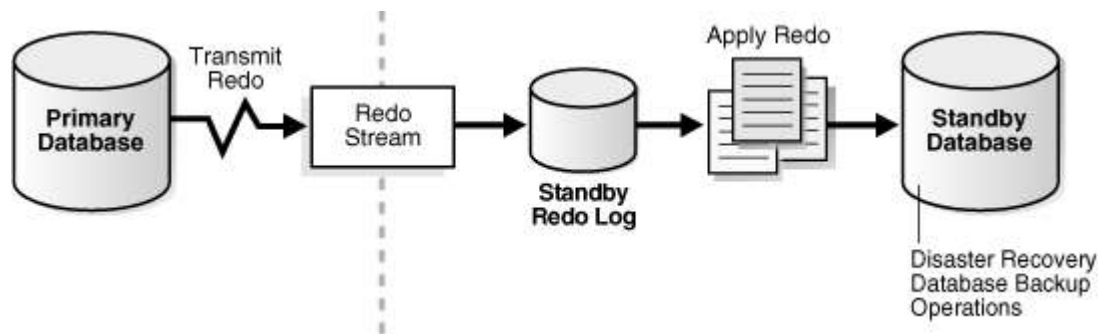


# Oracle Data Guard Overview

<http://andrewfraserdba.com/2011/08/09/oracle-data-guard-overview/>

Andrew Fraser, 15<sup>th</sup> August 2011

Oracle Data Guard is Oracle's main solution for high availability in the event of a disaster. This overview is for Oracle 11gR2.



Data Guard is controlled with either the 'dgmgrl' command line utility – to run, type 'dgmgrl' at the unix or windows command prompt – or through Oracle Enterprise Manager Grid Control. Version 8i lacked 'dgmgrl' and were controlled through alter system commands – those commands remain valid, but should now be avoided in favour of the superior dgmgrl utility.

This uses a Data Guard Broker background process, seen in the process list as 'dmon'.

Data Guard can also be referred to as 'log shipping' (it's name in version 8i) or 'standby' (because it has a standby database).

Data Guard advantages over RAC Real Applications Clusters:

1. It uses separate disk storage.  
So a failure at one SAN doesn't stop both nodes. RAC on its own isn't adequate for this, you either need RAC+DG or RAC+SAN replication.  
Data Guard is superior to SAN replication because it allows for automated failover, and also because it tends to be organisationally under the control of DBAs.
2. It can run at different sites at unlimited distances from each other.  
RAC clusters have to be physically close together (a few km at most).
3. RAC licensing is around 50% more expensive than Data Guard licensing.  
RAC licensing is an expensive extra cost option.

Data Guard is not an extra cost option, but you do have to pay for a full oracle license at the standby node.

Data Guard disadvantages compared to RAC:

1. It is active-passive.

So one server is idle, wasting hardware.

The best that Data Guard can do is have the idle server used for read only reporting.

### Data Guard Modes

In old versions (9i and earlier), the standby would lag half an hour or so behind primary, driven by on line redo log switch times through the 'arch' process. Not any more with 10g and later, where the 'lgwr' process performs redo transport in real time.

There are three data guard modes:

1. Maximum Availability – (synchronous) commit's try to wait for acknowledgement from standby (but they don't wait if the standby is completely down)
2. Maximum Performance – asynchronous propagation of redo, so standby can be a few seconds/minutes behind primary.
3. Maximum Protection – (synchronous) commit's wait for acknowledgement from standby. The live database shuts down if at least one standby isn't accessible – so the recommendation if using this is to have multiple standbys set up.

You switch between modes with:

```
DGMGRL> edit configuration set protection mode as maxavailability;
```

### Switchover and Failover

Switchover is an orderly role reversal between primary and standby databases. Data Guard continues to transport redo from the new primary database to the new standby database. You can switch back and forward multiple times without extra work or preparation. It is often used for testing DR, or for planned outages to the primary server.

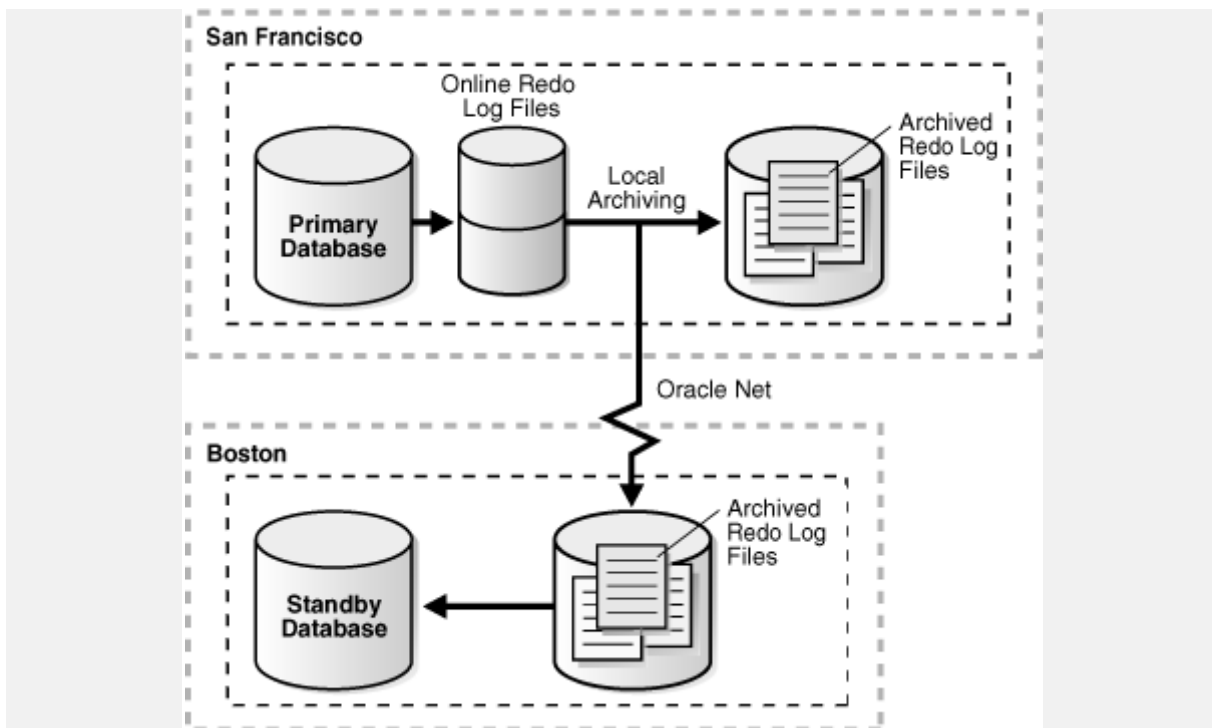
Failover is used when switchover is not possible – typically during a disaster so severe that the primary database is unreachable. Failover makes the standby database become the new primary database, but redo transport stops, there is no new standby database. After the disaster is over, a new standby database has to be manually recreated by a DBA using a backup and restore of the primary database.

Commands for these are:

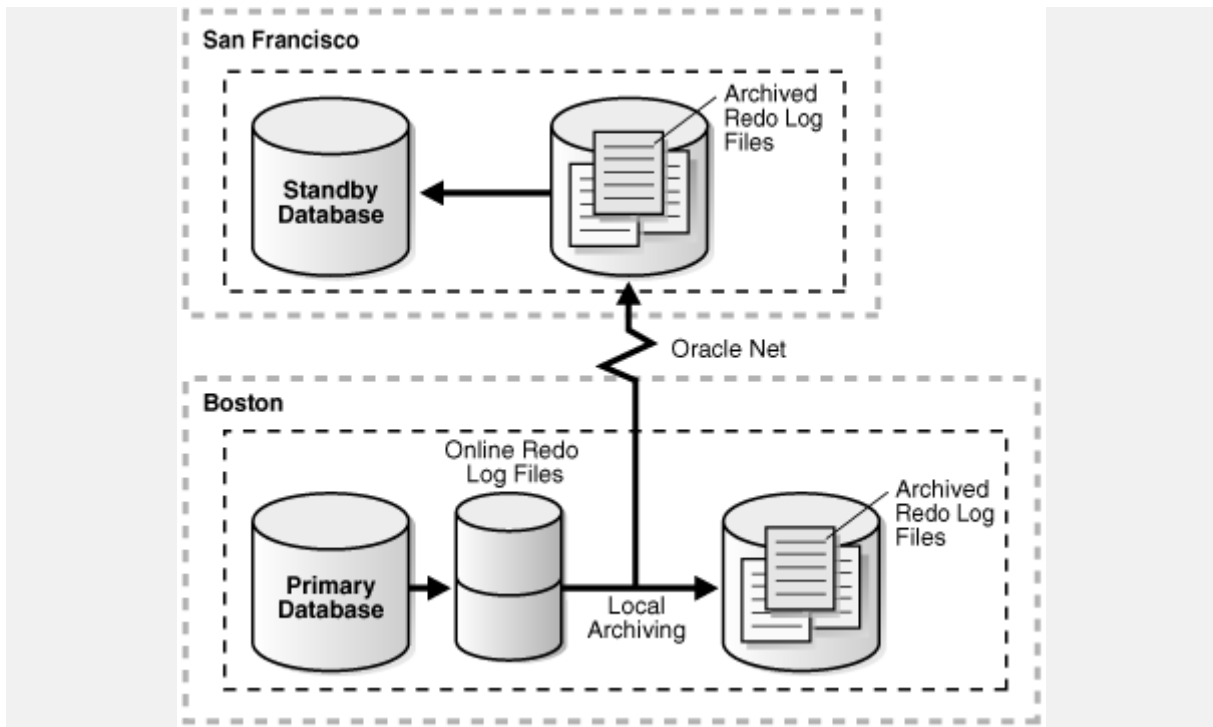
```
DGMGRL> switchover to mydb_sby ;
```

```
DGMGRL> failover to mydb_sby [immediate] ;
```

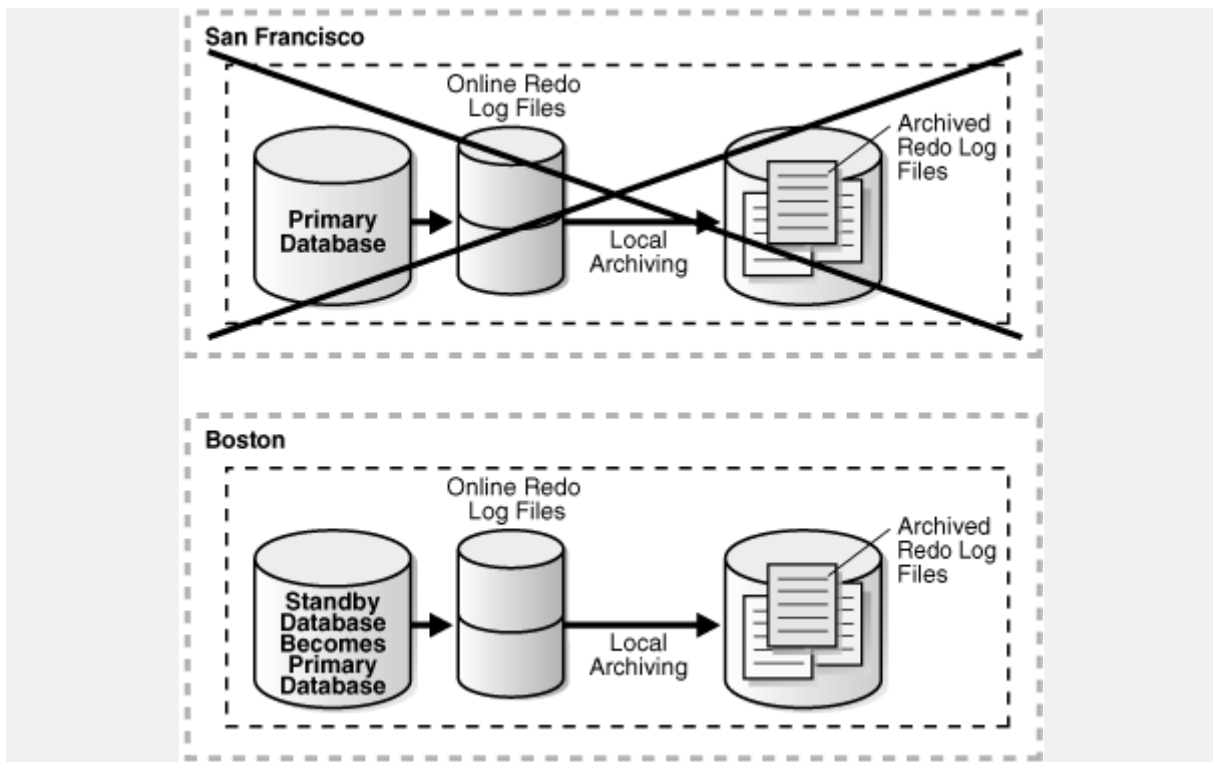
(Substitute 'mydb\_sby' with the service name of the standby database service as stated in "DGMGRL> show configuration" – normally set to be the same as its tnsnames connect identifier.)



1) Before Switchover



2) After Switchover



3) After Failover

**Tnsnames for Failover**

Often missed – but there is no point having DR databases if your users cannot connect to them when failover or switchover is done. One reason to regularly perform a switchover is to test that users all connect ok.

This requires all tnsnames.ora files, in PCs and in servers, to specify both primary and standby host addresses plus “(failover=on)”:

```
MYDB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST =
primaryserver.mydomain.com) (PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP) (HOST =
standbyserver.mydomain.com) (PORT = 1521))
    (FAILOVER = ON)
    (CONNECT_DATA =
      (SERVICE_NAME = mydb.mydomain.com))
```

Alternatively, you can change DNS entry each time switchover or failover happens – but the drawback of that is that it is often outwith the control of the DBA team, making its automation organisationally difficult.

### Fast Start Failover

You can let Oracle automatically initiate failover when it detects a problem with the primary database.

```
DGMGRL> enable fast_start failover ;
```

This uses a Fast Start Failover Observer process to check for problems. That should either be located on a third server, or on the DR standby server – because you want this observer to be functioning even (especially) if the primary server is dead.

```
DGMGRL> start observer ;
```

You can see current configuration with:

```
DGMGRL> show fast_start failover ;
```

```
DGMGRL> show configuration [verbose] ;
```

### Making use of a standby database – Snapshot Standby

This is very useful for reporting purposes.

Run this command:

```
DGMGRL> convert database 'mydb_sby' to snapshot standby ;
```

to open up your standby database to users.

This is not just opened read only – updates etc. are allowed, although they are temporary, they will be overwritten when you convert the database back to being a physical standby database. But that is useful for materialized views, data marts, temporary tables.

Snapshot standby databases continue to receive redo from the primary database, but do not apply it – it just queues up on the standby server – until they are reverted back to being a physical standby database with:

```
DGMGRL> convert database 'mydb_sby' to physical standby ;
```

Typically databases are switched back into physical standby overnight.

### **Making use of a standby database – Active Data Guard**

Active Data Guard allows the physical standby database to be open to users while still applying redo from the primary database. The standby database is restricted to read only operations.

This is an extra cost option for the Oracle license, similar to RAC, partitioning, etc. It costs around an extra 20% on top of the normal Enterprise Edition license fee.

### **Logical Standby instead of Physical Standby**

Logical Standby behaves similarly to physical standby, except it uses a more complex method of applying SQL statements (logical) instead of block level redo changes (physical). It is rarely used.

Possible reasons for using logical standby are:

- It can be used for rolling database upgrades (e.g. upgrade primary database first, standby database later).
- Saves extra license cost of Active Data Guard option if you want the standby database to be open read only while still keeping up to date with the primary database (logical standby databases do this by default without extra cost options).

See [Tom Kyte on this question](#).

### **Limitations of Data Guard**

- Nologging or unrecoverable operations are not applied to standby – so be cautious in using those options. Alarming data block corruption error messages are reported if a user attempts to access any nologging or unrecoverable data blocks.
- ‘Drop tablespace’, ‘drop datafile’, ‘rename datafile’, and add/drop online redo log files operations need done on both nodes manually.
- Initialization parameter changes need done on both nodes manually.
- For databases upgrades, the software in the new ORACLE\_HOME must be separately installed on both nodes. The database part of the upgrade is carried out only on the primary database.  
Often Data Guard replication is suspended for a significant period of time during and following the primary database upgrade. That is done with *DGMGRL> edit database 'mydb\_sby' set state='APPLY-OFF' ;*. The standby server is only upgraded, and replication resumed, once the primary upgrade has been determined to be a success.
- For patches, the software/binaries in the ORACLE\_HOME must be separately patched on both nodes. The database part of the patch (if any) is carried out only on the primary database.

### SQL Server Equivalents

Microsoft SQL Server’s **Database Mirroring** is their equivalent of Oracle Data Guard. Like Data Guard, Database Mirroring can run either synchronous and asynchronous. Unlike Oracle, Database Mirroring comes with Standard Edition – although only for synchronous database mirroring, the high performance asynchronous database mirroring does require Enterprise Edition.

Microsoft do make **Log Shipping** available with standard edition, which allows for asynchronous transaction log shipping to standby, but at set time intervals (such as every 30 minutes) rather than in real time/near real time.

### Licensing

A full license is required for the standby server, irrespective of whether or not it is also used for e.g. reporting purposes as well as DR. See [Oracle’s Licensing Data Recovery Environments](#).

Oracle Data Guard is included in Enterprise Edition at no extra cost. Active Data Guard (see above) is an extra cost option.

Standard Edition does not come with Oracle Data Guard, but for this you could use log shipping controlled through your own cron scripts as a cheap alternative.

### **Further Reading**

- [Kathy Rich and others: Oracle Data Guard Broker, 11g Release 2 \(11.2\)](#)
- [Kathy Rich and others: Oracle Data Guard Concepts and Administration, 11g Release 2 \(11.2\)](#)
- [Ritesh Chhajer: How to setup DGMGRL, broker with example](#)
- [Ritesh Chhajer: How to switchover using DGMGRL, broker with example](#)
- [Gavin Soorma: 11g Data Guard Broker DGMGRL Configuration quick steps](#)
- [Oracle FAQ: Data Guard implementation for Oracle 10gR2](#)