



VISCOSITY  
NORTH AMERICA

On the Twelfth Day of 18c/19c, Viscosity Gave to me...

## Data Guard New Features

December 23, 2020

A large portion of Oracle customers rely on Data Guard as their disaster recovery solution. With Data Guard, customers can achieve near real-time replication of the database, from the primary to the standby. Now, with Oracle Database 19c, even more innovative features have been added to Data Guard.

### Technology Overview

Database: 18c / 19c

Company: Oracle Corporation

Topic: Data Guard

### Viscosity can help with any of your Database Upgrade needs

Viscosity has performed numerous zero-downtime database migrations and upgrades over the years and has a proven track record with business critical and mission critical databases.

Viscosity's Database Migration & Upgrade Services can plan, upgrade, validate and migrate all database content - quickly and effectively with our automated approach and proven methodology.

Learn more about how you can maintain and maximize your investments at [viscosityna.com](https://viscosityna.com) or email us at [hello@viscosityna.com](mailto:hello@viscosityna.com).

### Active Data Guard DML Redirection

Industry experts claim that Active Data Guard-DML Redirection is the best new feature, introduced in Oracle Database 19c. ADG DML redirection allows data manipulation language (DML), in the form of inserts, updates, and deletes to be executed against the standby database. Incidental DML operations can be run on Active Data Guard standby databases. This allows more applications to benefit from using an Active Data Guard standby database, when some writes are required. On execution the DML operation is passed to the primary database.

DML redirection helps in load balancing between the primary and standby databases. When an incidental DML is issued on an Active Data Guard standby database, the update is passed to the primary database where it is executed. The resulting redo of the transaction updates the standby database, after which control is returned to the application.

DML operations on a standby can be transparently redirected to and run on the primary database. This includes DML statements that are part of PL/SQL blocks. The Active Data Guard session waits until the corresponding changes are shipped to and applied to the Active Data Guard standby.

DML redirection on the standby database requires the Active Directory license; and the COMPATIBLE database initialization parameter must be set to 19.0.0 or higher. The database must be in open mode with archive logs being applied on the standby database, flashback must be enabled as a prerequisite as well. Below are the steps that you can execute to effectively leverage the ADG DML redirection feature:

On the **primary database**, set the ADG\_REDIRECT\_DML parameter.



[www.viscosityna.com](https://www.viscosityna.com)

```
select open_mode, database_role, protection_mode, flashback_on
from v$database;
```

If flashback is not turned on, we need to enable flashback

```
alter database flashback on;
select open_mode, database_role, protection_mode, flashback_on
from v$database;
```

```
show parameter adg_redirect_dml;
alter system set adg_redirect_dml=true scope=both;
show parameter adg_redirect_dml;
```

On the **ADG standby database**, set the ADG\_REDIRECT\_DML parameter.

```
select open_mode, database_role, protection_mode, flashback_on
from v$database;
show parameter adg_redirect_dml;
alter system set adg_redirect_dml=true scope=both;
show parameter adg_redirect_dml;
```

On the **primary database**, create a database user and login as the database user (do not do sqlplus / as sysdba).

```
CREATE USER vnalab identified by lab123;
grant connect,resource to vnalab;
alter user vnalab quota unlimited on users;
sqlplus vnalab/lab123@dg11;

show user;

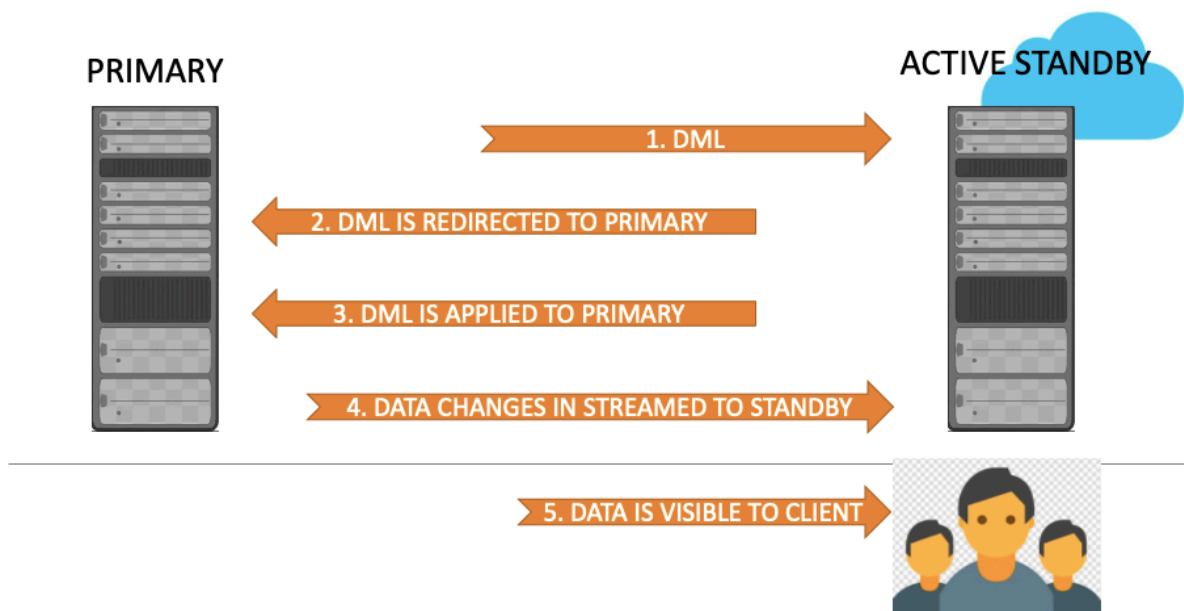
create table mytab as select * from dual;
insert into mytab (dummy) values ('X');
insert into mytab (dummy) values ('Y');
select count (*) from mytab;
commit;
```

On the **ADG standby database**, login as the same database user created earlier in primary database.

```
insert into mytab (dummy) values ('Z');
delete from mytab where dummy = 'X';
insert into mytab (dummy) values ('A');
commit;
```

On the **primary database**, when you execute the query against the mytab table, you will see only 3 values:

```
select * from mytab;
```



## Automatic Flashback of Standby Database

Another notable feature of Data Guard 19c, is the automatic flashback of the standby database. When flashback or point-in-time recovery is performed on the primary database in an Oracle Data Guard configuration, the same operation is also performed on the standby database. Following a flashback or PITR operation, the primary database is then opened with the RESETLOGS option. The RESETLOGS, leads to a new incarnation of the primary or the PDB in the primary. What's new then in Oracle 19c? The MRP process on the standby detects the new incarnation, moving the standby database to the new 'branch' of redo. It will then flash back the standby, or the pluggable database, on the standby to the same point in time, as that of the primary or the PDB on the primary.

In earlier releases, we had to obtain the RESETLOGS SCN# on the primary and then manually issue a FLASHBACK DATABASE command on the standby database, to enable managed recovery and continue with the redo apply process.

## Replicated Restore Points

Another new Oracle Database 19c feature, is when we create a Restore Point on the primary database, it will **automatically create a restore point on the standby database** as well. These restore points are called **Replicated Restore Points**, and have the restore point name suffixed with a "\_PRIMARY". Let's take a look at this feature in action; on the primary database we will create a guaranteed restore point.

```
SQL> select flashback_on from v$database;

FLASHBACK_ON
-----
YES

SQL> create table hr.myobjects as select * from all_objects;
```

Table created.

```
SQL> select count(*) from hr.myobjects;
```

COUNT(*)
71296

```
SQL> create restore point orcl_grp guarantee flashback database;
```

Restore point created.

```
SQL> select name from v$restore_point;
```

NAME
ORCL_GRP

Note that on the standby database, the restore point has been automatically created and the name has the suffix \_PRIMARY.

```
SQL> select flashback_on from v$database;
```

FLASHBACK_ON
YES

```
SQL> select count(*) from hr.myobjects;
```

COUNT(*)
71296

```
SQL> select name from v$restore_point;
```

NAME
ORCL_GRP_PRIMARY

On the primary database, we can see that the REPLICATED column has the value of NO for the restore point, while on

the standby database the value is YES.

Primary

```
SQL> select NAME,REPLICATED from v$restore_point;
```

NAME	REP
-----	
ORCL_GRP	NO

Standby

```
SQL> select NAME,REPLICATED from v$restore_point;
```

NAME	REP
-----	
ORCL_GRP_PRIMARY	YES

We now simulate a case where a human error has been made, and a flashback operation will need to be performed on the primary to resolve the human error.

Flashback is performed to the restore point created earlier and we then open the database with the RESETLOGS option.

```
SQL> truncate table hr.myobjects;
```

Table truncated.

```
SQL> shutdown immediate;
```

Database closed.

Database dismounted.

ORACLE instance shut down.

```
SQL> startup mount;
```

ORACLE instance started.

```
Total System Global Area 1241513488 bytes
Fixed Size                  8896016 bytes
Variable Size              335544320 bytes
Database Buffers          889192448 bytes
Redo Buffers               7880704 bytes
Database mounted.
```

```
SQL> flashback database to restore point orcl_grp;
```

```
Flashback complete.
```

```
SQL> alter database open resetlogs;
```

```
Database altered.
```

The standby database is placed in MOUNT mode, and we will see that the MRP process on the standby database will start and perform the automatic flashback operation on the standby database.

```
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
```

```
SQL> startup mount;
ORACLE instance started.
```

```
Total System Global Area 1241513488 bytes
Fixed Size                  8896016 bytes
Variable Size              318767104 bytes
Database Buffers          905969664 bytes
Redo Buffers               7880704 bytes
Database mounted.
```

```
...
```

```
...
```

```

...
...
rfs (PID:27484): Primary database is in MAXIMUM PERFORMANCE mode
2019-12-01T23:00:30.671381+08:00
rfs (PID:27484): Selected LN0:5 for T-1.S-3 dbid 1540291890 branch 1012517366
2019-12-01T23:00:34.853803+08:00
ARC0 (PID:27457): Archived Log entry 9 added for T-1.S-3 ID 0x5bcedc53 LAD:1
2019-12-01T23:00:34.930249+08:00
rfs (PID:27500): Primary database is in MAXIMUM PERFORMANCE mode
2019-12-01T23:00:35.123538+08:00
rfs (PID:27500): Selected LN0:5 for T-1.S-4 dbid 1540291890 branch 1012517366
2019-12-01T23:00:35.909833+08:00
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT NODELAY
2019-12-01T23:00:35.910522+08:00
Attempt to start background Managed Standby Recovery process (orcl)
Starting background process MRP0
2019-12-01T23:00:35.932806+08:00
MRP0 started with pid=49, OS id=27505
2019-12-01T23:00:35.935882+08:00
Background Managed Standby Recovery process started (orcl)
2019-12-01T23:00:40.940224+08:00
Serial Media Recovery started

...
...

MRP0 (PID:27505): Recovery coordinator performing automatic flashback of database to SCN:
0x00000000001fcce5 (2084069)
Flashback Restore Start
Flashback Restore Complete
Flashback Media Recovery Start
2019-12-01T23:01:01.852650+08:00
Setting recovery target incarnation to 2
2019-12-01T23:01:01.852994+08:00

```

```

Serial Media Recovery started
stopping change tracking
2019-12-01T23:01:01.976765+08:00
Media Recovery Log /u01/app/oracle/fast_recovery_area/ORCL_SB/archivelog/2019_12_01/o1_mf_
_1_9_gkn5pkx9_.arc
2019-12-01T23:01:02.142326+08:00
Resize operation completed for file# 3, old size 522240K, new size 552960K
Restore point ORCL_GRP_PRIMARY propagated from primary already exists
2019-12-01T23:01:02.215320+08:00
Media Recovery Log /u01/app/oracle/fast_recovery_area/ORCL_SB/archivelog/2019_12_01/o1_mf_
_1_10_gkn5pnf4_.arc
2019-12-01T23:01:02.657960+08:00
Media Recovery Log /u01/app/oracle/fast_recovery_area/ORCL_SB/archivelog/2019_12_01/o1_mf_
_1_11_gkn6s6g1_.arc
2019-12-01T23:01:02.795226+08:00
Media Recovery Log /u01/app/oracle/fast_recovery_area/ORCL_SB/archivelog/2019_12_01/o1_mf_
_1_12_gkn7cxv1_.arc
2020-12-01T23:01:02.956167+08:00
Completed: ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT NODELAY
2020-12-01T23:01:04.624321+08:00
Incomplete Recovery applied until change 2084069 time 12/01/2020 22:47:05
Flashback Media Recovery Complete

```

When we see the message “Flashback Media Recovery Complete” in the standby database alert log, we can now open the standby database.

*Note: that the Data Guard Broker configuration does not show any error(s), and we did not have to perform any manual steps on the standby database to enable the configuration following the flashback of the primary database.*

```

SQL> ALTER DATABASE OPEN;
Database altered.

DGMGRL> show configuration;

Configuration - orcl_dg

Protection Mode: MaxPerformance

Members:

```



```
orcl      - Primary database
orcl_sb   - Physical standby database
```

Fast-Start Failover: Disabled

Configuration Status:

SUCCESS (status updated 54 seconds ago)

## Tuning Automatic Resolution of Data Guard

Oracle introduced two additional parameters for Data Guard, in Oracle Database 19c, to tune automatic outage resolution:

- DATA\_GUARD\_MAX\_IO\_TIME
- DATA\_GUARD\_MAX\_LONG\_IO\_TIME

The DATA\_GUARD\_MAX\_IO\_TIME has a range of 10 to 7200, with a default value of 240; which specifies the maximum number of seconds that can elapse before Data Guard considers a process performing read, write, and status I/O operations on the Data Guard configuration in a hung state.

The DATA\_GUARD\_MAX\_LONG\_IO\_TIME also has a range of 10 to 7200, with a default value of 240; which specifies the maximum number of seconds that can elapse before Data Guard considers a process long I/O open, and closes operations on the Data Guard configuration in a hung state.

These two parameters come in handy when Data Guard detects hung processes, caused by delayed network outages, network disconnects, and disk I/O issues. These parameters allow for wait time to be customized and tuned for Data Guard before terminating redo apply processes, to automate outage resolution caused by network outages and disk I/O issues.

## Fast Start Failover (FSFO) in Observe Only Mode

FSFO allows for the broker to automatically failover the database to the standby, when the observer process detects a failed condition to the primary database. Until Oracle Database 19c, we could not really test a failover condition, without making changes to the configuration or impacting the primary database. Now, we can configure FSFO to be on Observe Only mode and create a test situation to see when a failover, or other interaction, would have occurred during the normal production processing. Within the Broker, we will issue the following command:

```
DGMGRL> ENABLE FAST_START FAILOVER OBSERVER ONLY;  
Enabled in Observe-Only Mode.
```

This feature allows for us to tune FSFO, for finer granularity and improve our failover validation. This method, allows you to see if FSFO would have actually occurred, without production impact.

## Export and Import Broker Configuration

Starting in Oracle Data Guard 19c, we can backup the configuration of the broker to a text file. Now, export and import of our Data Guard Broker configuration can be to an external text file. The export of DG configuration can be done with the EXPORT CONFIGURATION command:

```
DGMGRL> export configuration to 'dg11.txt';
Succeeded.
```

A snippet of the broker export file looks like this:

```
# cat /opt/oracle/diag/rdbms/dg11/DG11/trace/dg11.txt
<?xml version="1.0" encoding="UTF-8"?>
<DRC Version="19.0.0.0.0" CurrentPath="True" Name="dg1">
  <DefaultState>ONLINE</DefaultState>
  <DRC_UNIQUE_ID>1499567441</DRC_UNIQUE_ID>
  <IntendedState>ONLINE</IntendedState>
  <FastStartFailoverOBID1>191512127</FastStartFailoverOBID1>
  <FastStartFailoverOBID2>191512128</FastStartFailoverOBID2>
  <FastStartFailoverOBID3>191512129</FastStartFailoverOBID3>
  <Configuration_Name>dg1</Configuration_Name>
  <Member MemberID="1" CurrentPath="True" Enabled="True" MultiInstanced="True"
Name="dg11">
    <DefaultState>PRIMARY</DefaultState>
    <IntendedState>PRIMARY</IntendedState>
    <Status>
      <Severity>Success</Severity>
      <Error>0</Error>
      <Timestamp>1601355896</Timestamp>
    </Status>
    <StandbyType>PhysicalStandby</StandbyType>
    <DGConnectIdentifier>dg11</DGConnectIdentifier>
    <DbDomain/>
```

Likewise, we can import the Data Guard metadata configuration file with the IMPORT CONFIGURATION command:

```
DGMGRL> import configuration file from 'dg11.txt'
```

## Summary

The ADG DML Redirection feature is ideal for mostly read standby databases, where customers need some DML activity on the standby database. Now, with innovative features introduced in Oracle Database 19c, customers can offload more workloads from the primary database. As many are embarking on the cloud journey, customers are choosing Data Guard over RAC for high availability strategies.

Other features introduced in Oracle Database 19c, such as automatic flashback of the standby databases when the primary database executes a flashback, allows for easier management for the DBAs. Replicated restore points also

simplify the manageability of the primary and standby databases. Other innovative features for Oracle Database 19c worth noting are:

- The ability to dynamically change the FSFO target without FSFO being disabled
- Extending the capability to have Multi-Instance Redo Apply with In-Memory

One of the sayings we have at Viscosity is our customers “have four aces in their pocket”. Over the previous 11 days, the talented staff at Viscosity along with our Oracle ACEs have addressed more Oracle Database 18c and 19c new features. Continue to join us next year, as we continue our Oracle Database 19c hands-on-lab workshops.

Happy Holidays!