

Oracle Maximum
Availability Architecture

Role Transition Best Practices

Data Guard and Active Data Guard

ORACLE WHITE PAPER | AUGUST 2018





Table of Contents	
Introduction	2
Observed Role Transition Performance	2
Oracle Data Guard Switchover Best Practices	4
Optimization	4
Oracle Data Guard Failover Best Practices	6
Comparing Data Guard Fast-Start Failover and Manual Failover	6
Generic Failover Best Practices (Manual Failover and Fast-Start Failover)	7
Fast-Start Failover Best Practices	8
Manual Failover Best Practices	9
Client Failover and Application Continuity	9
Troubleshooting Role Transitions	10
Analysis of Role Transition Performance	10
Key Operations during Switchover	11
Key Operations during Failover	11
Methods for Role Transition Performance Analysis	11
Time Management Interface Event	11
Key Alert Log Tags Using X\$DBGALERTEXT	13
Additional Resources	15
Conclusion	15

Introduction

With proper planning and execution, Oracle Data Guard and Active Data Guard role transitions can effectively minimize downtime and ensure that the database environment is restored with minimal impact on the business. Performance tests using physical standby database and Oracle Maximum Availability Architecture (MAA) best practices provided in this paper have shown that switchover and failover times with Oracle Database 11g Release 2 can be reduced to seconds and further improvement is achieved with Oracle Database 12c.

Note that Active Data Guard is a superset of Data Guard functionality. All information in this white paper is relevant to both Active Data Guard and Data Guard except for where explicit reference is made to Active Data Guard.

Observed Role Transition Performance

The following results have been observed in the Oracle MAA lab environment.

An overview of Data Guard role transition timings across four Oracle Database releases, each with a RAC primary database and a RAC standby database, is provided in Table 1. These timings measure the time from the beginning of the role transition to the point where the new primary database is open and receiving client connections. The times shown in Table 1 represent a baseline measurement taken during periods with minimum workload during Data Guard failover and no workload during Data Guard switchover operations.

Most patch sets and Patch Set Updates will include optimizations for Data Guard. The fastest role transition times will be achieved with the latest updates for the most current releases of Oracle Database (Oracle Database 11.2.0.4 and Oracle Database 12.2.0.2 at the time this paper was published). Refer to [My Oracle Support Note 756671.1](#) for the latest information on recommended patches.

TABLE 1: ROLE TRANSITION TIMINGS

Database Releases	Failover Time	Switchover Time
10.2 RAC Database/ Data Guard	60 secs	65 secs
11.2 RAC Database/ Active Data Guard	44 secs	54 secs
12.1.0.1 RAC Database/ Active Data Guard	27 secs	41 secs
12.1.0.2 RAC Database/ Active Data Guard	16 secs	34 secs
12.2 RAC Database/ Active Data Guard	15 sec	21 sec

While the baseline measurements show that role transition times have steadily improved, there are many variables in a production deployment that can influence the speed at which Data Guard performs the transition. Variables

include: RAC or single instance databases, open or mounted standbys and varying workloads on a primary database or an Active Data Guard standby. Each of these variables will have an effect on the time it takes for a new primary database to open after a role transition.

Tables 2-5 show how role transition times are affected in Oracle Database Release 12.2 by different configurations and workloads. The tables also break these timings down by the particular step during the role transition. This is important as it will help you to understand where time is spent during role transitions in your environment.

Note: Any differences in report total time and the sum of the phases are a result of rounding

TABLE 2: ORACLE 12.2 SWITCHOVER - NO WORKLOAD

Configuration	Total Time	Convert to Standby	Cancel Recovery	Convert to Primary	Open Primary
Single Instance Databases/Mounted Standby	00:11.7	00:04	00:01	00:03	00:04
Single Instance Databases/Active Data Guard	00:12.0	00:04	00:01	00:03	00:04
Oracle RAC Databases/Mounted Standby	00:17.4	00:08	00:02	00:03	00:05
Oracle RAC Databases/Active Data Guard	00:21.8	00:08	00:02	00:07	00:05

TABLE 3: ORACLE 12.2 SWITCHOVER - ACTIVE DATA GUARD STANDBY AND WORKLOAD ON PRIMARY AND STANDBY

Configuration	Total Time	Convert to Standby	Cancel Recovery	Convert to Primary	Open Primary
Single Instance Databases- 250 Connections	00:22.4	00:07	00:01	00:10	00:05
Single Instance Databases- 500 Connections	00:26.6	00:08	00:01	00:13	00:05
Oracle RAC Databases- 500 Connections	00:31.8	00:12	00:02	00:13	00:06
Oracle RAC Databases- 1000 Connections	00:38.9	00:13	00:02	00:15	00:08

TABLE 4: ORACLE 12.2 FAILOVER - NO WORKLOAD

Configuration	Total Time	Convert to Standby	Terminal Recovery	Convert to Primary	Open Primary
Single Instance Databases/Mounted Standby	00:05.7	00:01	00:01	00:01	00:03
Single Instance Databases/Active Data Guard	00:09.0	00:04	00:01	00:01	00:03
Oracle RAC Databases/Mounted Standby	00:08.3	00:01.0	00:01	00:01	00:05
Oracle RAC Databases/Active Data Guard	00:15.6	00:05	00:04	00:01	00:05

TABLE 5: ORACLE 12.2 FAILOVER WITH ACTIVE DATA GUARD AND WORKLOAD ON PRIMARY AND STANDBY

Configuration	Total Time	Convert to Standby	Terminal Recovery	Convert to Primary	Open Primary
Single Instance Databases- 250 Connections	00:12.7	00:06.0	00:02.6	00:00.5	00:03.6
Single Instance Databases- 500 Connections	00:17.6	00:13	00:01	00:01	00:03
Single Instance Databases- 500 Connections with STANDBY_DB_PRESERVE_STATES=SESSION	00:13.9	00:09	00:01	00:01	00:03
Oracle RAC Databases- 500 Connections	00:24.6	00:09.8	00:08.1	00:00.5	00:06.1
Oracle RAC Databases- 1000 Connections	00:20	00:11	00:03	00:01	00:05
Single Instance Databases- 500 Connections with STANDBY_DB_PRESERVE_STATES=SESSION	00:18.3	00:09	00:03	00:01	00:05

As expected, Active Data Guard, Oracle RAC and the presence of active connections will each increase the time for a role transition to complete. This is due to the additional steps required such as closing a database, handling additional instances and cleanup of connections. Switchover (planned event) is also expected to take longer than failover due to the coordination between the primary and standby that ensures no data is lost in asynchronous configurations. This is not the case with a failover (unplanned event) where the potential for data loss is controlled by the protection mode that was in effect at the time the outage occurred; failover is immediate with the standby simply being converted to primary and opened. Additional factors that impact role transition times include resource contention, size of the SGA of the database and the number of data files in the database. Each of these factors affect the time required to open and close a database.

Oracle Data Guard Switchover Best Practices

A database switchover performed by Oracle Data Guard is a planned transition that includes a series of steps to switch roles between a standby database and a primary database. Following a successful switchover operation, the standby database assumes the primary role and the primary database becomes a standby database. Switchovers are typically completed in seconds to minutes. The term switchback is used to describe a second switchover operation that returns the databases to their original roles.

Data Guard enables you to change these roles dynamically using one of the following methods:

- » Using Oracle Enterprise Manager (recommended)
- » Using the Oracle Data Guard broker's DGMGRL command-line interface (recommended)
 - » Issuing SQL statements, as described in MAA best practice guide([11.2](#) | [12.1](#))

Refer to Oracle Data Guard Broker documentation ([11.2](#) | [12.1](#)) for information about using Oracle Data Guard broker's DGMGRL command-line interface to perform database switchover.

Optimization

To optimize switchover processing, perform the following steps before performing a switchover:

- » Gracefully stop application processing against the database or disconnect all sessions possible using the `ALTER SYSTEM KILL SESSION SQL*`Plus command.
- » Stop job processing by setting the `AQ_TM_PROCESSES` parameter to 0.
- » Cancel any specified apply delay by setting the Broker property `DelayMins=0`

To accomplish the same where broker is not configured issue the following commands on the standby database in SQL*Plus.

- » `ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;`
- » `ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE NODELAY DISCONNECT;`

The `USING CURRENT LOGFILE` clause is no longer required in 12.1 and above. By default SRLs are used where they exist.

You can view the current delay setting on the primary database by referencing the broker property `DelayMins` from the broker 'show database <db_unique_name> verbose' output or by querying the `DELAY_MINS` column of the `V$ARCHIVE_DEST` view.

- » In Oracle Database versions 11.2.0.2 through 11.2.0.3 for physical standby database configurations in an Oracle RAC environment, set "`_switchover_to_standby_option`"= `OPEN_ALL_IGNORE_SESSIONS`. This becomes the default behavior in version 11.2.0.4.

This parameter will terminate all remaining primary instances at the right time during execution and ignores active sessions. You can keep all nodes of the primary open when performing "switchover to standby". In lieu of this parameter, ensure that there is only one instance active for each primary and standby database and use the "WITH SESSION SHUTDOWN" clause with the switchover command.

- » In Oracle Database versions 11.2.0.2 through 11.2.0.3 set "`_defer_eor_orl_arch_for_so`"=`TRUE`. This becomes the default behavior in version 11.2.0.4.

This parameter defers the archiving of the final End-Of_Redo (EOR) redo log files at the old primary. Performing this archival when the old primary is mounted as a standby saves time during switchover.

- » Configure the standby database to use real-time apply and, if possible, ensure the databases are synchronized before the switchover operation to optimize switchover processing.

Using real-time apply ensures that the redo data is applied to the standby database as soon as it is received, and the standby database is synchronized with the primary database before the switchover operation to minimize switchover time.

The Data Guard Broker always uses real-time apply when the Broker property `DelayMins` is set to 0 so you can skip this step if you are using the Broker.

To enable real-time apply use the following SQL*Plus statement:

```
ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT USING CURRENT LOGFILE;
```

The `USING CURRENT LOGFILE` clause is no longer required in 12.1 and above. Real-time apply is the default where SRLs exist in 12c provided there is no apply delay explicitly specified.

- » If not using the Data Guard broker or Enterprise Manager you will need to manually shutdown the non-apply standby instances prior to Data Guard switchover. The broker and Enterprise Manager handle this automatically for you.

- » For the target physical standby database, reduce the number of archiver (ARCn) processes to the minimum needed for both remote and local archiving.

```
ALTER SYSTEM SET LOG_ARCHIVE_MAX_PROCESSES=
  ((#of defined LOG_ARCHIVE_DEST_n )+(#threads/instances))
  SCOPE=MEMORY SID='*';
```

Additional archiver processes can take additional time to shut down, thereby increasing the overall time it takes to perform a switchover. After the switchover has completed you can reenable the additional archiver processes.

- » Use Oracle Managed Files (OMF) or alternatively, set the LOG_FILE_NAME_CONVERT initialization parameter to any valid value for the environment.

As part of a switchover, the standby database must create/clear the online redo log files on the standby database before opening as a primary database. The time needed to complete the I/O can significantly increase the overall switchover time. By using OMF (or setting the LOG_FILE_NAME_CONVERT parameter) the standby database can pre-create the online redo logs the first time the MRP process is started. You can also manually pre-clear empty online redo logs by issuing the SQL*Plus ALTER DATABASE CLEAR LOGFILE statement on the standby database.

Oracle Data Guard Failover Best Practices

A failover is used when the primary database is deemed lost or unrecoverable, or the expected time to repair exceeds the required recovery time objective (RTO). During a failover the primary database is taken offline at one site and a standby database is brought online as the primary database.

Failover can be completely automated using Data Guard Fast-Start Failover or it can be a manual, administrator-driven process using Enterprise Manager, the Data Guard Broker, or SQL*Plus command line. Oracle recommends using Fast-Start Failover to eliminate the uncertainty inherent in a process that requires manual intervention, assuming similar measures have been taken to automate the failover of the application tier to the new primary database. Fast-Start Failover automatically executes a database failover within seconds of an outage being detected and can complete in seconds.

Comparing Data Guard Fast-Start Failover and Manual Failover

An administrator initiates manual failover when the primary database fails. In contrast, Data Guard automatically initiates a fast-start failover without human intervention after the primary database has been unavailable for a set period (the fast-start failover threshold). Manual and automatic database failover are compared in Table 6.

TABLE 6 COMPARING FAST-START FAILOVER AND MANUAL FAILOVER

Points of Comparison	Fast-Start Failover	Manual Failover
Benefits	<p>Allows you to increase availability with less manual intervention, thereby reducing management costs.</p> <p>RTO reduced to seconds</p> <p>Manageability costs and mistakes during failover operation reduce significantly</p>	<p>Gives you control over exactly when a failover occurs and to which target standby database.</p>

Points of Comparison	Fast-Start Failover	Manual Failover
	No user intervention in the case of a failure.	
Failover triggers	<p>The following conditions automatically trigger a fast-start failover:</p> <ul style="list-style-type: none"> » Database instance failure (or last instance failure in an Oracle RAC configuration). » Database failure, shutdown abort (or a shutdown abort of the last instance in an Oracle RAC configuration). » Specific conditions that are detected through the database health-check mechanism (for example, data files taken offline due to I/O errors). » Primary database is isolated from both the observer and the target standby but observer and target standby are still communicating (prevents split brain). » Application initiated fast-start failover using the <code>DBMS_DG.INITIATE_FS_FAILOVER</code> PL/SQL procedure. 	<p>A manual failover is user initiated and converts a standby database into a primary database. Manual failover is a response to an unplanned outage such as:</p> <ul style="list-style-type: none"> » Site disaster which results in the primary database becoming unavailable (all instances of an Oracle RAC primary database). » Any database outage that cannot be resolved within the required recovery time objective. » User errors that cannot be repaired in a timely fashion. » Data failures or corruptions, which impact the production database.
Management	<p>Use the following tools to manage fast-start failover:</p> <ul style="list-style-type: none"> » Oracle Enterprise Manager » The Oracle Data Guard broker command-line interface (DGMGRL) 	<p>Use the following tools to perform manual failovers:</p> <ul style="list-style-type: none"> » Oracle Enterprise Manager » The Oracle Data Guard broker command-line interface (DGMGRL) » SQL statements (If not using the Broker)
Restoring the original primary database after failover	<p>Following a fast-start failover, Oracle Data Guard broker can automatically reconfigure the original primary database as a standby database upon reconnection to the configuration (FastStartFailoverAutoReinstate), or you can delay the reconfiguration to allow diagnostics on the failed primary. Automatic reconfiguration enables Data Guard to restore disaster protection in the configuration quickly and easily, returning the database to a protected state as soon as possible.</p>	<p>The original primary database must be manually reinstated as a standby database to restore fault tolerance. This manual process is automated when using Fast-Start Failover.</p>
Restoring bystander standby databases after failover	<p>Oracle Data Guard broker coordinates the role transition on all databases in the configuration. Bystanders that do not require reinstatement are available as viable standby databases to the new primary. Bystanders that require reinstatement are automatically reinstated by the observer.</p>	<p>A benefit of using Oracle Data Guard broker is that it provides the status of bystander databases and indicates whether a database must be manually reinstated. Status information is not readily available when using SQL*Plus statements to manage failover.</p>

Generic Failover Best Practices (Manual Failover and Fast-Start Failover)

To optimize failover processing:

- » Use Data Guard Broker or Enterprise Manager
- » Use real-time apply to apply redo data to the standby database as soon as it is received.

- » Enable Flashback Database on all databases to reinstate a failed primary database after a failover operation has occurred. Assuming a failed primary can be repaired; Flashback Database will return it to a point in time prior to when the standby became the new primary. MAA best practice recommends a minimum of 2 hours of flashback log retention if used for reinstatement in a Data Guard configuration (the default flashback log retention period is 24 hours). Refer to [MOS 565535.1](https://support.oracle.com/epmos/faces/DocumentDisplay?id=565535.1)¹ for flashback database best practices.
- » Use Flashback Database to quickly rewind the database should user error or logical corruption be detected. Increase the flashback log retention by more than the minimum of 2 hours if a longer flashback period is desired for fast point-in-time recovery.
- » Consider configuring multiple standby databases to maintain data protection following a failover.
- » Set the LOG_FILE_NAME_CONVERT parameter. As part of a failover, the standby database must clear its online redo logs before opening as the primary database. The time needed to complete this I/O can add significantly to the overall failover time. By setting the LOG_FILE_NAME_CONVERT parameter, the standby pre-creates or clears the online redo logs when the MRP process is started. You can also pre-create empty online redo logs by issuing the SQL*Plus ALTER DATABASE CLEAR LOGFILE statement on the standby database. In Oracle Database 12c the broker command, VALIDATE DATABASE will identify redo logs which have not been cleared. Prior to Oracle Database 12c you can determine if the target standby database has had its Online Redo Logs cleared. On the target physical standby run the following query to determine if the online redo logs have not been cleared:

```
SQL> SELECT DISTINCT L.GROUP# FROM V$LOG L, V$LOGFILE LF
WHERE L.GROUP# = LF.GROUP# AND L.STATUS NOT IN
('UNUSED', 'CLEARING', 'CLEARING_CURRENT');
```

If this query returns rows, issue the following statement for each GROUP# returned on the standby:

```
SQL> ALTER DATABASE CLEAR LOGFILE GROUP <ORL GROUP# from the query above>;
```

- » Use fast-start failover. The MAA tests running Oracle Database 11g show that failovers performed using Oracle Data Guard broker and fast-start failover offer a significant improvement in availability. For more information, see the 'Fast Start Failover Best Practices' section below

Fast-Start Failover Best Practices

Fast-start failover automatically, quickly, and reliably fails over to a previously designated standby database if the primary database fails, without requiring manual intervention to execute the failover. You can use fast-start failover only in an Oracle Data Guard configuration that is managed by Oracle Data Guard broker.

The Oracle Data Guard configuration can be running in either the Maximum Availability or Maximum Performance mode with fast-start failover. When fast-start failover is enabled, the broker ensures fast-start failover is possible only when the configured data loss guarantee can be upheld. Maximum Availability mode provides an automatic failover environment guaranteed to lose no data. Maximum Performance mode provides an automatic failover environment guaranteed to lose no more than the amount of data (in seconds) specified by the administrator using the FastStartFailoverLagLimit configuration property.

Use the following fast-start failover best practices in addition to the generic best practices listed in the '[Failover Best Practices](#)' section above.

- » Run the fast-start failover observer process on a host that is not located in the same data center as the primary or standby database.

¹ <https://support.oracle.com/epmos/faces/DocumentDisplay?id=565535.1>

Ideally, you should run the observer on a system that is equally distant from the primary and standby databases. The observer should connect to the primary and standby databases using the same network as any end-user client. If the observer cannot run at a third site, then you should install the observer on the same network as the application. If a third, independent location is not available, then locate the observer in the standby data center on a separate host and isolate the observer as much as possible from failures affecting the standby database.

- » Make the observer highly available by configuring auto restart on the same server or on an alternate host using Oracle Enterprise Manager. .
- » Use the `FastStartFailoverAutoReinstate` configuration property set to `TRUE` (this is the default), to configure the original primary database to be automatically reinstated as a standby database when a connection to the database is reestablished.
- » Set the value of the `FastStartFailoverThreshold` property according to your configuration characteristics, as described in Table 7. As with all configuration settings, carefully test the value chosen to ensure that it meets your objectives.

TABLE 7: MINIMUM RECOMMENDED SETTINGS FOR FASTSTARTFAILOVERTHRESHOLD

Configuration	Minimum Recommended Setting
Single-instance primary, low latency, and a reliable network	10 seconds
Single-instance primary and a high latency network over WAN	30 seconds
Oracle RAC primary	Oracle RAC misscount + reconfiguration time + 30 seconds
Oracle RAC primary on Exadata with 12.1.0.2 BP8 or higher	5 seconds + reconfiguration time + 30 seconds

Note: The Data Guard Broker will allow `FASTSTARTFAILOVERTHRESHOLD` to be set as low as 6 seconds to minimize total failover time in Oracle Database 12c (the default is 30 seconds). Oracle only recommends considering such a low setting for single instance primary and standby databases deployed on a LAN characterized by low latency and high reliability. Even then the potential for false failovers due to transitory network issues will increase. Thorough testing is required if setting the threshold value lower than recommended in Table 7.

Manual Failover Best Practices

If fast-start failover is not used then a manual failover must be performed during unplanned outages. Use the best practices listed in the [‘Failover Best Practices \(Manual Failover and Fast-Start Failover\)’](#) section above.

In addition you will need to manually reinstate the original primary database as a standby database once it has been repaired to restore fault tolerance to your environment. The standby database can be quickly reinstated by using Flashback Database. See MAA Best Practice Guide ([11.2](#) | [12.1](#)).

STANDBY_DB_PRESERVE_STATES

As of Oracle Database version 12.2 the parameter `STANDBY_DB_PRESERVE_STATES` setting of `SESSION` will suspend the read-only sessions on an active physical standby database (ADG) instead of



killing them. This allows those sessions to continue where they left off after the standby is brought up as a primary database. This feature can help preserve long running reports on a standby through a role transition. Preserving the session states can also slightly improve the role transition time as seen in tables 3 and 5. Resource planning should account for those read only sessions running on the primary database after role transition if this parameter is to be used.

Client Failover and Application Continuity

The process by which application clients are reconnected to a new primary database after a failover can be automated using the following process:

- » Automatic relocation of database services to the new primary database as part of a Data Guard failover
- » Fast notification that a failure has occurred to immediately break client connections out of TCP timeout.
- » Automatic redirection of clients to the new primary database.
- » Controlling logon storms

Configuration details are covered in MAA Best Practices for Client Failover for Oracle [Database11g](#)² and for Oracle [Database 12c](#)³. Please consult these papers and configure your environment appropriately.

Troubleshooting Role Transitions

If a problem is encountered during a role transition the information you provide to Oracle Support when the service request is created will be essential to achieving fast resolution. Please provide the information listed below:

- » Simple problem statement: e.g. switchover failed with error at this date and current primary is down.
- » Data Guard topology: Primary database host names, Standby databases and corresponding hostnames
- » Exact timeline for the role transition. e.g. node and time for issuing the command, error time/location etc.
- » Exact command issued. e.g. switchover to..., failover to..., alter database... etc.
- » All broker logs, alert logs, crsd logs from all nodes and instances for ONLY the time involved in the role transition.
 - Broker logs (drc<sid>.log) and alert logs are located in the background_dump_dest directory for the instances.
 - crsd logs in 12c grid infrastructure- \$ORACLE_BASE/diag/crs/<node>/crs/trace
 - crsd logs in 11.2 grid infrastructure- \$GRID_HOME/log/<node>/crsd

Diagnostic Assistant and [Oracle Trace File Analyzer](#)⁴ collector is an efficient way of collecting trace files for a specific period of time.

Analysis of Role Transition Performance

The speed of role transitions has improved over time however there may still be occasions when switchover or failover take longer than expected. Identifying the main elements of the process and determining how much time is

² <http://www.oracle.com/technetwork/database/features/availability/maa-wp-11gr2-client-failover-173305.pdf>

³ <http://www.oracle.com/technetwork/database/availability/client-failover-2280805.pdf>

⁴ <http://www.oracle.com/technetwork/products/clustering/overview/tracefileanalyzer-2008420.pdf>

spent in each step during a role transition is the quickest way to uncover the root cause of a problem. Understanding the key operations for each type of role transition, switchover and failover, is critical for this task.

Key Operations during Switchover

Switchover is broken down into four main steps as follows.

- » Convert to Standby: Kill any existing production sessions, convert the control file into a standby control file and send a message to the standby to continue the switchover. The Convert to Standby steps will be found in the alert log of the original primary. All remaining steps are found in the original standby alert log.
- » Cancel Recovery: Apply remaining redo and stop recovery
- » Convert to Primary: Two-step close (to the mounted state) of instances (one instance, then all others), Clear Online Redo Logs, Convert control file to primary controlfile, Data Guard Broker book-keeping.
- » Open New Primary: Parallel open of all instances

The output of each step required for analysis will exist in the alert log of the one instance from each database which executes the switchover steps.

The most common deviations from lab results listed in the next section are in the 'Open New Primary' step which can vary based on size of the buffer cache and amount of undo to initialize due to uncommitted transactions when a workload is running. The 'Convert to Standby' step may also vary depending on the number of active sessions which need to be killed.

Key Operations during Failover

All steps will be documented in the alert log of the target standby where the failover was performed.

- » Cancel Recovery- Stop recovery and close all instances(to mounted) in parallel
- » Terminal Recovery- Archive standby redo logs and recover any unapplied redo
- » Convert to Primary- Clear online redo logs, convert control file to standby control file
- » Open Primary- open of all instances in parallel.

The most common deviations from lab results listed in the next section are in the 'Open New Primary' step which can vary based on size of the buffer cache.

Methods for Role Transition Performance Analysis

There are two options for role transition performance analysis in Oracle Database 11.2.0.4 and above.

Time Management Interface Event

The first option is a low overhead event which adds a line to the alert log whenever certain calls are executed in Oracle. These lines in the alert log or 'tags' delineate the beginning and end of a call and Tables 8-11 depict the delineation of the steps described above for [switchover](#) and [failover](#). This method is the most accurate.

Set the database level event `16453 trace name context forever, level 15` on all databases. There are two methods of enabling this trace, either using the EVENT database parameter or setting the EVENTS at the system level. The difference is that the EVENT parameter is not dynamic but is persistent across restarts. 'set EVENTS' is dynamic but NOT persistent across database restarts. `ALTER SYSTEM SET EVENT='16453 trace name context forever, level 15' scope=spfile sid='*'`

or

```
ALTER SYSTEM SET EVENTS '16453 trace name context forever, level 15';
```

With Oracle Database 12.2 and above, the trace does not need to be set for each role transition. The tags listed below are printed to the alert log intrinsically.

TABLE 8: SWITCHOVER STEP DELINEATION TAGS WITH TIME MANAGEMENT INTERFACE EVENT SET FOR 12c

Oracle Database 12c Switchover w/TMI		
Convert To Standby (primary alert log)	BEGIN	TMI: dbsdrv switchover to target BEGIN <DATE> <TIMESTAMP>
	END	TMI: kcv_switchover_to_target send 'switchover to primary' msg BEGIN <DATE> <TIMESTAMP>
Cancel Recovery (standby alert log)	BEGIN	TMI: kcv_commit_to_so_to_primary wait for MRP to die BEGIN <DATE> <TIMESTAMP>
	END	TMI: kcv_commit_to_so_to_primary wait for MRP to die END <DATE> <TIMESTAMP>
Convert to Primary (standby alert log)	BEGIN	TMI: kcv_commit_to_so_to_primary BEGIN CTSO to primary <DATE> <TIMESTAMP>
	END	TMI: adbdrv BEGIN 10 <DATE> <TIMESTAMP>
Open Primary (standby alert log)	BEGIN	TMI: adbdrv BEGIN 10 <DATE> <TIMESTAMP>
	END	TMI: adbdrv END 10 <DATE> <TIMESTAMP>

TABLE 9: SWITCHOVER STEP DELINEATION TAGS WITH TIME MANAGEMENT INTERFACE EVENT SET FOR 11.2

Oracle Database 11.2 Switchover w/TMI		
Convert To Standby (primary alert log)	BEGIN	TMI: adbdrv BEGIN 72 <DATE> <TIMESTAMP>
	END	TMI: adbdrv END 72 <DATE> <TIMESTAMP>
Cancel Recovery (standby alert log)	BEGIN	TMI: adbdrv BEGIN 62 <DATE> <TIMESTAMP>
	END	TMI: adbdrv END 62 <DATE> <TIMESTAMP>
Convert to Primary (standby alert log)	BEGIN	TMI: adbdrv BEGIN 72 <DATE> <TIMESTAMP>
	END	TMI: adbdrv BEGIN 10 <DATE> <TIMESTAMP>
Open Primary (standby alert log)	BEGIN	TMI: adbdrv BEGIN 10 <DATE> <TIMESTAMP>
	END	TMI: adbdrv END 10 <DATE> <TIMESTAMP>

TABLE 10: FAILOVER STEP DELINEATION TAGS WITH TIME MANAGEMENT INTERFACE EVENT SET FOR 12C

Oracle Database 12c Failover w/TMI (all events in standby alert log)		
Cancel Recovery	BEGIN	TMI: adbdrv termRecovery BEGIN <DATE> <TIMESTAMP>
	END	TMI: adbdrv termRecovery END <DATE> <TIMESTAMP>
Terminal Recovery	BEGIN	TMI: krdsmr full BEGIN Starting media recovery <DATE> <TIMESTAMP>

	END	TMI: krdevr full END end media recovery <DATE> <TIMESTAMP>
Convert to Primary	BEGIN	TMI: kcv_commit_to_so_to_primary BEGIN CTSO to primary <DATE> <TIMESTAMP>
	END	TMI: adbdrv BEGIN 10 <DATE> <TIMESTAMP>
Open Primary	BEGIN	TMI: adbdrv BEGIN 10 <DATE> <TIMESTAMP>
	END	TMI: adbdrv END 10 <DATE> <TIMESTAMP>

TABLE 11: FAILOVER STEP DELINEATION TAGS WITH TIME MANAGEMENT INTERFACE EVENT SET FOR 11.2

Oracle Database 11.2 Failover w/TMI (all events in standby alert log)		
Cancel Recovery	BEGIN	TMI: adbdrv BEGIN 62 <DATE> <TIMESTAMP>
	END	TMI: adbdrv END 62 <DATE> <TIMESTAMP>
Terminal Recovery	BEGIN	TMI: adbdrv BEGIN 52 <DATE> <TIMESTAMP>
	END	TMI: adbdrv END 52 <DATE> <TIMESTAMP>
Convert to Primary	BEGIN	TMI: adbdrv BEGIN 72 <DATE> <TIMESTAMP>
	END	TMI: adbdrv BEGIN 10 <DATE> <TIMESTAMP>
Open Primary	BEGIN	TMI: adbdrv BEGIN 10 <DATE> <TIMESTAMP>
	END	TMI: adbdrv END 10 <DATE> <TIMESTAMP>

Key Alert Log Tags Using X\$DBGALERTEXT

Breaking down the times for the steps in role transition is possible when the TMI event has not been previously set provided the alert log from the role transition is intact. In this case the X\$DBGALERTEXT view can be used to gather the timestamps of key alert log entries (Tables 12-15). This method is not quite as exact as the Time Management Interface but is acceptable in most cases.

Run the following query for each of the following text strings for your situation (11.2 vs 12.1 and failover vs. switchover).

```
SELECT MIN(ORIGINATING_TIMESTAMP)
FROM x$DBGALERTEXT
WHERE MESSAGE_TEXT LIKE '<TXT>'
AND ORIGINATING_TIMESTAMP > <Convert to Standby/Cancel Recovery BEGIN timestamp>;
```

Note: For all role transitions the begin time of the first step (Convert to Standby/Cancel Recovery) is used to drive the other queries. This ensures finding the first instance of the given messages after the start of the role transition.

Note: The queries against X\$DBGALERTEXT are not optimal for large alert logs and could take some time.

TABLE 12: SWITCHOVER STEP DELINEATION TAGS USING X\$DBGALERTTEXT FOR 12.1

Oracle Database 12.1 Switchover Using X\$DBGALERTTEXT		
Convert To Standby (primary alert log)	BEGIN	'SWITCHOVER VERIFY: Send VERIFY request to switchover target%'
	END	'SWITCHOVER: received request 'ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY' from primary database%'
Cancel Recovery (standby alert log)	BEGIN	'Role Change: Canceling MRP - no more redo to apply%'
	END	'Role Change: Canceled MRP%'
Convert to Primary (standby alert log)	BEGIN	'Role Change: Canceled MRP%'
	END	'SWITCHOVER: completed request from primary database%'
Open Primary (standby alert log)	BEGIN	'ALTER DATABASE OPEN%'
	END	'Completed: ALTER DATABASE OPEN%'

TABLE 13: SWITCHOVER STEP DELINEATION TAGS USING X\$DBGALERTTEXT FOR 11.2

Oracle Database 11.2 Switchover Using X\$DBGALERTTEXT		
Convert To Standby (primary alert log)	BEGIN	'ALTER DATABASE COMMIT TO SWITCHOVER TO PHYSICAL STANDBY WITH SESSION SHUTDOWN%'
	END	'ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL%'
Cancel Recovery (standby alert log)	BEGIN	'ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL%'
	END	'Completed: ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL%'
Convert to Primary (standby alert log)	BEGIN	'ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY WAIT WITH SESSION SHUTDOWN%'
	END	'Completed: ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY WAIT WITH SESSION SHUTDOWN%'
Open Primary (standby alert log)	BEGIN	'ALTER DATABASE OPEN%'
	END	'Completed: ALTER DATABASE OPEN%'

TABLE 14: FAILOVER STEP DELINEATION TAGS USING X\$DBGALERTTEXT FOR 12.1

Oracle Database 12.1 Failover Using X\$DBGALERTTEXT (all events in standby alert log)		
Cancel Recovery	BEGIN	'Terminal Recovery requested%';
	END	'Attempt to do a Terminal Recovery%'
Terminal Recovery	BEGIN	'Attempt to do a Terminal Recovery%'
	END	'Terminal Recovery: successful completion%'

Convert to Primary	BEGIN	'Terminal Recovery: successful completion%'
	END	'Switchover: Complete - Database mounted as primary%'
Open Primary	BEGIN	'ALTER DATABASE OPEN%'
	END	'Completed: ALTER DATABASE OPEN%'

TABLE 15: FAILOVER STEP DELINEATION TAGS USING X\$DBGALERTEXT FOR 11.2

Oracle Database 11.2 Failover Using X\$DBGALERTEXT (all events in standby alert log)		
Cancel Recovery	BEGIN	'ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL%'
	END	'Completed: ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL%'
Terminal Recovery	BEGIN	'Completed: ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL%'
	END	'Completed: ALTER DATABASE RECOVER MANAGED STANDBY DATABASE FINISH FORCE%'
Convert to Primary	BEGIN	'Completed: ALTER DATABASE RECOVER MANAGED STANDBY DATABASE FINISH FORCE%'
	END	Completed: ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY WAIT WITH SESSION SHUTDOWN%'
Open Primary	BEGIN	'ALTER DATABASE OPEN%'
	END	'Completed: ALTER DATABASE OPEN%'

Additional Resources

Support notes for switchover best practices for Data Guard Physical Standby (11.2.0.2):

- » If using SQL*Plus, see "11.2 Data Guard Physical Standby Switchover Best Practices using SQL*Plus" in [My Oracle Support Note 1304939.1](#)
- » If using the Oracle Data Guard broker or Oracle Enterprise Manager, see "11.2 Data Guard Physical Standby Switchover Best Practices using the Broker" in [My Oracle Support Note 1305019.1](#)
- » The MAA white paper "Switchover and Failover Best Practices" from the MAA Best Practices area for Oracle Database at <http://www.oracle.com/goto/maa>

For a comprehensive review of Oracle Data Guard failover best practices, see:

- » [Oracle Data Guard Broker](#) for information about Switchover and Failover Operations
- » Data Guard Fast-Start Failover documentation for [Oracle Database 11g](#) and [Oracle Database 12c](#).
- » "Oracle Database 11.2 Data Guard Physical Standby Switchover Best Practices using SQL*Plus" in [My Oracle Support Note 1304939.1](#)
- » "Oracle Database 11.2 Data Guard Physical Standby Switchover Best Practices using Data Guard broker" in [My Oracle Support Note 1305019.1](#)

Conclusion



Role Transitions have become more streamlined and efficient in recent versions of the Oracle Database. The best practices identified in this paper will help you achieve the best possible role transition times. Analyzing where the time is being spent during a role transition can help tune and set proper expectations for your environment.



Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

AUTHOR: ANDREW STEINORTH

CONNECT WITH US

 blogs.oracle.com/oracle

 facebook.com/oracle

 twitter.com/oracle

 oracle.com

Hardware and Software, Engineered to Work Together

Copyright © 2016, Oracle and/or its affiliates. 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 and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0818