This section describes how to perform a failover to a physical standby database.

**Step 1 Flush any unsent redo from the primary database to the target standby database.**

If the primary database can be mounted, it may be possible to flush any unsent archived and current redo from the primary database to the standby database. If this operation is successful, a zero data loss failover is possible even if the primary database is not in a zero data loss data protection mode.

Ensure that Redo Apply is active at the target standby database.

Mount, but do not open the primary database. If the primary database cannot be mounted, go to Step 2.

Issue the following SQL statement at the primary database:

SQL> ALTER SYSTEM FLUSH REDO TO target\_db\_name;

For target\_db\_name, specify the DB\_UNIQUE\_NAME of the standby database that is to receive the redo flushed from the primary database.

This statement flushes any unsent redo from the primary database to the standby database, and waits for that redo to be applied to the standby database.

If this statement completes without any errors, go to Step 5. If the statement completes with any error, or if it must be stopped because you cannot wait any longer for the statement to complete, continue with Step 2.

**Step 2 Verify that the standby database has the most recently archived redo log file for each primary database redo thread.**

Query the V$ARCHIVED\_LOG view on the target standby database to obtain the highest log sequence number for each redo thread.

For example:

SQL> SELECT UNIQUE THREAD# AS THREAD, MAX(SEQUENCE#) OVER (PARTITION BY thread#) AS LAST from V$ARCHIVED\_LOG;

THREAD LAST

---------- ----------

1 100

If possible, copy the most recently archived redo log file for each primary database redo thread to the standby database if it does not exist there, and register it. This must be done for each redo thread.

For example:

SQL> ALTER DATABASE REGISTER PHYSICAL LOGFILE 'filespec1';

**Step 3 Identify and resolve any archived redo log gaps.**

Query the V$ARCHIVE\_GAP view on the target standby database to determine if there are any redo gaps on the target standby database.

For example:

SQL> SELECT THREAD#, LOW\_SEQUENCE#, HIGH\_SEQUENCE# FROM V$ARCHIVE\_GAP;

THREAD# LOW\_SEQUENCE# HIGH\_SEQUENCE#

---------- ------------- --------------

1 90 92

In this example the gap comprises archived redo log files with sequence numbers 90, 91, and 92 for thread 1.

If possible, copy any missing archived redo log files to the target standby database from the primary database and register them at the target standby database. This must be done for each redo thread.

For example:

SQL> ALTER DATABASE REGISTER PHYSICAL LOGFILE 'filespec1';

**Step 4 Repeat Step 3 until all gaps are resolved.**

The query executed in Step 3 displays information for the highest gap only. After resolving a gap, you must repeat the query until no more rows are returned.

If, after performing Step 2 through Step 4, you are not able to resolve all gaps in the archived redo log files (for example, because you do not have access to the system that hosted the failed primary database), some data loss will occur during the failover.

**Step 5 Stop Redo Apply.**

Issue the following SQL statement on the target standby database:

SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE CANCEL;

**Step 6 Finish applying all received redo data.**

Issue the following SQL statement on the target standby database:

SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE FINISH;

If this statement completes without any errors, proceed to Step 7.

If an error occurs, some received redo data was not applied. Try to resolve the cause of the error and re-issue the statement before proceeding to the next step.

Note that if there is a redo gap that was not resolved in Step 3 and Step 4, you will receive an error stating that there is a redo gap.

If the error condition cannot be resolved, a failover can still be performed (with some data loss) by issuing the following SQL statement on the target standby database:

SQL> ALTER DATABASE ACTIVATE PHYSICAL STANDBY DATABASE;

Proceed to Step 9 when the ACTIVATE statement completes.

**Step 7 Verify that the target standby database is ready to become a primary database.**

Query the SWITCHOVER\_STATUS column of the V$DATABASE view on the target standby database. For example:

SQL> SELECT SWITCHOVER\_STATUS FROM V$DATABASE;

SWITCHOVER\_STATUS

-----------------

TO PRIMARY

1 row selected

A value of either TO PRIMARY or SESSIONS ACTIVE indicates that the standby database is ready to be switched to the primary role. If neither of these values is returned, verify that Redo Apply is active and continue to query this view until either

TO PRIMARY or SESSIONS ACTIVE is returned.

**Step 8 Switch the physical standby database to the primary role.**

Issue the following SQL statement on the target standby database:

SQL> ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY WITH SESSION SHUTDOWN;

**Step 9 Open the new primary database.**

SQL> ALTER DATABASE OPEN;

**Step 10 Back up the new primary database.**

Oracle recommends that a full backup be taken of the new primary database.

**Step 11 Restart Redo Apply if it has stopped at any of the other physical standby databases in your Data Guard configuration.**

For example:

SQL> ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING CURRENT LOGFILE DISCONNECT FROM SESSION;

**Step 12 Optionally, restore the failed primary database.**

After a failover, the original primary database can be converted into a physical standby database of the new primary database using the method described in Section 13.2 or Section 13.7, or it can be re-created as a physical standby database from a backup of the new primary database using the method described in Section 3.2.

Once the original primary database is running in the standby role, a switchover can be performed to restore it to the primary role.