Oracle Database 10*g*:
Data Guard Role Transitions and Streams

An Oracle White Paper
February 2005

ORACLE

Oracle Database 10*g*: Data Guard Role Transitions and Streams

INTRODUCTION

Oracle's technology offers many features that administrators can use to meet their High Availability requirements. Two of these technologies are Oracle Data Guard and Oracle Streams. This paper describes the best practices in configuring Data Guard within a Streams environment to provide for additional data and disaster recovery protection of one of the Streams database. The best practices are applicable for both Oracle9*i* and Oracle Database 10*g* databases leveraging Data Guard's physical standby databases. While the testing was not performed using Real Application Clusters for either the primary or standby database, no major issues are foreseen. For the reminder of the paper, the term standby database refers to a physical standby database.

Summary of Key Points

- To achieve seamless Data Guard role transitions that involve a switchover the configuration should
    - Have identical Oracle Network configuration files at all sites
    - Have client failover configured for Oracle Net services
    - Have the same supplemental logging directives enabled on all standbys as they exist on the primary
- Data can be restored to a source database that performs failover by utilizing the destination database.
- Data lost as a result of a failover of a destination database can be recovered using the source database.

CONTRASTING STREAMS AND DATA GUARD

While Streams and Data Guard do share some common underlying infrastructure and technology, they are both independent features that are built to solve different business needs.

Streams is designed for information sharing and distribution but can also provide a very efficient high availability solution. This is accomplished through Streams'

ability to propagate and transform data, transactions, and events from one database to another, independent of database platform, release, or character set. Streams provide the DBA with full control in relation to what information is put into a data stream, how the stream is routed from node to node, what happens to events in the stream as they flow into each node, and how the stream terminates. This allows the DBA to replicate the entire database or just a subset.

Oracle Data Guard is designed for protecting from data failure and disasters. Data Guard does this by providing the ability to create, maintain, manage, and monitor one or more standby database. Physical standby databases use Redo Apply to process the transmitted redo to make the database transactionally consistent to the primary database. In the event of the data failure or disaster, Data Guard can quickly switchover or failover to the standby in minutes.

While both technologies are designed to service different needs, they can complement each other very well. A highly available information sharing solution can be achieved by creating standby databases for either the source or destination databases in a Streams configuration.

STREAMS CONFIGURATION FOR A DATA GUARD CONFIGURATION

Since a standby is an exact duplicate of the primary database, all aspects of a Streams configuration such as capture process, propagation rules, and apply services will be maintained on the standby. In order to assure continuous operation whenever the standby transitions roles to be a primary database, we must account for a few considerations.

- To provide seamless replication through the switchover, the source primary site and standby site must have identical tnsnames.ora files. This assures that all dblinks used by the Streams configuration can correctly resolve Oracle Net service names.
- A Streams configuration requires supplemental logging. The same supplemental logging directives that exist on the primary must be enabled on the physical standby regardless if it was created after supplemental logging was enabled on the primary. To verify the current state of supplemental logging on the standby run the following query:

```
SQL> SELECT SUPPLEMENTAL_LOG_DATA_MIN,
  SUPPLEMENTAL_LOG_DATA_PK,
  SUPPLEMENTAL_LOG_DATA_UI,
  SUPPLEMENTAL_LOG_DATA_FK,
  SUPPLEMENTAL_LOG_DATA_ALL FROM V$DATABASE;
```

- To enable supplemental logging at the database level run the following command on the standby while at the mount state:

```
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG
  DATA (PRIMARY KEY,UNIQUE INDEX,FOREIGN
  KEY,ALL) COLUMNS;
```

- All dblinks used by the Streams configuration should utilize Oracle Net service names that are configured for connect time failover. Specifically, the Oracle Net service name that the source database uses to propagate changes to the destination database should be configured for connect time failover as such:

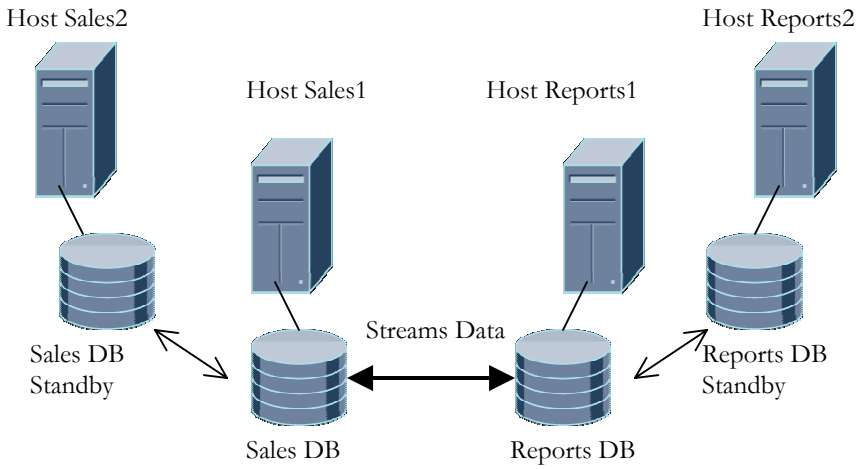
```
SALES =
(DESCRIPTION =
 (ADDRESS_LIST=
    (ADDRESS = (PROTOCOL = TCP)(HOST = Sales1)
      (PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP)(HOST = Sales2)
      (PORT = 1521)))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = SALES)
  )
)
```

With the above Oracle Net service name a connection attempt is made using the first host in the address list. If the connection does not complete then a second attempt is made using the second host in the address list. While this simple modification solves the majority of the issue, there are other issues that must be addressed which are covered below.

STREAMS CONSIDERATIONS DURING DATA GUARD ROLE TRANSITIONS

A database can operate in one of two modes in a Data Guard configuration; primary or standby. Role transitions occur when administrators invoke either a switchover or a failover operation.

A switchover allows a primary and standby to reverse roles without any data loss and without any need to reinstantiate the previous primary. In contrast, a failover may result in data loss and require reinstantiation in Oracle9*i* or a flashback database in Oracle Database 10*g*. The following sections will discuss Streams considerations when a role transition occurs. The following is a diagram of our example configuration:



Performing a switchover in a Streams environment

In order to assure a successful transition the following actions should have been completed prior to performing the switchover (as described in the previous section):

- Enable supplemental logging on the standby prior to the switchover
- Identical Oracle Net configuration files on the primary and standby hosts
- Configure client failover capabilities

When performing a switchover of a destination database or one of the source databases in a multi sourced configuration the remaining databases in the configuration must be able to send changes to the new primary. When a switchover of the database occurs the primary database (SALES) running on the primary host (Sales1) becomes a standby and the standby database (SALES_PS) running on the standby host (Sales2) becomes a primary. The new standby (SALES) running on the old primary host (Sales1) is normally at the mount standby with media recovery actively applying changes from the new primary. When the connection from the source database (REPORTS) not involved in the switchover is made it will first try the first host in the address list, in this case the old primary host (Sales1). The connection will get the following error:

```
ORA-01033: ORACLE initialization or shutdown in progress
```

The above error is normal for any non-sysdba connection attempt to a mounted database. Unfortunately connect time failover will not be invoked with the ORA-1033 error so a connection to the next host in the address list (Sales2) will not be attempted. This issue can be resolved by using a primary specific service name. Perform the following steps given that the primary database service name is SALES:

1. On the primary database (SALES) that is to become the standby run the following command:

SQL> alter system set service_names = 'SALES_PS' scope=spfile;

2. On the standby database (SALES_PS) that is to become the new primary run the following command:

   SQL> alter system set service_names = 'SALES' scope=spfile;

3. Initiate the switchover by performing the following command on the current primary (SALES):

   SQL> alter database commit to switchover to standby;

4. Once the above command completes shutdown the old primary and bring it to the mount state as the new standby. Not that when it comes up as the new standby it will be registering a service name of SALES_PS.

5. On the old standby (SALES_PS) run the following command to convert the database to the new primary:

   SQL> alter database commit to switchover to primary;

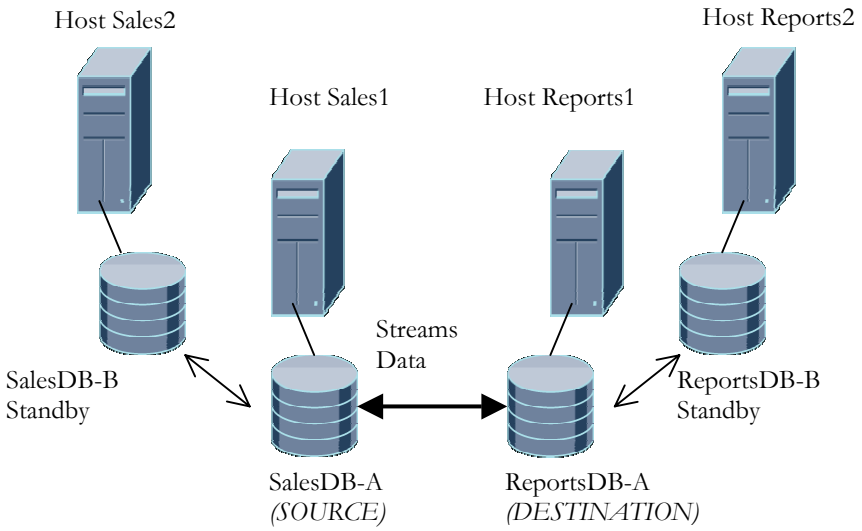
6. Once the above command completes shutdown the old standby and bring it to the open state as the new primary. Note that when it comes up as the new primary it will register a service name of SALES.

When the other databases in the Streams configuration (REPORTS) attempt to make a connection to the database on the old primary host (Sales1) they will receive an error as the SALES service name does not exist with that listener. Using client failover setup in the previous step, a second connection attempt will be made on the old standby host (Sales2) where it will find the correct service name.

Using the above SQL commands during the switchover will allow the propagation process to continue without any modifying any dblinks or TNS aliases.

Recovering from a failover of the destination database

If the destination database has a Data Guard physical standby configuration and a failover is performed that results in data loss then you must reapply captured changes that had already been applied after the recovery point in time. This can be accomplished by either resetting the SCN for the existing capture or by creating a new capture process on the source. The following discussion covers resetting the SCN for an existing capture in a single source environment that is not using directed networks. Consider this diagram of our configuration:



1. Following a failover of the destination database (such that ReportsDB-B is now the new primary), drop the propagation that propagates changes from the source (SalesDB-A) to the old primary destination database (ReportsDB-A).

   SQL> select propagation_name from dba_propagation;

   ```
   PROPAGATION_NAME
   ------------------------------
   STRMADMIN_PROPAGATE
   ```

   SQL> exec dbms_propagation_adm.drop_propagation ('STRMADMIN_PROPAGATE');

2. Query the new primary destination database (ReportsDB-B) database and determine the highest applied SCN (oldest message number) from DBA_APPLY_PROGRESS.

   SQL> select oldest_message_number from dba_apply_progress;

   ```
   OLDEST_MESSAGE_NUMBER
   ---------------
    697731
   ```

3. On the source database (SalesDB-A) stop the capture process.

   SQL> select capture_name from dba_capture;

   ```
   CAPTURE_NAME
   ------------------------------
   STRMADMIN_CAPTURE
   ```

   SQL> exec dbms_capture_adm.stop_capture (STRMADMIN_CAPTURE');

4. On the source database (SalesDB-A), using the `DBMS_CAPTURE_ADM` package reset the start SCN for an existing capture process to the value obtained in step 2.

```
BEGIN
    DBMS_CAPTURE_ADM.ALTER_CAPTURE(
    capture_name => 'STRMADMIN_CAPTURE',
    start_scn    => 697731);
    END;
    /
```

5. As the streams administrator create a new propagation process on the source database (SalesDB-A) that uses the same rule set as the previous propagation.

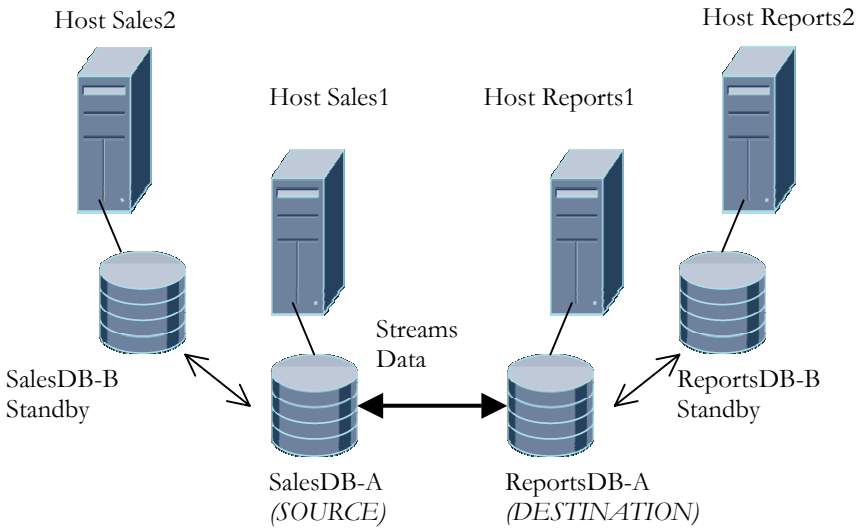
```
BEGIN
    DBMS_PROPAGATION_ADM.CREATE_PROPAGATION(
    propagation_name   => 'PROPAGATION',
    source_queue       => 'STREAMS_QUEUE',
    destination_queue  => 'STREAMS_QUEUE',
    destination_dblink => 'SALES',
    rule_set_name      => 'RULESET$_78');
    END;
    /
```

6. As the Streams administrator start the capture process on the source database (SalesDB-A).

```
SQL> exec dbms_capture_adm.start_capture
    ('STRMADMIN_CAPTURE');
```

Recovering from a data loss failover of the source database

Consider our sample configuration:



If the source database (SalesDB-A) has a Data Guard physical standby configuration and a failover is performed (such that SalesDB-B is the new primary), that results in data loss then it is possible for the destination (ReportsDB-A or ReportsDB-B) to have transactions that the new primary source database (SalesDB-B) does not. In order to resolve this conflict we must manually enter the missing transaction and restart the Streams process.

1. Perform failover to the standby (SalesDB-B) of the source database (SalesDB-A).
2. Open the new source database (SalesDB-B) in restricted mode and make note of the point in time recovery SCN.

   SQL> startup restrict;

   SQL> select STANDBY_BECAME_PRIMARY_SCN from v$database;

```
STANDBY_BECAME_PRIMARY_SCN
--------------------------
                    671698
```

3. On the new source database (SalesDB-B), stop the capture process using the `STOP_CAPTURE` procedure in the `DBMS_CAPTURE_ADM` package.

   SQL> select capture_name,status from dba_capture;

```
CAPTURE_NAME                   STATUS
------------------------------ --------
STRMADMIN_CAPTURE              ENABLED
```

   SQL> exec dbms_capture_adm.stop_capture (capture_name =>'STRMADMIN_CAPTURE');

4. On the source database (SalesDB-B), perform a data dictionary build:

```
SET SERVEROUTPUT ON
    DECLARE
    scn  NUMBER;
    BEGIN
    DBMS_CAPTURE_ADM.BUILD(
    first_scn => scn);
    DBMS_OUTPUT.PUT_LINE('First SCN Value = ' || scn);
    END;
    /
```

   In our example the above procedure returned the following output:

```
First SCN Value = 688604
```

5. Assure that all transaction have been applied at the destination database (ReportsDB-A). This can be done by waiting for the state of the apply process to become idle as seen by querying the `STATE` column in the `V$STREAMS_APPLY_SERVER view`.

6. Perform a query at the destination database (ReportsDB-A) to determine the highest SCN for a transaction that was applied. If the apply process is running, then perform the following query:

```
SQL> select hwm_message_number from
    v$streams_apply_coordinator where apply_name =
    'STRMADMIN_SALES_APPLY';

    HWM_MESSAGE_NUMBER
    ------------------
                679027
```

7. If the apply process is disabled, then perform the following query in the destination database (ReportsDB-A):

```
SQL> select applied_message_number from dba_apply_progress
    where Apply_name = 'STRMADMIN_SALES_APPLY';

    APPLIED_MESSAGE_NUMBER
    ----------------------
                679027
```

8. If the highest apply SCN obtained in step 6 is greater than or equal to the STANDBY_BECAME_PRIMARY_SCN step 1 then the apply process has applied transactions beyond what the new source database (SalesDB-B) has. If the SCN in step 6 is less than the STANDBY_BECAME_PRIMARY_SCN from step 1, then refer to Section titled "*Performing Point-in-Time Recovery on the Source in a Single Source Environment*" in the Oracle Streams Replication Administrator's Guide 10g Release 1 (10.1) for more details.

9. On the source database (SalesDB-B) set a Streams tag in the current session so that any transactions entered on the source database (SalesDB-B) will not be captured by the capture process.

```
BEGIN
    DBMS_STREAMS.SET_TAG(
    tag  =>  HEXTORAW('1D'));
    END;
    /
```

10. Manually execute transactions applied after the STANDBY_BECAME_PRIMARY_SCN at the source database (SalesDB-B).

11. Disable the restricted session at the source database (SalesDB-B).

12. Drop the original capture process on the source database (SalesDB-B).

```
SQL> exec dbms_capture_adm.drop_capture
    ('STRMADMIN_CAPTURE');
```

13. On the source database (SalesDB-B) create a new capture process using the CREATE_CAPTURE procedure in the DBMS_CAPTURE_ADM package to replace the capture process you dropped in Step 11. Specify the SCN returned by the data dictionary build in Step 4 for both the first_scn and start_scn parameters. As the Streams administrator user:

```
BEGIN
    DBMS_CAPTURE_ADM.CREATE_CAPTURE(
    queue_name        => 'STREAMS_QUEUE',
    capture_name      => 'STRMADMIN_CAPTURE',
    rule_set_name     => 'RULESET$_63',
    start_scn         => 688604,
    source_database   => NULL,
    use_database_link  => false,
    first_scn         => 688604);
    END;
    /
```

14. Start the new capture as the Streams administrator user on the source database (SalesDB-B):

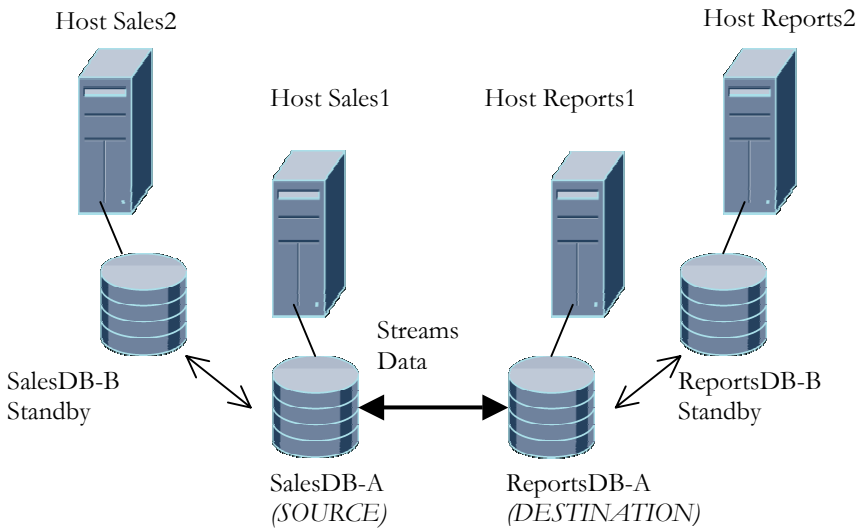
```
SQL> exec dbms_capture_adm.start_capture
    ('STRMADMIN_CAPTURE');
```

Recovering from a data loss failover in a multi-source configuration

Streams does support recovering from data loss in a 2-way bidirectional replication by utilizing the DBMS_STREAMS_ADM.GET_SCN_MAPPING package. Please refer to Section titled "*Performing Point-in-Time Recovery in a Multiple Source Environment*" in the Oracle Streams Replication Administrator's Guide 10g Release 1 (10.1) for more details.

Switchover or Failover of a downstream capture site

Again, consider our sample configuration:



When downstream capture is configured, changes generated on the source database (SalesDB-A) are sent to a downstream database (ReportsDB-A) (most commonly a destination database) where a downstream capture process mines DML and DDL. Streams uses the same mechanism as Data Guard log transport services to transfer changes to the downstream database (ReportsDB-A). An example of the remote archive destination set on the source database (SalesDB-A) for downstream capture is as follows:

LOG_ARCHIVE_DEST_2='SERVICE=SALES ARCH OPTIONAL NOREGISTER REOPEN=60 TEMPLATE = /u01/oradata/SALES/arch/_arch_%t_%s.log'

If, in the above example, the SALES Oracle Net service name is configured for connect time failover, a primary specific service name is used, and the same

directory exists on both primary and standby hosts then no changes are necessary to continue shipping to a downstream site (ReportsDB-B) that has performed either a switchover or failover.

CONCLUSION

Data Guard is an excellent disaster recovery solution for a Streams configuration. With proper planning and implementation very little to no changes are necessary to have a Streams configuration continue to operate when a switchover of either the source or destination databases occur. In the event of a failure it is possible to resync the source and destination database thus preventing the need to rebuild the streams environment.



Oracle Database 10*g*: Data Guard Role Transitions and Streams
Feburary 2005
Author: Michael Smith
Contributing Authors: Lawrence To, Ashish Ray
Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

Oracle is a registered trademark of Oracle Corporation. Various product and service names referenced herein may be trademarks of Oracle Corporation. All other product and service names mentioned may be trademarks of their respective owners.

Copyright © 2005 Oracle Corporation
All rights reserved.