

Oracle Streams Configuration
Best Practices: Oracle Database
10g Release 10.2

*An Oracle White Paper
August 2008*

Maximum Availability Architecture

Oracle Best Practices for High Availability

ORACLE

Oracle Streams Configuration Best Practices: Oracle
Database 10g Release 2

Introduction.....3
Overview of Oracle Streams Architecture and Processing.....3
 Capture Component.....4
 Propagation Component.....5
 Apply Component.....5
Oracle Streams Configuration Best Practices5
 Guidelines for Preparing All Streams Configurations.....6
 Recommendations for Downstream Capture Configurations12
 Recommendations for Upstream (Local) Capture Configurations18
Tasks to Perform After Configuring Streams21
Troubleshooting Streams Configurations.....23
 Repairing the Initial Configuration.....23
 Repairing Common Runtime Errors.....25
 Investigate Errors for the Capture Process.....26
 Investigate Errors for the Propagation Process.....28
 Investigate Errors for the Apply Process.....29
Conclusion.....32
Appendix A: Using Streams Configurations Over a Network.....33
 Network Tuning Best Practices.....33
 Set TCP/IP Network Parameters On All Servers.....33
 Oracle Net Session Data Unit (SDU) Size.....34
 TCP Socket Buffer Sizes.....34
 Network Device Queue Sizes.....36
 Standby Redo Log I/O Tuning for Downstream Database.....37
 Use Service Names in Oracle Net Alias Descriptors.....38
Appendix B: Monitoring Streams Configuration Progression.....38
References.....39

Businesses share information in a variety of ways, including disseminating price lists to field offices, consolidating sales data from multiple stores, provisioning data in a grid environment, or notifying a payroll application of a new employee. Using Oracle Streams, businesses can share information in near real time and provide improved availability and scalability by synchronizing multiple copies of data.

INTRODUCTION

The [Maximum Availability Architecture \(MAA\)](#) [1] is a best practices blueprint for achieving high availability using Oracle technologies. This MAA white paper describes best practices for Oracle Streams configurations for both downstream capture and upstream (local) capture. The paper contains the following sections:

- [Overview of Oracle Streams Architecture](#)
- [Downstream Capture Configuration](#)
- [Upstream Capture \(Local Capture\) Configuration](#)
- [Tasks to Perform After Configuring Streams](#)
- [Troubleshooting Streams Configurations](#)
- [Recommendations for Streams Configurations Used Over a Network](#)

The MAA testing described in this white paper used Oracle Streams with Oracle Database 10g Release 2 (10.2.0.4). Also, see the “*Oracle Streams Performance, Best Practices: Oracle Database 10g Release 2*” [11] MAA white paper that provides practical tuning and troubleshooting techniques for Streams configurations.

OVERVIEW OF ORACLE STREAMS ARCHITECTURE AND PROCESSING

Oracle Streams enables the propagation and management of data, transactions, and events in a data stream either within a database, or from one database to another. Oracle Streams consists of three components: capture, propagation, and apply. Each of these components is made up of several Operating System processes, as shown in Figure 1.

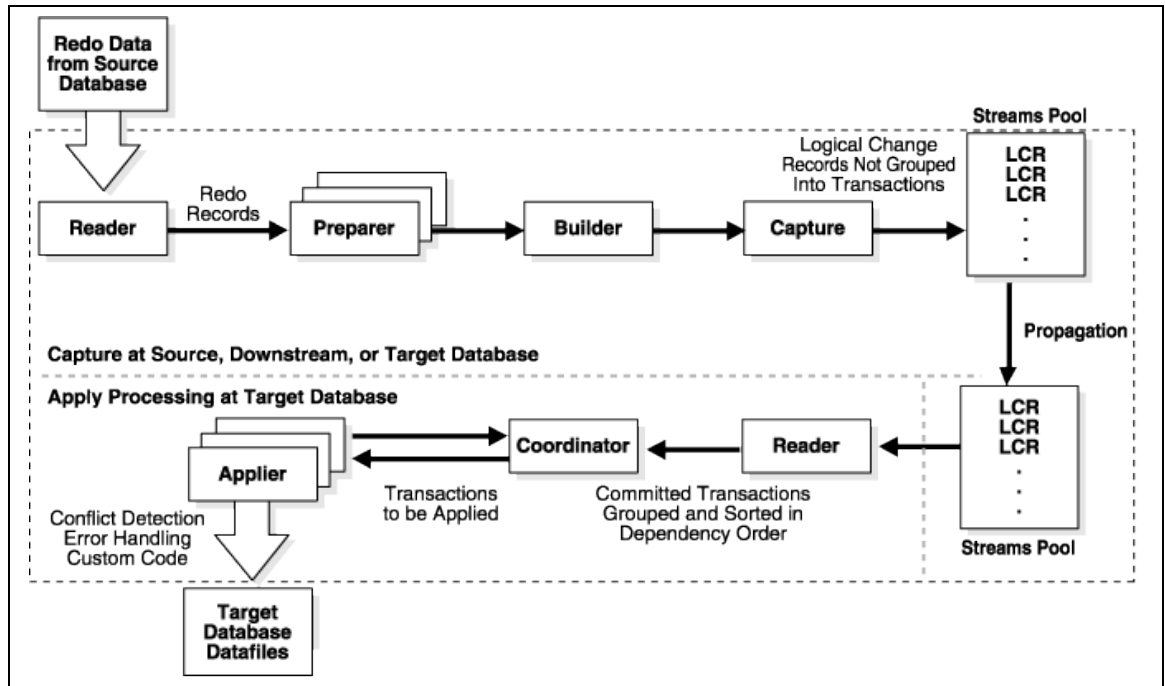


Figure 1: Oracle Streams Processing Architecture

Capture Component

The capture component mines redo logs from a source database and captures changes. It consists of multiple operating-system processes. A reader process reads the redo log and divides the redo log into regions. The preparer processes scan the regions defined by the reader in parallel and pre-filter changes found in the redo log based upon user-defined rules. A builder process merges redo records from the preparers and passes the merged redo records to the capture process. The capture process then formats each change into a Logical Change Record (LCR) and enqueues to a queue if it satisfies the defined rules.

You can configure capture locally at a source database or remotely at a downstream database. A local capture process runs at the source database and captures changes from the local source database redo log. You can configure a downstream capture process for real-time capture in which redo data from source database is transmitted to the downstream database and written to the *standby redo log files*, from which the capture process captures changes.

A single capture process can send changes to multiple propagation and apply processes. You can also configure multiple capture processes.

Propagation Component

The propagation component uses queues to transfer the LCRs from the source database to the destination database.

Apply Component

The apply component receives the LCRs and applies the changes at the target database. The **reader** process dequeues LCRs from a queue, assembles the changes into transactions, calculates the transaction dependencies, and passes the transactions to the coordinator. The **coordinator** process assigns transactions to available appliers, based on the transaction dependencies and ordering. The **applier** process executes the changes at the target database. You can configure multiple appliers to increase apply parallelism.

ORACLE STREAMS CONFIGURATION BEST PRACTICES

This section recommends best practice for configuring Streams for Oracle Database 10g Release 2. The discussions are divided into the following sections:

- [Guidelines for Preparing All Streams Configurations](#)
- [Recommendations for Downstream Capture Configurations](#)
- [Recommendations for Upstream \(Local\) Capture Configurations](#)

All users should perform the instructions in the [Guidelines for Preparing All Streams Configurations](#) section of this white paper before branching off to either the downstream or the upstream configuration recommendations.

The examples in this paper are based on a configuration that uses:

- Two databases
- Stream database administrator privileges
- Automatic Storage Management (ASM).

If your environment does not use ASM, then you must adjust the examples to use standard file system file names, raw device targets (or logical unit numbers (LUNs)), or raw volume names.

Note: Oracle recommends that you test all configuration recommendations prior to deploying Streams in a production environment.

Guidelines for Preparing All Streams Configurations

The following list summarizes the tasks you must perform to properly prepare each Streams database:

- [Use Oracle Database 10g release 2 \(10.2.0.4\) and apply all relevant patches](#)
- [Verify platform requirements for downstream capture](#)
- [Prepare the source and target database redo logs for Streams](#)
- [Create a tablespace dedicated to Streams](#)
- [Create the Streams administrator database user](#)
- [Grant Streams authorization and DBA privileges](#)
- [Set key initialization parameters](#)
- [Create database links between the source and target databases](#)
- [Set Up Directory Object](#)
- [Account for object or tablespace name differences when replicating DDLs](#)

Use the following checklist to perform the preparation tasks:

Use Oracle Database 10g release 2 (10.2.0.4)

Ensure that all Streams databases are running Oracle Database 10g Release 2 (release 10.2.0.4 is recommended) and apply any critical patch sets. Although the source database must be running Oracle Database 10g Release 2, the downstream capture database can run Release 10.2.0.4 or a later release.

See the [Oracle Database Upgrade Guide](#) [7] for database upgrade information and the “Patches and Updates” tab on [OracleMetalink](#) for information about critical Streams patches for release 10.2 and higher releases.

Verify platform requirements for downstream capture

Downstream capture requires that the source and destination database are running on the same platform.

Prepare the source and target databases redo logs for Streams

a) Configure the source and target databases in ARCHIVELOG mode.

For one-way Streams local capture, you must configure the source database to run in ARCHIVELOG mode, but the best practice is for both the source and target databases to be in ARCHIVELOG mode in case media recovery is required for either of the databases. If replication is bi-directional, then you must configure both source and target databases in ARCHIVELOG mode.

b) Configure the local archive destination, LOG_ARCHIVE_DEST_1, parameter and do not use a flash recovery area.

Maximum Availability Architecture

Streams local capture and downstream capture require online redo logs and archived redo logs. Do not place archived redo log files in the Flash Recovery area, because this is a fixed-size storage area and the files may be deleted if the amount of space becomes low.

❑ Create a tablespace dedicated to Streams

Create a dedicated Streams tablespace to contain the Streams AQ tables and other objects related to Streams, and to allow for better space management for the Streams objects:

- For downstream capture, create the Streams tablespace on the downstream capture database only.
- For upstream capture, create the Streams tablespace on both the source and target databases.

For example, to create a Streams tablespace of 50 MB, which is the minimum recommended size, issue the following statements:

```
CREATE TABLESPACE streams_ts
  LOGGING
  EXTENT MANAGEMENT LOCAL AUTOALLOCATE
  SEGMENT SPACE MANAGEMENT AUTO
  DATAFILE '+ENG' SIZE 50M
  AUTOEXTEND ON NEXT 50M MAXSIZE UNLIMITED ;
```

❑ Create the Streams administrator database user

Create the Streams administrator account that you will use to create and modify all Streams configurations and perform administrator functions with the DBMS_STREAMS_ADM PL/SQL package:

For example, to create a new user named `streamsadmin` with the default `streams_ts` tablespace, enter the following SQL*Plus statement as SYSDBA:

```
create user streamsadmin identified by streamsadmin
  default tablespace streams_ts
  temporary tablespace temp
  /
```

❑ Grant Streams authorization and DBA privileges

Use the DBMS_STREAMS_AUTH PL/SQL procedure to grant all of the privileges necessary for the Streams package procedures to function properly:

```
execute
DBMS_STREAMS_AUTH.GRANT_ADMIN_PRIVILEGE( 'STREAMSADMIN' );
grant DBA to streamsadmin;
```

Maximum Availability Architecture

□ Set key initialization parameters

On each Streams database in the configuration, specify the initialization parameters shown in [Table 1](#).

Table 1: Initialization Parameters for Streams Database Names and Services

Parameter Name	Description
AQ_TM_PROCESSES	<p>Do not explicitly set this parameter to 0 or 10. Doing so could disable the queue monitoring processing and impact the Streams pool memory utilization.</p> <ul style="list-style-type: none"> • If this parameter has not been set, the parameter will be autotuned. • If this parameter has been explicitly set to 0 or 10, either in the initialization parameter file or by the ALTER SYSTEM statement, then reset the parameter to 1.
DB_NAME	<p>On each Streams database, specify the name that was given to the database when it was first created. For example, if your source database is called STREAMSS and your target database is called STREAMSD, be sure to set the DB_NAME parameter on each database accordingly.</p> <p>To determine the value of DB_NAME, log into SQL*Plus as SYSDBA. Issue the following query to determine the name:</p> <pre>SELECT NAME FROM V\$DATABASE;</pre>
DB_DOMAIN	<p>On each Streams database, specify the network domain where the database resides. For example, if the source database is in mydomainA.com domain and the target database is mydomainB.com domain, then ensure that the DB_DOMAIN parameter is set to the correct domain. For example, on the source database, specify DB_DOMAIN=mydomainA.com and on the target database, specify DB_DOMAIN=mydomainB.com.</p>
GLOBAL_NAME=TRUE	<p>On each Streams database, specify GLOBAL_NAMES=TRUE to ensure that each database link used by Streams has the same name as the destination's database GLOBAL_NAME to which the link points.</p>
COMPATIBLE=10.2	<p>On each Streams database, set the COMPATIBLE parameter to "10.2" to use the new features available with Oracle Database 10g release 2 (10.2). Note that once you set COMPATIBLE to 10.2, you can no longer downgrade to your previous release.</p>
JOB_QUEUE_PROCESSES=4	<p>On each Streams database, specify JOB_QUEUE_PROCESSES=4 (the minimum recommended value) to start and use four job queue processes (<i>jnnn</i>) for Streams propagation jobs. If this parameter is already set, then increment the JOB_QUEUE_PROCESSES parameter</p>

Maximum Availability Architecture

	by the number of independent Stream propagations.
<code>_JOB_QUEUE_INTERVAL=1</code>	On each Streams database, specify how many seconds the job queue is scanned for work. It is recommended that you set this parameter to 1 (seconds) to improve the propagation job performance and minimize the delay for how often propagation jobs will execute.
<code>TIMED_STATISTICS=TRUE</code>	On each Streams database, set the <code>TIMED_STATISTICS=TRUE</code> parameter to allow performance statistics to be gathered for analysis of potential performance bottlenecks. Setting this parameter may have some performance impact. However, it is necessary to collect the statistics required to conduct performance tuning once Streams is running.
<code>STATISTICS_LEVEL=TYPICAL</code>	On each Streams database, set this parameter to <code>TYPICAL</code> to collect the necessary statistics. This parameter works with the <code>TIMED_STATISTICS</code> parameter to determine where performance issues occur when Streams is running. Set both the <code>TIMED_STATISTICS</code> and <code>STATISTICS_LEVEL</code> parameters because: <ul style="list-style-type: none"> • The performance impact is minimal for both. • The Automatic Workload Repository (AWR) report and Active Session History (ASH) reports depend on these parameters. <p>Note: Ensure that AWR is installed and running on each Streams database.¹</p>
<code>SHARED_POOL_SIZE=256M</code>	On each Streams database, set this parameter to a minimum value of 256 MB because Streams uses a significant amount of PL/SQL and library cache components. Future performance analysis may provide recommendations for how much to increase the size of <code>SHARED_POOL_SIZE</code> , especially for the database running Stream apply processes. Also, see the “ <i>Oracle Streams Performance and Troubleshooting Best Practices: Oracle Database 10g Release 2</i> ” [11] MAA white paper that provides practical tuning and troubleshooting techniques for Streams configurations.
<code>STREAMS_POOL_SIZE=256M</code>	On each database where there is either a capture or apply process running, set the <code>STREAMS_POOL_SIZE</code> parameter to a minimum of 256M. Memory from the Streams pool is used by both the Streams processes and the buffered queue, with the LCRs in the buffered queue the largest component. Use the <code>V\$STREAMS_POOL_ADVICE</code> view to help you size the buffered queue. <p>For downstream capture, do not set the <code>STREAMS_POOL_SIZE</code></p>

¹ Automatic Workload Repository (AWR) requires that you purchase a license for the Oracle Diagnostics Pack. See the [Oracle Enterprise Manager Licensing Information](#) [12] documentation for more information.

Maximum Availability Architecture

	<p>parameter on the source database if there are no capture processes configured.</p> <p>For local capture, set the <code>STREAMS_POOL_SIZE</code> parameter on the source and target databases.</p>
--	---

❑ Create database links between the source and target databases

Before creating database links, ensure that all Streams databases are reachable using Oracle Net aliases. See [Appendix A: Recommendations for Streams Configurations Used Over a Network](#) for further details about network configurations.

Perform the following steps to create database links:

1. Log into the target database as the Streams administrator user and determine the global database name by issuing the following query:

```
select * from global_name;
```

You will use the name returned by this query to create the database link in the next step.

2. Log into the source database as the Streams administrator, and create a database link first from the source database to the target database.

For example, on the source database, enter the following statement to create a database link from the source database to a target database named `streamsd.us.oracle.com` using the `TNSNAMES.ORA` alias entry `'streamsdest10g_halinux06.us.oracle.com'`:

```
SQL> create database link streamsd.us.oracle.com
connect to streamsadmin identified by
    streamsadmin
    using
    'streamsdest10g_halinux06.us.oracle.com';
```

3. Validate the database link is working by issuing the following `select * from dual@<db_link_name>` query on the source database:

```
select * from dual@streamsd.us.oracle.com;
```

If an error is returned, it is probably due to an incorrect `TNSNAMES.ORA` descriptor or to an incorrect database name. Do not proceed to the next step until the database link is working properly.

4. Log into the Streams administrator account on the target database.
5. Create a database link from the target database to the source database.

Maximum Availability Architecture

For example, enter the following statements on the target to create a database link to the source database `streamss.us.oracle.com`:

```
SQL> create database link streamss.us.oracle.com
connect to streamssadmin identified by streamssadmin
using 'streamssrc10g_halinux02.us.oracle.com';
```

Note: In the context of this white paper, the database link from the destination to the source database is required for downstream capture. For local capture, the database link from the destination to the source database is optional. For more details, see the section titled "Create One or More Database Links" in chapter 6 of the [Oracle Streams Replication Administrator's Guide 10g Release 2 \(10.2\)](#) [13].

6. Validate the database link is working by issuing the following `select * from dual@<db_link_name>` query on the target database:

```
select * from dual@streamss.us.oracle.com;
```

If an error occurs, it is probably because of an incorrect `TNSNAMES.ORA` descriptor or an incorrect database name. Do not proceed to the next section until the database link is working properly.

❑ Set up Directory Objects

In many cases, Oracle Streams instantiates the objects for which changes will be captured and uses Data Pump to initially populate the replica tables at the target database. As a part of this instantiation process, Data Pump needs a file system directory location on both the source and target database server for staging and moving all Streams objects. Streams uses Oracle Directory objects that point to these file system directory locations.

Directory objects are needed by Streams for instantiation purposes. Ensure that the locations to which the directory objects point have sufficient storage space to support all objects that Streams instantiates. On both the source and downstream databases servers, determine an appropriate location for the file system directory and create Oracle directory objects on each of the databases. For example, as the Streams administrator user, create the Oracle Directory objects as follows²:

```
CREATE DIRECTORY streams_dir as
'/home/oracle/projects/workspace';
```

❑ Account for object or tablespace name differences when replicating DDLs

² The directory objects used in the example are file system directories. You can also specify ASM as a target of the directory object, because ASM is supported by Oracle Data Pump. You can also instantiate Streams objects over the network. See the [Oracle Database PL/SQL Supplied Packages and Types Reference](#) [4] documentation for information about using the `INSTANTIATION` parameter on the `DBMS_STREAMS_ADM.MAINTAIN_SCHEMAS` package.

Maximum Availability Architecture

If you choose to replicate DDLs, then you need to account for object or tablespace name differences between the source and target databases. Here are some best practices:

- Avoid system-generated naming for constraints or indexes
- Keep the same tablespace names between databases or use a DDL handler to explicitly address the naming differences

For more information, see the “[Considerations for Applying DDL Changes](#)” section in [Oracle Streams Replication Administrator's Guide 10g Release 2 \(10.2\)](#) [13].

Recommendations for Downstream Capture Configurations

Downstream capture is typically used to replicate a full database, to offload processing from the source database to the target database, or to reduce data loss with ASYNC or SYNC redo transport.

Note: For upstream configuration recommendations, see the [Recommendations for Upstream Local Capture](#) section in this white paper.

This section describes how to prepare the source database to ship redo to a downstream database, and prepare the target database to receive and apply the redo. The steps in this section set up the source and target databases as follows:

- Configures a Streams administration user
- Configures a database link pointing to the downstream database
- Configures real-time redo log shipping configured from the source database to the downstream database
- Enables supplemental logging on schema objects
- Configures an apply process on the target database for applying the changes from the source database

Note: A Streams capture process is not run on the source database.

Perform the following steps to prepare the source and target databases:

1. **Ensure the configuration meets all of the requirements in the “[Guidelines for Preparing All Streams Configurations](#)” section.**

2. Specify initialization parameters on the source and target databases.

[Table 2](#) shows the parameter settings for the example configuration. See the *Oracle Database Reference 10g Release 2 (10.2)* [9] for complete information about each initialization parameter. Note that the names in [Table 2](#) are for the purpose of example only; be sure that the parameters you specify match the locations and parameter values in your Streams configuration.

Table 2: Recommended Initialization Parameter Settings for Downstream Capture

Source Database	Target (Downstream) Database
<pre>LOG_ARCHIVE_DEST_1= 'LOCATION=+ENG/streamssrc10g/ VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) MANDATORY' LOG_ARCHIVE_DEST_2= 'SERVICE=streamsdest10g_halinux06.us.oracle.com LGWR ASYNC NOREGISTER VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=streamsdest10g' LOG_ARCHIVE_DEST_STATE_1=ENABLE LOG_ARCHIVE_DEST_STATE_2=DEFER³ LOG_ARCHIVE_CONFIG='SEND, DG_CONFIG=(streamssrc10g,streamsdest10g)'⁴ LOG_ARCHIVE_MAX_PROCESSES=4⁵</pre>	<pre>LOG_ARCHIVE_DEST_1= 'LOCATION=use_db_recovery_file_dest VALID_FOR=(ONLINE_LOGFILE,PRIMARY_ROLE)' LOG_ARCHIVE_DEST_2= 'LOCATION=+ENG/streamsdest10g/arch_srl_streamss/ VALID_FOR=(STANDBY_LOGFILE,PRIMARY_ROLE)' LOG_ARCHIVE_DEST_STATE_1=ENABLE LOG_ARCHIVE_DEST_STATE_2=ENABLE⁶ LOG_ARCHIVE_CONFIG='RECEIVE, DG_CONFIG=(streamssrc10g,streamsdest10g)' LOG_ARCHIVE_MAX_PROCESSES=4</pre>

3. On the downstream database, configure standby redo logs.

To enable real-time mining for capture at the downstream database, you must configure standby redo logs (SRLs) on the downstream database to accept redo from the source database. The Streams capture process mines the standby redo logs and any archived redo logs created from the standby redo logs.

³ On the source database, set the LOG_ARCHIVE_DEST_STATE_2 parameter to DEFER until the target downstream capture database is configured to prevent the source database from sending any redo data. Reset this parameter to ENABLE after the downstream database is configured.

⁴ On the source and target databases, set the DG_CONFIG qualifier to the source database's DB_UNIQUE_NAME and the target database's DB_UNIQUE_NAME.

⁵ On the source database, specify a minimum of four background processes to perform archiving and transmission of archived redo logs to the target database. Set this parameter to a minimum of 4 because there are two archived redo log destinations configured on the database.

⁶ On the downstream database, set both the LOG_ARCHIVE_DEST_STATE_1 and LOG_ARCHIVE_DEST_STATE_2 parameters to ENABLE.

Maximum Availability Architecture

To configure SRLs, perform the following steps on the downstream database:

- a) **On the source database, query the V\$LOG view to find the number of online redo log groups, and use the following equation to determine the number of SRLs:**

Number of SRLs = sum of all production online redo log groups for each thread + number of threads

See Also: The following sections in the [Oracle Database High Availability Best Practices](#) [10] documentation:

- The section titled “Use Standby Redo Logs and Configure Size Appropriately” for information about configuring and sizing standby redo logs appropriately
- The section titled “Storage Recommendations” for information about when the standby redo logs and online redo logs are active

- b) **Add standby redo logs to the downstream database.**

The following example shows how to add standby redo logs to a Streams configuration that uses ASM and for which the log file members are explicitly specified under ASM. The standby redo log files are separate from the online redo log files.

```
ALTER DATABASE ADD STANDBY LOGFILE
  ( '+ENG/streamsdest10g/standbylog/srl_m1.dbf' ,
    '+ENG/streamsdest10g/standbylog/srl_m2.dbf' )
SIZE 1024M;
```

- c) **Query the V\$STANDBY_LOG view to validate the log groups and the status. For example:**

```
SELECT GROUP#, SEQUENCE#, STATUS FROM V$STANDBY_LOG;
```

Initially, the `sequence#` is set to 0 and the status is set to UNASSIGNED.

4. Enable the remote archived redo log destinations

On the source database, perform the following steps to enable the remote archived redo log destinations:

- a) **Begin to transmit redo data.**

On the source database, enable the `LOG_ARCHIVE_DEST_STATE_2` parameter:

```
ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2='ENABLE'
SCOPE=BOTH;
```

- b) **Switch log files:**

```
ALTER SYSTEM SWITCH LOGFILE;
```

Maximum Availability Architecture

- c) **Query the V\$ARCHIVE_DEST_STATUS view** to validate that there are no errors and to verify that the archive log destination for LOG_ARCHIVE_DEST_2 is active:

```
SELECT DEST_ID, DEST_NAME, DESTINATION, DATABASE_NAME,
       SRL, ERROR
FROM V$ARCHIVE_DEST_STATUS;
```

5. Replicate schemas on the downstream database.

Run the DBMS_STREAMS_ADM.MAINTAIN_SCHEMAS PL/SQL procedure on the downstream database to automate many of the following steps to set up Streams:

- Instantiation of all objects
- Enabling supplemental logging on the source database
- Configuring the Streams capture and LogMiner processes
- Configuring the Streams buffered queues
- Configuring the Streams apply Process
- Performing the dump and load of the database dictionary needed by LogMiner

The following DBMS_STREAMS_ADM.MAINTAIN_SCHEMAS () PL/SQL procedure example configures and starts Streams and replicates six different database schemas (TEST1, TEST2, ... TEST6) as a one-way replication from the source database. The example uses all of the configuration names and recommendations (Oracle Directory objects, database links, and so on) described in the [“Guidelines for Preparing All Streams Configurations”](#) section of this white paper.

Note: If you use the DBMS_STREAMS_ADM.MAINTAIN_SCHEMAS PL/SQL procedure example in your Streams environment, you must modify the parameters to match the locations and parameter values in your Streams configuration. Running this procedure should have minimal impact on the source database while creating the initial database dictionary dump, moving any table data to the destination database, and enabling supplemental logging.

On the downstream database, run the following DBMS_STREAMS_ADM.MAINTAIN_SCHEMAS () PL/SQL procedure:

```
BEGIN
    DBMS_STREAMS_ADM.MAINTAIN_SCHEMAS (
        SCHEMA_NAMES =>
        'TEST1 , TEST2 , TEST3 , TEST4 , TEST5 , TEST6 ' ,
```

Maximum Availability Architecture

```
        SOURCE_DATABASE           =>
'STREAMSS.US.ORACLE.COM' ,
        SOURCE_DIRECTORY_OBJECT   => 'STREAMS_DIR' ,
        DESTINATION_DATABASE      =>
'STREAMSD.US.ORACLE.COM' ,
        DESTINATION_DIRECTORY_OBJECT => 'STREAMS_DIR' ,
        CAPTURE_QUEUE_NAME        => 'DS_STREAMS_QUEUE' ,
        APPLY_QUEUE_NAME          => 'DS_STREAMS_QUEUE' ,
        BI_DIRECTIONAL            => FALSE ,
        INCLUDE_DDL               => FALSE
    );
END;
/
```

When using the `DBMS_STREAMS_ADM.MAINTAIN_SCHEMAS` procedure to configure Streams for downstream capture, you should:

- Use the value of the **GLOBAL_NAME** parameter of the database that was created in the “[Guidelines for Preparing All Streams Configurations](#)” section with the **SOURCE_DATABASE** and **DESTINATION_DATABASE** parameters. Ensure that the name of the database links match the value specified by the **GLOBAL_NAME** parameter for each database. Streams requires that you use the **GLOBAL_NAME** parameter so the capture process can function correctly.
- Use the directory objects created on each database with the **SOURCE_DIRECTORY_OBJECT** and **DESTINATION_DIRECTORY_OBJECT** parameters. In the example configuration, the directory objects happen to have the same name. These parameters specify the location where Oracle Data Pump stores the export/import files and log files. You should ensure that there is enough disk space available in the location for the dump files.
- Set the **CAPTURE_QUEUE_NAME** and **APPLY_QUEUE_NAME** parameters to the same name so that both the capture and apply processes share the same queue. This provides the best efficiency for Streams where both capture and apply are configured on the same database. Also, in the example, Streams propagation is not set up because the capture and apply processes are set up to use the same Streams queue.
- Set the **BI_DIRECTIONAL** parameter to `FALSE` for one-way replication from the source database schemas to the destination database schemas.
- Set the **INCLUDE_DDL** parameter to `FALSE`, as shown in the example, if you do not want to capture or apply any DDL changes.

When running the `DBMS_STREAMS_ADM.MAINTAIN_SCHEMAS ()` procedure, note that the procedure may take some time to complete. Once the

Maximum Availability Architecture

procedure starts to execute, you can monitor the procedure's actions using the `DBA_RECOVERABLE_SCRIPT` view or by viewing the `ALERT.LOG` files of both databases. The `ALERT.LOG` on the downstream database shows when LogMiner begins mining the standby redo log files.

You can also monitor the current progress of the Streams build process by querying the `STREAMS_BUILD_STATUS` view on the downstream capture database. Query the `STREAMS_BUILD_STATUS` view while the `DBMS_STREAMS_ADM.MAINTAIN_SCHEMAS()` procedure is running. See [Appendix B: “Monitoring Streams Configuration Progress”](#) in this white paper for an example.

6. Configure real-time mining for the capture process.

On the same (downstream) database where you ran the `DBMS_STREAMS_ADM.MAINTAIN_SCHEMAS` PL/SQL procedure in Step 5, run the `DBMS_CAPTURE_ADM.SET_PARAMETER` procedure to configure the Streams capture process to perform real-time mining of the redo log that is shipped from the source database. (**Note:** To obtain the name of the capture process, query the `CAPTURE_NAME` column of the `DBA_CAPTURE` view.)

```
BEGIN
  DBMS_CAPTURE_ADM.SET_PARAMETER(
    capture_name => 'DS_REALTIME_CAPTURE',
    parameter    => 'downstream_real_time_mine',
    value        => 'y');
END;
/
```

Once the parameter has been set for the capture process, you can monitor the `ALERT.LOG` on the downstream database to watch LogMiner mining the standby redo log files.

7. Verify the downstream database.

On the downstream database, query the `V$STANDBY_LOG` view and notice that a sequence number is being assigned to one of the log groups and the log may be marked “ACTIVE.” To see this change:

- a. On the source database, issue the `ALTER SYSTEM SWITCH LOGFILE` statement
- b. On the downstream database, query the `V$STANDBY_LOG` view.

For example:

```
SELECT GROUP#,SEQUENCE#, FIRST_TIME, FIRST_CHANGE#,
LAST_CHANGE#, ARCHIVED,STATUS FROM V$STANDBY_LOG;
```

Maximum Availability Architecture

If the sequence numbers are being assigned to the standby log groups and the `FIRST_CHANGE#` and `LAST_CHANGE#` columns are being updated for the assigned standby redo log, then the real-time mining for downstream capture has been properly configured. If you do not see the `SEQUENCE#` from the source database being assigned to the standby log group, do not proceed until this is corrected. Query the `V$ARCHIVE_DEST_STATUS` view on the source database to ensure that the remote database destination is `VALID` and that the destination does not have any errors. Also, ensure that the `LOG_ARCHIVE_CONFIG` parameter is set on both databases. See [Table 2](#) above for an example parameter file.

In a downstream configuration, the source database and downstream capture databases should have the characteristics described in the following table:

Source Database	Downstream Capture Database
1. A Streams administration user	1. A tablespace dedicated to Streams
2. Supplemental logging enabled for application schema objects	2. A Streams administration user
3. A database link pointing to the downstream capture database	3. A database link pointing back to the source database
4. Redo shipping configured to transmit redo data to the downstream capture database	4. Standby redo logs configured to receive redo data from the source database
	5. A Streams buffered queue
	6. A capture and an apply process

Recommendations for Upstream (Local) Capture Configurations

This section provides configuration recommendations for local capture configurations.

Note: For downstream configuration recommendations, see the [Recommendations for Downstream Capture](#) section.

Upstream capture, which is also referred to as **local capture**, occurs when the Streams capture process captures changes on the source database. Then, a propagation process dequeues the LCRs from the local buffered queue and propagates them to a destination database where they are enqueued into another buffered queue where typically an apply process is configured.

Perform the following steps to prepare the source and target databases for local capture:

1. **Ensure the configuration meets all of the requirements in the [“Guidelines for Preparing All Streams Configurations”](#) section.**

Maximum Availability Architecture

2. Specify initialization parameters on the source and target databases.

[Table 3](#) shows the parameter settings for the example configuration. See the *Oracle Database Reference 10g Release 2 (10.2)* [9] for complete information about each initialization parameter. Note that the names in [Table 3](#) are for the purpose of example only; be sure that the parameters you specify match the names of objects in your Streams configuration.

Table 3: Recommended Initialization Parameter Settings in a Local Capture Configuration

Source Database	Target Database
<pre>LOG_ARCHIVE_DEST_1= 'LOCATION=+ENG/streamsrc10g/archivelog/ VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) MANDATORY' LOG_ARCHIVE_DEST_STATE_1=ENABLE LOG_ARCHIVE_MAX_PROCESSES=4⁷</pre>	<pre>LOG_ARCHIVE_DEST_1= 'LOCATION=+CX700/streamsdestb10g/ VALID_FOR=(ONLINE_LOGFILE,PRIMARY_ROLE)' LOG_ARCHIVE_DEST_STATE_1=ENABLE</pre>

3. Replicate schemas

To configure Streams, run the `DBMS_STREAMS_ADM.MAINTAIN_SCHEMAS` PL/SQL procedure on the database where the capture process runs, which in this case is the source database. The `DBMS_STREAMS_ADM` PL/SQL package automates many of the steps needed to set up the Streams configuration, including:

- Instantiation of all objects
- Enabling supplemental logging on the source database
- Configuring the Streams Capture and LogMiner processes on the source database
- Configuring the Streams buffered queues on both source and target databases
- Configuring propagation between the source and target databases
- Configuring the Streams apply process on the target database
- Performing the dump and load of the database dictionary needed by LogMiner

The following `DBMS_STREAMS_ADM.MAINTAIN_SCHEMAS()` PL/SQL procedure example configures six different database schemas (`TEST1`, `TEST2`, ... `TEST6`) as a one-way replication. The example uses all of the configuration

⁷ Specify a minimum of four background processes given that there may be two or more destinations being configured on the database. By setting this parameter you improve archiving performance and reduce the need to wait for archiving to either an online redo log or to remote destinations.

Maximum Availability Architecture

names and recommendations (Oracle Directory objects, database links, and so on) described in the “[Guidelines for Preparing All Streams Configurations](#)” section of this white paper.

Note: If you use this procedure example to configure your Streams environment, be sure to edit the parameters to match the names of objects in your Streams configuration.

```
BEGIN
DBMS_STREAMS_ADM.MAINTAIN_SCHEMAS(
  SCHEMA_NAMES    => 'TEST1,TEST2,TEST3,TEST4,TEST5,TEST6',
  SOURCE_DATABASE => 'STREAMSS.US.ORACLE.COM',
  SOURCE_DIRECTORY_OBJECT    => 'STREAMS_DIR',
  DESTINATION_DATABASE    => 'STREAMSD.US.ORACLE.COM',
  DESTINATION_DIRECTORY_OBJECT => 'STREAMS_DIR',
  BI_DIRECTIONAL          => FALSE,
  INCLUDE_DDL             => FALSE
);
END;
/
```

When configuring the `DBMS_STREAMS_ADM.MAINTAIN_SCHEMAS` procedure for Streams local capture:

- Use the **GLOBAL_NAME** that was created in the “[Guidelines for Preparing All Streams Configurations](#)” section with the **SOURCE_DATABASE** and **DESTINATION_DATABASE** parameters. Ensure that the name of the database links match that of the **GLOBAL_NAME** parameter specified for each database. Streams requires the **GLOBAL_NAME** parameter for the capture process to function correctly.
- Use the directory objects created on each database with the **SOURCE_DIRECTORY_OBJECT** and **DESTINATION_DIRECTORY_OBJECT** parameters. In the example configuration above, the directory objects happen to have the same name.
- For local capture, do not specify the **CAPTURE_QUEUE_NAME** and **APPLY_QUEUE_NAME** parameters.
- Set the **BI_DIRECTIONAL** parameter to `FALSE` for one-way replication from the source database schemas to the destination database schemas.
- Set the **INCLUDE_DDL** parameter to `FALSE` if you do not want to capture or apply any DDL changes.

Monitor the progress by reviewing the `ALERT.LOG` file on both databases. The source database’s `ALERT.LOG` will show LogMiner mining the local archive and the online redo log files. Once the procedure starts to execute, you can monitor the procedure’s actions using the `DBA_RECOVERABLE_SCRIPT` view or by viewing

Maximum Availability Architecture

the ALERT.LOG files of both databases. The ALERT.LOG on the downstream database shows LogMiner beginning to mine the standby redo log files. You can also monitor the progress of the Streams build process by creating and querying the STREAMS_BUILD_STATUS view as described in [Appendix B “Monitoring the Streams Configuration Progress.”](#)

In a one-way configuration, the source database and target databases should have the characteristics described in the following table:

Source Database	Target Database
1. A tablespace dedicated to Streams	1. A tablespace dedicated to Streams
2. A Streams administration user	2. A Streams administration user
3. Supplemental logging enabled for application schema objects	3. A database link pointing to the source database (optional)
4. A database link pointing to the target database	4. A Streams buffered queue
5. A Streams buffered queue for staging and propagating LCRs	5. An apply process for dequeuing and applying LCRs
6. A capture process	
7. A Streams propagation process and its associated database job	

TASKS TO PERFORM AFTER CONFIGURING STREAMS

After setting up your downstream or local capture configuration, perform the following steps to finalize and verify the configuration:

1. Set Capture Checkpoint Retention Time Parameter

Set the CHECKPOINT_RETENTION_TIME capture parameter to specify the number of days of checkpoints the capture process will retain. The default value for this parameter is 60 days but the recommended initial setting is 7 days.

Changing this parameter to 7 days can reduce the volume of checkpoint information that the capture process purges, and therefore improves the overall performance of the capture process.

For example:

```
BEGIN
    DBMS_CAPTURE_ADM.ALTER_CAPTURE (
```

Maximum Availability Architecture

```
capture_name => 'DS_REALTIME_CAPTURE',  
checkpoint_retention_time => 7 );  
END;
```

Query the `CAPTURE_NAME` of the `DBA_CAPTURE` view to obtain the name of the capture process.

2. Set apply parallelism to 4.

As a starting point, set the degree of parallelism for the apply process to “4” as shown in the following example:

```
BEGIN  
    DBMS_APPLY_ADM.SET_PARAMETER('APPLY$_STREAMSS_36',  
    'PARALLELISM', '4');  
END;  
/
```

See the *Oracle Streams Performance and Troubleshooting Best Practices* [11] white paper for a discussion about how to determine the optimum degree of parallelism.

Note: The example used in this step specifies the name `APPLY$_STREAMSS_36` for the apply process. To find the name of the apply process in your configuration, query the `APPLY_NAME` column of the `DBA_APPLY` view on the target database.

3. Run the Streams Health Check script.

The Streams Health Check script provides information about the current configuration of Streams, and must be run on all databases where Streams is configured. The Health Check script and instructions are available from [OracleMetalink](#) Note 273674.1. If the Health Check script returns errors, then see the “[Troubleshooting Streams Configurations](#)” section in this white paper for help with tuning and configuring the network.

4. Verify the Streams Configuration for use over a network

See [Appendix A](#) for recommendations and guidelines to help you ensure your Streams configuration is properly configured for use over a network.

TROUBLESHOOTING STREAMS CONFIGURATIONS

This section provides the following troubleshooting topics:

- [Repairing the initial configuration](#)
- [Repairing common runtime errors](#)

Repairing the Initial Configuration

If either an abnormal termination occurs or a procedure fails, the problem typically occurs because the `DBMS_STREAMS_ADM.MAINTAIN_#####` scripts or other Streams PL/SQL packages are executed in a configuration in which some component is not configured properly. Oracle Streams maintains its operational and configuration steps as it performs each task for configuring all components of Streams. These set of tasks are bundled into a scripts maintained in the Streams metadata tables.

To recover from an abnormal termination or failure when running the `DBMS_STREAMS_ADM.MAINTAIN_SCHEMAS()` package, perform the following steps while logged into SQL*Plus as the `STREAMSADMIN` user:

1. Identify the `script_id` that encountered the error.

Query the view `DBA_RECOVERABLE_SCRIPT_ERRORS` to determine the script ID for the recent error. For example:

```
SELECT TO_CHAR(ERROR_CREATION_TIME, 'DD-MON-YYYY
hh24:mi:ss'),
       SCRIPT_ID, BLOCK_NUM,
       ERROR_NUMBER, ERROR_MESSAGE
FROM DBA_RECOVERABLE_SCRIPT_ERRORS;
```

Record the script ID number and block number for use in the next several steps. The script ID will be a very long number. Examine the error number and error message.

2. Examine the script block.

Assuming the numbers that you recorded for the `SCRIPT_ID` and `BLOCK_NUM` columns from step 1 are `437A1B88F5944DFCE040238238A6273D` and `11` respectively, then issue the following query to identify the specific block in the script:

```
SET LONG 1024
SELECT STATUS, FORWARD_BLOCK_DBLINK, FORWARD_BLOCK
FROM DBA_RECOVERABLE_SCRIPT_BLOCKS
WHERE SCRIPT_ID =
      '437A1B88F5944DFCE040238238A6273D'
AND BLOCK_NUM = 11;
```

Note: You must issue the `SET LONG` command before issuing the query, because the `FORWARD_BLOCK` column is a CLOB data type.

3. Diagnose the problem.

- a. Assess the error message returned in conjunction with the PL/SQL code (displayed in step 2) and determine whether the error is recoverable or nonrecoverable:

A **recoverable** error can be corrected. For example, errors such as supplying an invalid value to a database system initialization parameter or if `ARCHIVELOG MODE` is not enabled are recoverable errors.

A **nonrecoverable** error is typically caused when parameters supplied to the `DBMS_STREAMS_ADM.MAINTAIN_nnn` procedures are too long or are incorrect.

4. Recover from the error you encountered in step 3.

- If the error is recoverable, then correct the error and complete the Streams configuration by executing the following procedure:

```
BEGIN
  DBMS_STREAMS_ADM.RECOVER_OPERATION(
    script_id =>
      '437A1B88F5944DFCE040238238A6273D' ,
    operation_mode => 'FORWARD');
END;
/
```

If the procedure finishes with no further errors, then the initial configuration has been completed.

- If the error is nonrecoverable, then:
 - a. Run the following procedure to roll back the incomplete Streams configuration:

```
BEGIN
  DBMS_STREAMS_ADM.RECOVER_OPERATION(
    script_id =>
      '437A1B88F5944DFCE040238238A6273D' ,
    operation_mode => 'ROLLBACK');
END;
/
```

Maximum Availability Architecture

- b. Correct the error (review and correct any parameters passed to the `DBMS_STREAMS_ADM.MAINTAIN_nnn` procedures).
- c. Rerun the `DBMS_STREAMS_ADM.MAINTAIN_SCHEMAS ()` procedure that you ran in either the downstream capture or local capture configurations.

If the procedure finishes without errors, then the configuration is complete.

5. **If errors are returned in step 4** after running either the procedure to perform a FORWARD or the procedure to perform a ROLLBACK action, then perform the following steps to purge the Streams scripts from recoverable views. Note that while the following steps do not remove any Streams components from the database, the steps do remove the scripts that are used to build the Streams environment.

- a. **Perform a recover operation to purge the specific Streams configuration:**

```
BEGIN
    DBMS_STREAMS_ADM.RECOVER_OPERATION(
        SCRIPT_ID =>
        '437A1B88F5944DFCE040238238A6273D' ,
        OPERATION_MODE => 'PURGE' );
END;
```

- b. **Once the procedures have completed, correct the errors** encountered in “Step 3 Diagnose the problem” before rerunning the `DBMS_STREAMS_ADM.MAINTAIN_SCHEMAS` (or the other `DBMS_STREAMS_ADM.MAINTAIN_nnnn` procedures):

CAUTION: In rare situations where performing the preceding steps do not repair the initial Streams configuration, you may need to run the `DBMS_STREAMS_ADM.REMOVE_STREAMS_CONFIGURATION` procedure to remove the Streams components from the database where you are attempting to configure Streams. Run this procedure only when you are sure you want to remove the entire Streams configuration from the local database.

Repairing Common Runtime Errors

To determine if there are common runtime errors, run the Health Check script on all databases where Streams has been configured with the capture, propagation, or apply process.

Maximum Availability Architecture

You can download the Health Check script from [OracleMetalink](#) Note 273674.1. The Health Check script provides useful scripts that quickly retrieve all of the current status information related to the Streams configuration in the Oracle database. The output from the Health Check script is in HTML format and is organized with links to the key areas of interest. Use these links to access and review the specific areas for any errors.

The following sections describe how to investigate errors for the capture, propagation, and apply processes:

- [Investigate Errors for the Capture Process](#)
- [Investigate Errors for the Propagation Process](#)
- [Investigate Errors for the Apply Process](#)

Investigate Errors for the Capture Process

At the top of the Health Check report, find the line that starts with `Configuration` and click **Capture**.

In the report under the `++CAPTURE PROCESSES IN DATABASE++` heading, look for the Status, Error Number, and Error Message columns. The following screenshot shows this portion of the Health Check report:

- **If there is an error, review the error number and message provided.**

You can also issue the following query to list all of the capture processes with their current status and the last time the status was changed:

```
SELECT CAPTURE_NAME, STATUS,
       TO_CHAR(STATUS_CHANGE_TIME, 'DD-MON-YYYY HH24:MI'),
       ERROR_NUMBER, ERROR_MESSAGE
FROM DBA_CAPTURE
WHERE ERROR_NUMBER IS NOT NULL;
```

The query returns the error number and error message that provide more diagnostic information.

- **If the capture process was unable to access an archived redo log file.**

Errors can occur if an archived redo log file was accidentally removed due to lack of space or the file could have been deleted manually. You may need to restore the missing archived redo log file (as reported by the Health Check report).

To determine which archived redo log file is needed to restart the capture process, click **Database** at the top of the report and then scroll up until you see following line:

```
++ Minimum Archive Log Necessary to Restart Capture ++
```

Maximum Availability Architecture

The start SCN and the name of the archived redo log files are listed under this heading.

For example, the following example shows the start SCN and the name of the archived redo log file under the heading:

```
STREAMS Health Check (V3.0.20) for STREAMSB.US.ORACLE.COM on Instance=STRM10g8
●
●
●
++ Minimum Archive Log Necessary to Restart Capture ++
Note: This query is valid for databases where the capture processes exist for the same source database
Capture will restart from SCN 44250857 in the following file:
+CX700/streamsdestb10g/arch_srl_streamsa/1_283_650909295.dbf (16:30:54 07/16/08)
```

You can query V\$ARCHIVED_LOG view (on the database where capture is configured) to determine where the control file indicates the corresponding archived redo log file resides.

If online or archived redo log files are missing during normal Streams operations, the capture process pauses and waits for the required redo data. The STATE column of the V\$STREAMS_CAPTURE view shows that the state of the capture is "WAITING FOR REDO, LAST SCN MINED <SCN number>". Use the SCN in the V\$ARCHIVED_LOG view to determine which archive redo log file is missing for local capture.

This state can also occur for downstream capture where the LNS process (of the redo transport service) on the source database falls behind. The downstream capture database configured with real-time mine capture mines all redo it has received in the standby redo logs and waits for more redo to be delivered. In this case, the SCN is the last SCN seen by the capture process. Ensure that redo data is being shipped to the down stream capture database. See the [Recommendations for Configuring Downstream Capture](#) section.

- **Restart the capture process.**

Once you have addressed the error, restart the capture process by running the DBMS_CAPTURE_ADM.START_CAPTURE PL/SQL procedure. Execute the following query to determine if the capture process has the "ENABLED" status:

```
SELECT capture_name, status,
       TO_CHAR(status_change_time, 'DD-MON-YYYY HH24:MI'),
       error_number, error_message
FROM DBA_CAPTURE
```

Investigate Errors for the Propagation Process

At the top of the Health Check report, click **Propagation** if this is a local capture configuration. A table displays showing the propagation name, the source and destination queue names, and the database link used to reach the remote database.

1. **Examine the `Error Date` and `Error Message` columns to determine if there are any errors.**

If there are errors in the `Error Message` column, the probable causes may include the following:

- Remote database instance is down or unreachable.
- Remote TNS listener is down.
- The local `TNSNAMES.ORA` file containing the TNS descriptor used by the database link might not be accessible or it was incorrectly modified.
- A possible network outage.

Review the section about configuring TNS and networking best practices in the “[Recommendations for Streams Configurations Used Over a Network](#)” section of this white paper.

2. **Query the `DBA_PROPAGATION` view to determine the list of all defined propagations, the current status of the propagations along with any errors, and the date and time when the errors occurred.**

For example:

```
SELECT PROPAGATION_NAME, SOURCE_QUEUE_NAME,
       DESTINATION_QUEUE_NAME, DESTINATION_DBLINK, STATUS,
       ERROR_MESSAGE, TO_CHAR(ERROR_DATE, 'DD-MON-YYYY
                               HH24:MI ')
       FROM DBA_PROPAGATION;
```

As the 'streamsadmin' or as a user that owns the database link, attempt to connect to the remote database using the database link from the `DESTINATION_DBLINK` column that was returned by the above query. Successfully connecting to the remote database validates that the database link is working.

See Also: The “Monitoring Replication” section of the [Oracle Streams Replication Administrator's Guide](#) [13] for information about monitoring the apply process and displaying detailed information about error transactions to see how the LCRs are shown in the error queue.

3. **Query the `DBA_QUEUE_SCHEDULES` view to determine if the queue propagation is disabled.**

It is possible that no errors are returned by the `DBA_PROPAGATION` view until the number of failed propagation attempts has reached 16. In this case,

Maximum Availability Architecture

the error displays when you query the `DBA_QUEUE_SCHEDULES` view, as follows:

```
SELECT QNAME, DESTINATION,
       SCHEDULE_DISABLED,
       FAILURES,
       LAST_ERROR_DATE,
       LAST_ERROR_TIME,
       LAST_ERROR_MSG
FROM DBA_QUEUE_SCHEDULES;
```

If the `FAILURES` column contains the number 16, then this queue propagation schedule is disabled (`SCHEDULE_DISABLED` should contain 'Y'). Examine the `LAST_ERROR_MSG` column, because there may be more than one error message. Diagnose and correct the errors. Then, restart the propagation.

For example:

```
BEGIN
DBMS_PROPAGATION_ADM.STOP_PROPAGATION('PROPAGATION$_11');
DBMS_PROPAGATION_ADM.START_PROPAGATION('PROPAGATION$_11');
END;
```

Query the `PROPAGATION_NAME` from the `DBA_PROPAGATION` view. Query the `DBA_QUEUE_SCHEDULES` column again to see if the propagation is working properly.

You can also use the Streams Health Check report to check for any errors in the propagation queue schedule. Click on **Propagation** on the "Statistics Quick Link" line at the top of the report.

Investigate Errors for the Apply Process

A variety of run-time errors can occur in the course of applying transactions on the target database. The Health Check utility collects diagnostic information for the apply process from the `DBA_APPLY` view and displays the information in the `ERROR_NUMBER` and `ERROR_MESSAGE` columns of the Health Check report. The following steps describe how to investigate any reported errors:

1. At the top of the Health Check report for the target database, find the Configuration line and click the **Apply** link to see the apply configuration information. The following example shows the top portion of the report:

STREAMS Health Check (V3.0.20) for STREAMSB.US.ORACLE.COM on Instance=STRM10g8

Configuration: Queue Capture Propagation Apply Database

Analysis: [History Rules Notifications Configuration Performance Wait Analysis](#)

Statistics: [Streams Statistics Queue Capture Propagation Apply Apply Errors](#)

Maximum Availability Architecture

2. In the apply process configuration information, examine the contents of the **Error Number** and **Apply Error Message** columns to determine if errors have occurred.
3. Query the DBA_APPLY_ERROR view to gather diagnostic information about specific transactions that encountered errors when they were applied.

Specifically, examine the MESSAGE_NUMBER column (specific to the LCR that encountered the error), and the ERROR_NUMBER and ERROR_MESSAGE columns.
4. Run the 'oerr' Oracle utility on the database server where the apply process is running to obtain further information about the error.

The following examples demonstrate how to investigate apply process errors.

Example 1

In the following example, assume that the Health Check report displays the ORA-26687 error:

```
ORA-26687: no instantiation SCN provided ...
```

This error indicates that the instantiation SCN of the objects to which the apply process is attempting to apply changes has not been set. Perform the following steps:

1. At the operating system command line on the server running the apply process, issue the oerr command to obtain more information:

```
% oerr ora 26687
```

The output from this command is similar to the following:

```
26687, 00000, "no instantiation SCN provided for \"\%s\".\"%\s\" in
source database \"\%s\"\"
// *Cause: Object SCN was not set. If the object is a table,
//       then both fields will be filled, for example "SCOTT"."EMP".
//       If the object is a schema, only one field will be set, for
//       example "SCOTT"."". And if the object is the entire
//       database, no fields will be set, for example "".".
// *Action: Set the SCN by calling DBMS_APPLY_ADM.SET_
%_INSTANTIATION_SCN
```

Maximum Availability Architecture

The error message text in the `ERROR_MESSAGE` column of the `DBA_APPLY_ERROR` view provides the object's name and schema depending on the level of replication (`TABLE`, `SCHEMA` or `GLOBAL`).

2. Run the `DBMS_APPLY_ADM.SET_XXXXX_INSTANTIATION_SCN` PL/SQL procedure to set the SCN, using `TABLE`, `SCHEMA` or `GLOBAL` in place of the `XXXXX` in the procedure name.
3. Query the `DBA_APPLY_INSTANTIATED_OBJECTS` view on the target database to list those objects that have instantiation SCNs.

See the “Troubleshooting Streams Replication” section of the [Oracle Streams Replication Administrator's Guide 10g Release 2](#) [13] and the “ORA-26687 Instantiation SCN Not Set” topic for more information about addressing this error.

To obtain details about a specific LCR, see the “[Displaying Detailed Information About Apply Errors](#)” section in [Oracle Streams Concepts and Administration 10g Release 2 \(10.2\)](#) [2]. This section also provides examples showing how to print LCRs for a particular transaction in the error queue. Use the `MESSAGE_NUMBER` from the `DBA_APPLY_ERROR` view to identify the specific LCR number.

Example 2

In the following example, assume that the Health Check report displays the ORA-26572 error:

```
ORA-26572: <schema>.<table_name>.<column_name>_UPDATE: argument
2 does not match replication catalog
```

This error is a configuration error example that is less common but may still occur when the tables on the source database use unique keys instead of a primary key. Streams identifies the key columns to use when a primary key is used on the source tables.

The ORA-26572 error displays if you query the `DBA_APPLY_ERROR` view on the target database. To address this issue, perform the following steps:

1. Use the `DBMS_METADATA.GET_DDL ()` PL/SQL procedure on the source database to obtain the DDL for the tables in question.
2. Use the `DBMS_APPLY_ADM.SET_KEY_COLUMNS ()` PL/SQL procedure to set the key columns that the apply process should use for primary keys with the columns that make up the unique key.
3. Use the `DBMS_APPLY_ADM.EXECUTE_ALL_ERRORS` PL/SQL procedure to apply the pending changes.
4. Query the `DBA_APPLY_ERROR` view to verify that the errors no longer exist.
5. Restart the apply process with `DBMS_APPLY_ADM.START_APPLY ()`.

Maximum Availability Architecture

You can configure the Streams apply process to run after an error is returned. To do this, set the apply parameter “DISABLE_ON_ERROR” to “N.” Whether or not the apply process continues depends on the type of application transactions. If the transactions being applied require that recent transactions be applied that may have failed, any future transactions will also fail.

For example, to set the DISABLE_ON_ERROR apply parameter, run the following procedure on the database where the apply process is running:

```
EXECUTE  
DBMS_APPLY_ADM.SET_PARAMETER('APPLY$_STREAMS_44','disable_on_e  
rror','N');
```

Do not set this parameter when you first configure Streams, or when the Streams environment is not working properly. Configure the parameter only when Streams is running without errors. Once you set the parameter to “N” you should monitor the DBA_APPLY_ERROR view to detect any errors or periodically run the Streams Health Check script.

CONCLUSION

Oracle Streams provides a powerful set of features that enable the enterprise to address a new wide set of business needs, from data sharing to business continuity techniques. Streams offers the flexibilities needed to address these needs. The best practices and guidance offered in this white paper provide a good starting point for configuring Oracle Streams in the enterprise. After implementing these guidelines, see the [Oracle Streams Performance and Troubleshooting Best Practices: Oracle Database 10g Release 2](#) [11] white paper for information about Streams monitoring and performance tuning.

APPENDIX A: USING STREAMS CONFIGURATIONS OVER A NETWORK

Proper configuration is crucial to Streams performance where propagation is done over any significant distances. This section provides best practices for configuring Streams for use over a network and includes the following topics:

- [Network Tuning Best Practices](#)
- [Standby Redo Log I/O Tuning for Downstream Databases](#)
- [Using Service Names in Oracle Net Alias Descriptors](#)

Network Tuning Best Practices

This section describes network tuning applicable to both Streams downstream and local capture configuration, and are most applicable over a Wide Area Network (WAN). The sample commands are based on a Linux configuration.

- Set TCP/IP parameters on all servers
- Set the Oracle Net `RECV_BUF_SIZE` and `SEND_BUF_SIZE` parameters equal to three times the *Bandwidth Delay Product* (BDP). Doing so provides the largest increase in network throughput.
- Set the Oracle Net Session Data Unit (SDU) size to 32767.
- Increase the default `SEND` and `RECEIVE` queue sizes associated with networking devices. For Linux systems, increase the `TXQUEUELENGTH` interface option via the `ifconfig` command on the sending side, and the `NET_DEV_MAX_BACKLOG` kernel parameter on the receiving side. As a proactive measure to prepare for future role transitions, change both of these parameters on the production database and on all standby databases.
- Ensure that the Oracle Net `TCP_NODELAY` parameter is set to `YES` (which is the default value).

Set TCP/IP Network Parameters On All Servers

The following examples are from a Linux system.

1. Set the network core write and receive memory buffer maximums:

```
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
```

2. Set the TCP/IP minimum, default, and maximum buffer sizes:

```
net.ipv4.tcp_rmem = 4096 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
```

Maximum Availability Architecture

On Linux systems, you can set these parameters without rebooting the system. Ensure that these parameters are set in the `/etc/sysctl.conf` file. It will also be necessary to run the system configuration command: `sysctl -p`. By default, running this command reads from the `/etc/sysctl.conf` file and sets the runtime network parameters. The parameter values that you specify will persist after you reboot the system.

Oracle Net Session Data Unit (SDU) Size

When sending data across the network, Oracle Net buffers data into session data unit (SDU) sized units. When large amounts of data are being transmitted or when the message size is consistent, increase the size of the SDU buffer to improve performance and network utilization. You can configure SDU size in an Oracle Net connect descriptor or globally in the `SQLNET.ORA` file. See the [Oracle Database Net Services Reference 10g Release 2 \(10.2\)](#) documentation [5] for more information.

You can override the current settings in the primary database `SQLNET.ORA` file. In a connect descriptor, specify the SDU parameter in a description list, as shown in the following example:

```
streamsdest10g_halinux06.us.oracle.com=
(DESCRIPTION=
  (SDU=32767)
  (ADDRESS=(PROTOCOL=tcp)
  (HOST=halinux06vip)
  (PORT=1521))
  (CONNECT_DATA=
  (SERVER = DEDICATED)
  (SERVICE_NAME = streamsd.us.oracle.com))
)
```

On the standby (or the Streams target) database, set SDU in the `SID_LIST` of the `LISTENER.ORA` file. For example:

```
SID_LIST_listener_name=
(SID_LIST=
(SID_DESC=
(SDU=32767)
(GLOBAL_DBNAME=streamsd.us.oracle.com)
(SID_NAME=STRM10g6)
(ORACLE_HOME=/usr/oracle)))
```

TCP Socket Buffer Sizes

TCP socket buffer settings will control how much network bandwidth can be used regardless of the bandwidth available in the network circuit. Socket buffer sizes need to be increased from their default values to improve utilization of available

Maximum Availability Architecture

bandwidth. When network latency is high, larger socket buffer sizes are needed to fully utilize network bandwidth.

The optimal socket buffer size is three times the size of the Bandwidth Delay Product (BDP). To compute the BDP, the bandwidth of the link and the network Round Trip Time (RTT) are required. RTT is the time required for a network communication to travel from the production database to the standby and back and is measured in milliseconds (ms). The following example assumes a gigabit network link with a RTT of 25 ms:

```
BDP= 1,000 Mbps * 25msec (.025 sec)
      1,000,000,000 * .025
      25,000,000 Megabits / 8 = 3,125,000 bytes
```

Given this example, the optimal SEND and RECEIVE socket buffer sizes are calculated as follows:

```
socket buffer size = 3 * bandwidth * delay
                   = 3,125,000 * 3
                   = 9,375,000 bytes
```

The size of the socket buffers can be set at the operating system level or at the Oracle Net level. Because socket buffer size requirements can become quite large (depending on network conditions), it is recommended to set them at the Oracle Net level (described in the [Oracle Database Net Services Reference 10g Release 2 \(10.2\)](#) documentation [5]) so that normal TCP sessions, such as telnet, do not use additional memory. Note that some operating systems have parameters that set the maximum size for all SEND and RECEIVE socket buffers. You must ensure that these values have been adjusted to allow Oracle Net to use a larger socket buffer size.

Configure the desired SEND and RECEIVE buffer sizes. For example:

```
RECV_BUF_SIZE=9375000
SEND_BUF_SIZE=9375000
```

The following example shows the SEND and RECEIVE socket buffer size being set as a DESCRIPTION attribute for a particular connect descriptor in the client-side SQLNET.ORA file:

```
streamsdest10g_halinux06.us.oracle.com =
  (DESCRIPTION=
    (SDU=32767)
    (SEND_BUF_SIZE=9375000)
    (RECV_BUF_SIZE=9375000)
```

Maximum Availability Architecture

```
(ADDRESS=(PROTOCOL=tcp)
(HOST=halinux06vip)(PORT=1521))
(CONNECT_DATA=
(SERVER = DEDICATED)
(SERVICE_NAME = streamsd.us.oracle.com)))
```

The socket buffer sizes must be configured at all sites. On a downstream capture database this can be accomplished in either the `SQLNET.ORA` file or the `LISTENER.ORA` file.

In the `LISTENER.ORA` file, you can specify the buffer space parameters either for a particular protocol address or for a description.

```
LISTENER=
(DESCRIPTION=
(ADDRESS=(PROTOCOL=tcp)
(HOST=halinux06vip)(PORT=1521)
(SEND_BUF_SIZE=9375000)
(RECV_BUF_SIZE=9375000)))
```

Network Device Queue Sizes

You can regulate the size of the queue between the kernel network subsystems and the driver for network interface card. You should size queues so that losses do not occur due to local buffer overflows. Therefore, careful tuning is required to ensure that the sizes of the queues are optimal for your network connection, particularly for high-bandwidth networks.

These settings are especially important for TCP, because losses on local queues cause TCP to fall into congestion control, which limits the TCP sending rates.

For Linux systems, there are two queues to consider:

- The interface transmit queue. The transmit queue size is configured with the network interface option `txqueuelen`.
- The network receive queue. The network receive queue size is configured with the kernel parameter `netdev_max_backlog`.

For example:

```
echo 20000 > /proc/sys/net/core/netdev_max_backlog
echo 1 > /proc/sys/net/ipv4/route/flush
ifconfig eth0 txqueuelen 10000
```

Using the default value of 100 for `txqueuelen` is inadequate for long-distance, high-throughput network links. For example, a gigabit network with a latency of 100ms would benefit from a `txqueuelen` of at least 10000.

Maximum Availability Architecture

[Table 4](#) shows the effects of tuning adjustments to TCP/IP. Consult with your operating system vendor for additional information about setting the queue sizes for various latencies.

Table 4: TCP/IP Tuning Effects

Testing Stage	Test duration (seconds)	Amount of data transferred	Network throughput achieved (Megabits/sec)	% change
Prior to tuning	60	77.2 MB	10.8 Mbps	N/A
After increasing network socket buffer size to 3*BDP from default of 16K	60	5.11 GB	731.0 Mbps	665% improvement over baseline prior to tuning
After above adjustment and increase of device queue lengths to 1,000 from default of 100	60	6.55 GB	937.0 Mbps	28% improvement

Standby Redo Log I/O Tuning for Downstream Database

As redo data is received by the downstream database, the redo is written to disk. In synchronous configurations, the disk write I/O must occur prior to sending acknowledgment back to the primary database that the redo has been received. Therefore it is important to optimize I/O on the standby database. To improve I/O performance on the standby database, employ the following best practices:

- Ensure that Oracle is able to use ASYNC I/O. Note that (by default) the Oracle database is configured for asynchronous I/O. However, you must also properly configure the operating system, HBA driver, and storage array.
- Maximize the write I/O size through all layers of the I/O stack. The layers will include one or more of the following: operating system (including ASYNC write size), device drivers, storage network transfer size, and disk array.
- Place standby redo logs in an ASM disk group that has at least the same number of disks as the ASM disk group where the primary online redo logs reside.
- Do not multiplex the standby redo logs.

Typically, RAID controllers configured in RAID5 perform write I/Os slower than when the controllers are configured with mirroring. If the process of writing to the

standby redo log becomes the bottleneck, consider changing your RAID configuration.

Use Service Names in Oracle Net Alias Descriptors

For Streams, use service names (as defined in the `SERVICE_NAMES` system parameter) in Oracle Net alias descriptors in the `TNSNAMES.ORA` file. Doing this allows Streams to inherit service migration when a specific Oracle RAC instance fails. For instance, the database link connection that the Queue-to-Queue Streams propagation uses is reestablished automatically. Use of static service Ids (SIDs) binds the descriptor to a specific instance and the node that instance runs on and will not benefit from automatic service migration.

APPENDIX B: MONITORING STREAMS CONFIGURATION PROGRESSION

You can monitor the current progress of the Streams build process by creating the `STREAMS_BUILD_STATUS` view on the downstream capture database and then querying the view. For example:

```
CREATE OR REPLACE VIEW STREAMS_BUILD_STATUS AS
    SELECT TO_CHAR(rs.creation_time, 'HH24:Mi:SS MM/DD/
    YY') CREATE_DATE, rs.status,
    rs.done_block_num||' of '||rs.total_blocks||'
    Steps Completed' PROGRESS,
    TO_CHAR(to_number(sysdate-
    rs.creation_time)*86400,9999.99)
    ELAPSED_SECONDS,
    substr(to_char(rsb.forward_block),1,100)
    CURRENT_STEP,
    rs.invoking_package||'.'||rs.invoking_procedure
    PROCEDURE, rs.script_id

FROM DBA_RECOVERABLE_SCRIPT rs,
     DBA_RECOVERABLE_SCRIPT_BLOCKS rsb

WHERE rs.script_id = rsb.script_id
      AND rsb.block_num = rs.done_block_num + 1;
```

Query the `STREAMS_BUILD_STATUS` view while the `DBMS_STREAMS_ADM.MAINTAIN_SCHEMAS()` package is running.

REFERENCES

1. Oracle Maximum Availability Architecture
<http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm>
2. *Oracle Streams Concepts and Administration 10g Release 2 (10.2)*
<http://otn.oracle.com/pls/db102/db102.toc?partno=b14229>
3. *Oracle Database 10g Release 2 Best Practices: Data Guard Redo Transport and Network Configuration* white paper
http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_DataGuardNetworkBestPractices.pdf
4. *Oracle Database PL/SQL Packages and Types Reference 10g Release 2 (10.2)*
<http://otn.oracle.com/pls/db111/db111.toc?partno=b28419>
5. *Oracle Database Net Services Reference 10g Release 2 (10.2)*
<http://otn.oracle.com/pls/db102/db102.toc?partno=b14213>
6. *Oracle Performance and Tuning Guide 10g Release 2 (10.2)*
<http://otn.oracle.com/pls/db102/db102.toc?partno=b14211>
7. *Oracle Database Upgrade Guide 10g Release 2 (10.2)*
<http://otn.oracle.com/pls/db102/db102.toc?partno=b14238>
8. *Oracle Database SQL Reference 10g Release 2 (10.2)*
<http://otn.oracle.com/pls/db102/db102.toc?partno=b14200>
9. *Oracle Database Reference 10g Release 2 (10.2)*
<http://otn.oracle.com/pls/db102/db102.toc?partno=b14237>
10. *Oracle Database High Availability Best Practices 10g Release 2 (10.2)*
<http://otn.oracle.com/pls/db102/db102.toc?partno=b25159>
11. *Oracle Streams Performance and Troubleshooting Best Practices: Oracle Database 10g Release 2* (available soon on the MAA Web site):
<http://www.oracle.com/technology/deploy/availability/htdocs/maa.htm>
12. *Oracle Enterprise Manager Licensing Information 10g Release 4 (10.2.0.4)*
<http://otn.oracle.com/pls/db102/db102.toc?partno=b40010>
13. *Oracle Streams Replication Administrator's Guide 10g Release 2 (10.2)*
<http://otn.oracle.com/pls/db102/db102.toc?partno=b14228>



Oracle Streams Configuration Best Practices: Oracle Database 10g Release 10.2
August 2008

Author: Darryl Presley and Pat McElroy

Contributing Authors: Andrew Babb, Viv Schupmann, Lawrence To, Doug Utzig

Oracle USA, Inc.
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

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

Copyright © 2008, Oracle. 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, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.