

Performing Database Failover with Oracle 11g Data Guard

By [Jim Czuprynski](#)

Synopsis. Data Guard's capability to recover a mission-critical database upon the loss of the entire production site is at the heart of Oracle 11g's disaster recovery features. This article - the sixth in this ongoing series - explores how to manually fail over a production database to its corresponding physical standby database, as well as reinstate a "failed" primary database to a physical standby.

The [prior article](#) in this series demonstrated:

- How to leverage Oracle 11g Data Guard *Real Time Query* features
- How Oracle 11g Data Guard *snapshot standby database* features simultaneously provide disaster recovery and quality assurance testing

This article will show how to:

- *Manually initiate* a failover operation when the primary database is no longer accessible
- *Reinstate* a failed primary database by transforming it back into a physical standby database

Pondering the Unthinkable: Failover

I don't think I've ever met an Oracle DBA who's been thrilled to discuss the possible loss of her complete production database site, even if she had simulated that loss and practiced the required steps to full database recovery. But let's face it: Even if a DBA follows Oracle's recommended best practices precisely to guard against data loss, there's always a miniscule chance that the production database site could become untenable through no fault of our own.

Thankfully, once an Oracle 11g Data Guard standby environment has been enabled, Oracle 11g offers excellent survivability of a production database while simultaneously insuring that any data loss is limited depending on the data protection mode selected.

Related Articles

- [Using Oracle 11g's Active Data Guard and Snapshot Standby Features](#)
- [Leveraging Oracle 11g Data Guard for Database Backup and Recovery](#)

It Can't Hurt To Ask... Though it might seem to be inappropriate at the time, the first question that I'll typically ask in an apparent failover situation is "Do we really need to perform a failover in this situation?" There may be a way around the apparent loss of the primary database, or it's possible that the apparent disaster is really not that disastrous. For example, if the production database is unavailable for only a few moments, or maybe even an hour, during off-peak business hours, is it *really* a problem, or just a matter of perception and extra paperwork? The reason I'm not afraid to challenge the necessity of failover is that there are some potentially serious consequences of completing the failover operation:

- *The original production database is "toast."* The primary database essentially becomes just a bunch of servers and disk drives at this point, and it may not be restorable to physical standby state until a full backup can be obtained using the new primary database as its source.
- *The physical standby becomes the primary.* Unless there is a second physical standby already configured, I now have a single point of failure in my production system, and my top priority must be re-establishment of the primary-standby relationship.
- *The physical standby may be underpowered for peak business hours.* Let's be honest: Many organizations look upon a physical standby environment as a necessary evil in terms of hardware and licensing costs, and my new primary database

may soon become overwhelmed by peak-time transaction volumes.

Detecting a Failover Situation. Once I've established that it's truly necessary to continue the failover operation, I'll use *Data Guard Broker* (DGB) to assist in the transition. To illustrate how DGB detects a failover situation, I'll start with the **ORCL_PRIMARY** and **ORCL_STDBY1** databases both fulfilling their initial intended respective roles of primary database and counterpart physical standby database. The **DGMGRL** utility reflects this as well, and it shows that everything is copasetic:

```
[oracle@11gStdbby ~]$ dgmgrl
DGMGRL for Linux: Version 11.1.0.6.0 - Production

Copyright (c) 2000, 2005, Oracle. All rights reserved.

Welcome to DGMGRL, type "help" for information.
DGMGRL> connect sys/oracle
Connected.
DGMGRL> show configuration

Configuration
Name:          MAA_orcl
Enabled:       YES
Protection Mode:  MaxPerformance
Databases:
  orcl_primary - Primary database
  orcl_stdby1 - Physical standby database

Fast-Start Failover: DISABLED

Current status for "MAA_orcl":
SUCCESS
```

To simulate the necessity for a failover to the physical standby database, I'll simply issue the **kill -9 <pid>** command against the *Server Monitor* (**SMON**) background process for the primary

database. The primary database's alert log detects this almost immediately, of course:

```
. . .
Tue Aug 25 18:54:10 2009
Errors in file
/u01/app/oracle/diag/rdbms/orcl_primary/orcl_primary/trace/orcl_pri
mary_pmon_6166.trc:
ORA-00474: SMON process terminated with error
PMON (ospid: 6166): terminating the instance due to error 474
Instance terminated by PMON, pid = 6166
. . .
```

The corresponding physical standby database's *Remote File Server* (**RFS**) background process also detects the loss of connectivity to the primary database, as reflected in that database's alert log:

```
. . .
Tue Aug 25 18:54:49 2009
RFS[2]: Possible network disconnect with primary database
Tue Aug 25 18:54:49 2009
RFS[1]: Possible network disconnect with primary database
Tue Aug 25 18:55:49 2009
. . .
```

To ascertain what's wrong here, I'll immediately verify the Data Guard configuration by connecting via **DGMGRL** and run the **SHOW CONFIGURATION** and **SHOW DATABASE** commands:

```
DGMGRL> show configuration verbose;

Configuration
Name:          MAA_orcl
Enabled:       YES
Protection Mode:  MaxPerformance
Databases:
  orcl_primary - Primary database
  orcl_stdbyl1 - Physical standby database
```

Fast-Start Failover: DISABLED

Current status for "MAA_orcl":

Error: ORA-16625: cannot reach the database

Failing Over to the Physical Standby. It's obvious that something is seriously wrong here! Once I've asked my usual probing questions and determined that the failover is indeed necessary, I'll start the failover process with one simple DGB command, **FAILOVER TO <standby database>**:

```
DGMGRL> failover to orcl_stdbby1;
```

```
Performing failover NOW, please wait...
```

```
Failover succeeded, new primary is "orcl_stdbby1"
```

```
DGMGRL> show configuration
```

```
Configuration
```

```
Name:          MAA_orcl
```

```
Enabled:       YES
```

```
Protection Mode:  MaxPerformance
```

```
Databases:
```

```
  orcl_stdbby1 - Primary database
```

```
  orcl_primary - Physical standby database (disabled)
```

```
Fast-Start Failover: DISABLED
```

```
Current status for "MAA_orcl":
```

```
SUCCESS
```

```
DGMGRL> show database orcl_stdbby1
```

```
Database
```

```
Name:          orcl_stdbby1
```

```
Role:          PRIMARY
```

```
Enabled:       YES
```

```
Intended State: TRANSPORT-ON
```

```
Instance(s):
  orcl_stdb1

Current status for "orcl_stdb1":
SUCCESS
```

During the failover to the physical standby database, the Oracle 11g DGB performs the following steps:

- First, it validates that the target standby database is *ready to accept the primary role*. If it's not, DGB will not allow the failover to continue until the DBA has manually resolved any discrepancies.
- It then *halts the Redo Apply process (MRPO)* on the target standby database once all unapplied redo has been applied successfully.
- It *completes the failover* to the physical standby database by opening the standby in READ WRITE MODE.
- Finally, DGB determines if there are any other physical standby databases that need to be re-enabled; it then restarts redo transport from the new primary database to any extant physical standby databases.

The corresponding alert log entries from the original standby database (now the primary database) are displayed in [Listing 6.1](#).

Have Any Data Been Lost? As with many questions we're asked as DBAs, the answer to this one is "It depends." In this case, it's going to depend on which *data protection mode* - either *Maximum Performance*, *Maximum Availability*, or *Maximum Protection* - that I've chosen for the Data Guard configuration. (See my [prior article](#) on this topic for a comparison on these data protection options and their corresponding effect on data loss.)

However, there's one other factor to consider in a failover situation. The Data Guard **FAILOVER TO <standby database>** command also offers an optional directive, **IMMEDIATE**, which instructs DGB

to an *immediate* failover instead of a *complete* failover, and here are the implications:

- During a **COMPLETE** failover (the default), Oracle 11g will attempt to recover the maximum amount of redo data based on the DG configuration's specified protection mode. After a complete failover, DGB *avoids disabling any standby databases* that are not the failover target. For obvious reasons, Oracle recommends using complete failover.
- During an **IMMEDIATE** failover, on the other hand, Oracle 11g applies no additional redo data to the standby database *after the failover is invoked*. While it's the fastest type of failover, it does require that the original primary database as well as all standby databases that aren't targets of the failover *must be re-enabled*.

Failing Over To a Logical Standby Database: Considerations. Though it's definitely possible, I'd like to avoid using a logical standby database as a failover target:

- A logical standby doesn't use Redo Apply to apply changes to data; instead, it uses *SQL Apply* just as Oracle Streams does during basic database replication. There are intrinsic limitations on which data types can accept DML for changed data, and thus *not all the data in the primary database may be present in the logical standby database*.
- SQL Apply can be configured to *accept only selected subsets* of logical change records, so even if all of a table's data types are completely supported, *not all of the changes on the primary database may have been applied on the logical standby database*.
- Perhaps most importantly, once a failover to a logical standby database is complete, *all other standby databases will be permanently disabled* and must be recreated manually.

I'll explore these limitations (but also highlight the features) of logical standby databases in an upcoming article in this ongoing series.

Reinstatement: Post-Failover Stress Relief

Prior to Oracle Database Release 10g, once the primary database had failed, it was impossible to bring the original primary database back into the Data Guard configuration as a new physical standby database without having to rebuild it completely. This was usually accomplished by using the new primary database as the target database and the original primary as the auxiliary database within the bounds of the **DUPLICATE DATABASE** Recovery Manager (RMAN) command set.

As long as certain prerequisites are in place, however, Oracle 10g offered the ability to reinstate the original primary database as a physical standby database:

- *Flashback Logging* must have been enabled on the original primary database. (Since this is a prerequisite for activating a physical standby database as a *snapshot standby* database in Oracle 11g, this requirement is usually already in place if I'm following Oracle 11g's best practice recommendations.)
- Sufficient Flashback Logs must be present on the original primary database server to allow *Flashback Database* to rewind the database to a state prior to its failure.
- Finally, the original primary database must be able to communicate over the network to the new primary database and Data Guard Broker so that it can be *restored* properly, Redo Apply can be *re-established*, and it can be *resynchronized* with the Data Guard configuration.

Reinstating a Failed Primary Database. To reinstate my failed **ORCL_PRIMARY** database as a new physical standby database, the original primary database has to be shut down and then brought back to **MOUNT** state. Otherwise, the initial attempt at reinstatement will fail as shown below in this **DGMGRL** session:

```
DGMGRL> reinstat database orcl_primary  
Reinstating database "orcl_primary", please wait...  
Error: ORA-16653: failed to reinstate database
```

Failed.

Reinstatement of database "orcl_primary" failed

After connecting to the **ORCL_PRIMARY** database via SQL*Plus and issuing the **STARTUP MOUNT** command, I'll again attempt to reinstate the database:

```
DGMGRL> reinstat database orcl_primary
```

```
Reinstating database "orcl_primary", please wait...
```

```
Operation requires shutdown of instance "orcl_primary" on database "orcl_primary"
```

```
Shutting down instance "orcl_primary"...
```

```
ORA-01109: database not open
```

```
Database dismounted.
```

```
ORACLE instance shut down.
```

```
Operation requires startup of instance "orcl_primary" on database "orcl_primary"
```

```
Starting instance "orcl_primary"...
```

```
Reinstating database "orcl_primary", please wait...
```

```
Reinstatement of database "orcl_primary" succeeded
```

The successful reinstatement of **ORCL_PRIMARY** as a physical standby database is reflected in the Data Guard configuration as well:

```
DGMGRL> show configuration
```

```
Configuration
```

```
Name: MAA_orcl
```

```
Enabled: YES
```

```
Protection Mode: MaxPerformance
```

```
Databases:
```

```
orcl_stdb1 - Primary database
```

```
orcl_primary - Physical standby database
```

```
Fast-Start Failover: DISABLED
```

```
Current status for "MAA_orcl":  
SUCCESS  
  
DGMGRL> show database orcl_primary  
  
Database  
Name:          orcl_primary  
Role:          PHYSICAL STANDBY  
Enabled:       YES  
Intended State: APPLY-ON  
Instance(s):  
    orcl_primary  
  
Current status for "orcl_primary":  
SUCCESS
```

The corresponding alert log entries from the successful reinstatement process for the **ORCL_PRIMARY** database are shown in [Listing 6.2](#).

Reverting to the Original Data Guard Configuration. Now that I've successfully reinstated my original primary database to a physical standby database, I'll simply perform a switchover operation via DGB to return the original primary and physical standby databases to their initial roles in the Data Guard configuration:

```
DGMGRL> switchover to orcl_primary;  
Performing switchover NOW, please wait...  
New primary database "orcl_primary" is opening...  
Operation requires shutdown of instance "orcl_stdbby1" on database  
"orcl_stdbby1"  
Shutting down instance "orcl_stdbby1"...  
ORA-01109: database not open  
  
Database dismounted.  
ORACLE instance shut down.
```

```
Operation requires startup of instance "orcl_stdb1" on database
"orcl_stdb1"
Starting instance "orcl_stdb1"...
ORACLE instance started.
Database mounted.
Switchover succeeded, new primary is "orcl_primary"
DGMGRL>
```

Next Steps

In the next article in this series, I'll explore ...

- *Configure* the primary database and one physical standby database for *Fast-Start Failover* (FSF)
- *Activate, monitor, and relocate* the *Fast-Start Failover Observer* (FSFO)
- Insure against the loss of a single FSFO via *Enterprise Manager Grid Control*