Oracle 9i Streams

Inderpal S. Johal
Principal Consultant
Data Softech, Inc.
July 24, 2003

Agenda

- Available High Availability Solution
- What is Oracle 9i Streams
- Architecture of Oracle Streams
- Common terminology
- Oracle Streams Application/Advantage
- Pre-requisite for Streams Implementation
- Comparison of High Availability Option
- Questions & Answers

Available High Availability/Failover Solutions

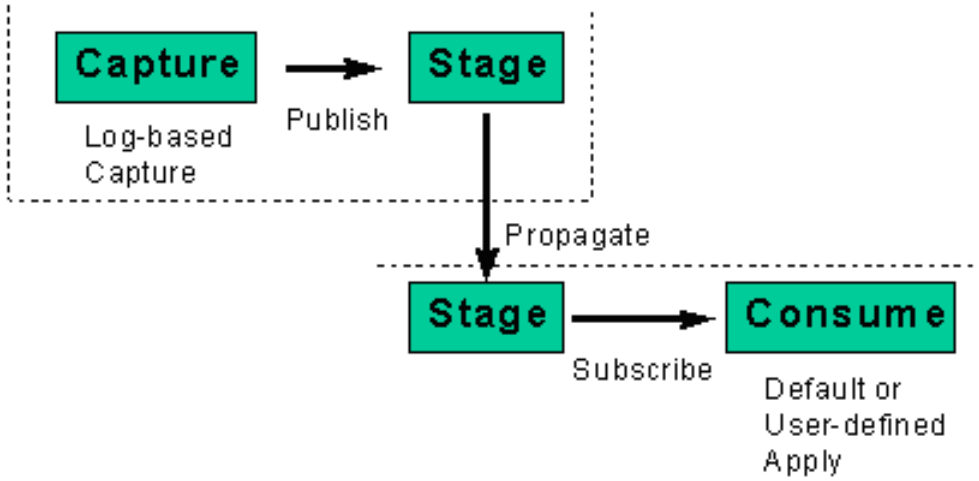
- Real Application Cluster
- Veritas / HP Clusters
- Oracle Advance Replication
- Oracle Standby Databases [Physical/ Logical]
- Oracle Failsafe
- Oracle 9i Streams

What is Oracle STREAMS

- New Oracle 9i feature to share information between Oracle as well as non-Oracle databases.

- Enables the propagation of Data and Events either within a database or from one database to another

Architecture of Oracle STREAMS



Streams Contains 3 basic elements that enables you to control

- What Information is put into the stream
  [Capture]
- How the Stream Flow from node to Node
  [Stage & Propagate]
- What Happens to events as they flow into each node
  [Optional Transformation]
- How the Stream terminates
  [Apply]

Common Terminology

RULES

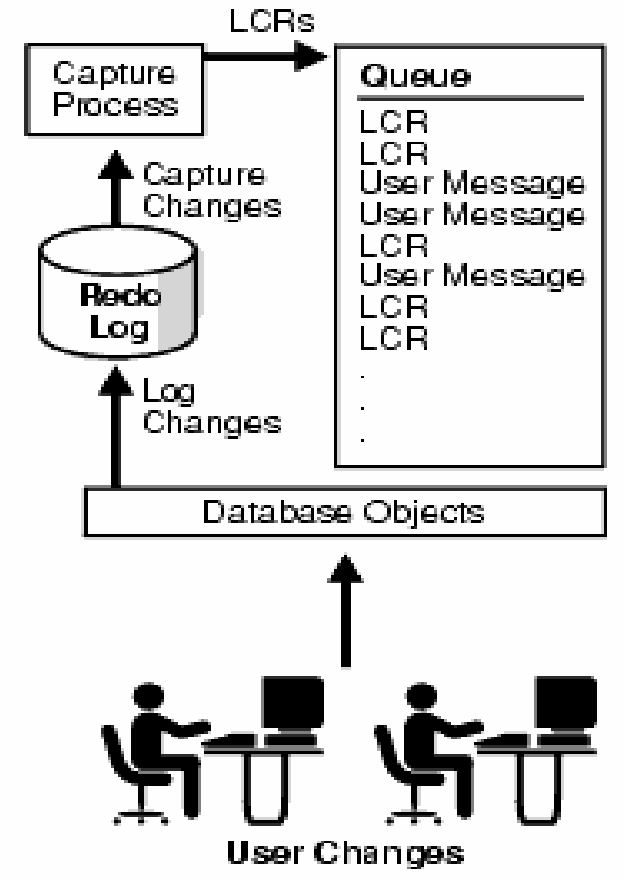
- A Rule is a database object that enables a client to perform an action when an event occurs and a condition is satisfied.
- Rules are evaluated by Oracle9i's built-in *rules engine* and evaluate to a Boolean result (*TRUE or FALSE)*
- Rules can be defined at different levels such as table level, schema level and database level.
- You can group related rules together into rule sets
- DBMS_RULE_ADM.CREATE_RULE_SET , DBMS_RULE_ADM.CREATE_RULE , DBMS_RULE_ADM.ADD_RULE

Common Terminology

LCR → Logical Change Record

- An LCR is an object with a specific format that describes a database change. LCRs are of two types : row LCRs and DDL LCRs.
- A row LCR describes a change to the data in a single row or a change to a single LOB column in a row as a result of DML
- A DDL LCR describes a data definition language (DDL) change
- Each LCR [DDL or DML] Contain the following main information
  - The name of the source database where the DDL/DML change occurred
  - The type of DDL/DML statement like Insert/Update/Alter table
  - The schema name of the user
  - The name of the database object
  - The SCN when the change was written to the redo log

CAPTURE PROCESS



- Reads the Redo logs
- Extracts the DDL/DML as per predefined set of **RULES** which define what changes to be captured
- Format the information into events also called **LCR**
- Place the information in the queue also called **Staging**

DSI DATA SOFTECH INC.
Complete Database Solution Provider

Capture Process - Log Miner

- A capture process captures changes from the redo log by using the infrastructure of Log Miner. Streams configures LogMiner automatically.
- By Default Logminer Tables are created to use SYSTEM tablespace and is not recommended
- Use the following to re-create all logminer tables in different tablespace before configuring stream setup

SQL> Execute DBMS_LOGMNR_D.SET_TABLESPACE('<TblSpNam>');

- If using OEM to configure Stream environment, then it will check and prompt your for different tablespace

Capture Process – Redo Log

- Seamless transition from reading an online redo log to reading an archived redo log and vice versa
- A capture process captures changes based on rules that you define.
  - A table rule captures either DML or DDL changes to a particular table.
  - A schema rule captures either DML or DDL changes to the database objects in a particular schema.
  - A global rule captures either all DML or all DDL changes in the database.

Capture Process - Creation

```
BEGIN
    DBMS_STREAMS_ADM.ADD_TABLE_RULES(
    table_name => 'hr.employees',
    streams_type => 'capture',
    streams_name => 'strm01_capture',
    queue_name => 'strm01_queue',
    include_dml => true,
    include_ddl => true,
    include_tagged_lcr => false);
END;
```

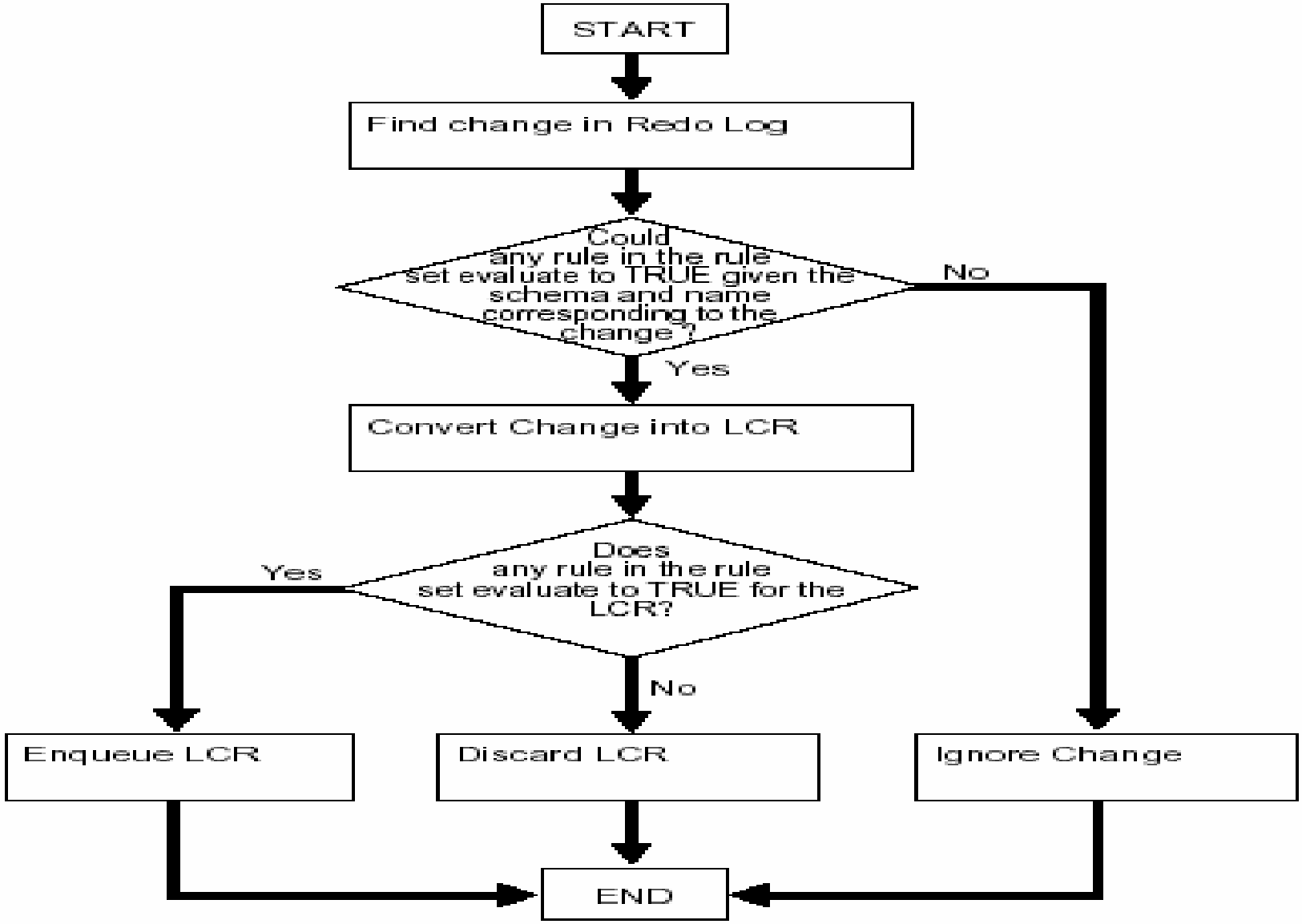
Capture Process - Components

- It depend on **PARALLELISM** setting. If parallelism is set to a value of 3 or greater,
  - Reader server for Reading Redologs
  - Preparer server for Formatting the information to LCR
  - Builder server for enqueue the LCR to the Queue
- If **parallelism** is set to 5, then a capture process uses one reader server, three **Preparer** servers, and one builder server.
- Note : **LOG_PARALLELISM** initialization parameter must be set to 1 for using **PARALLELISM** else get ORA-1374

Capture Process - Parallelism

```
BEGIN
    DBMS_CAPTURE_ADM.SET_PARAMETER(
    capture_name => 'strm01_capture',
    parameter => 'parallelism',
    value => '3');
END;
```

- The parallelism capture process parameter controls the number of Preparer servers used by a capture processes.
- If parallelism is set to 2 or lower, then a capture process itself [cp nn (01-99)] performs all the work without using any parallel execution servers.



Datatype Not Captured

LONG , LONG RAW, BFILE, ROWID, UROWID and User-defined types (object types, REFs, varrays, nested tables)

DML Not Captured

- CALL, EXPLAIN PLAN, or LOCK TABLE statements.
- Changes made to temporary tables, index-organized tables, or object tables.
- MERGE command is converted to INSERT or UPDATE

DDL Changes Not Captured

ALTER DATABASE, CREATE CONTROLFILE, CREATE DATABASE, CREATE PFILE, CREATE SPFILE

Capture Process Contd…

- A capture process never captures changes in the SYS and SYSTEM schemas
- A capture process does not capture DBMS_REDEFINITION package changes
- A capture process uses queue buffers available in shared pool area unlike queue tables on disk in AQ
- You can create, alter, start, stop, and drop a capture process

```
BEGIN
    DBMS_capture_ADM.Start_capture
    (Capture_name=>'capture_hr');
END
```

STAGING PROCESS

- It is a queue that provides a service to store and manage captured events.
- Message remain in staging area until consumed by all subscribers
- If the subscriber is another staging area, the event is propagated to the other staging area, either within the same database or in a remote database

STAGING PROCESS

- There are two types of events that can be staged in a Streams queue:
  - » logical change records (LCRs) and
  - » User messages.
- Your applications can enqueue/dequeue user messages using

  PL/SQL (DBMS_AQ package),JMS, OCI
- Staged events can be consumed or propagated, or both.

PROPAGATION

- Streams uses job queues to propagate events using job queue processes (J nnn)
- You can CREATE/DROP a propagation Using DBMS_STREAMS_ADM DBMS_PROPAGATION_ADM package
- The default schedule has the following properties:
  - The start time is SYSDATE().
  - The duration is NULL, which means infinite.
  - The next time is NULL
- You can alter the schedule for a propagation with ALTER_PROPAGATION_SCHEDULE procedure in the DBMS_AQADM package.

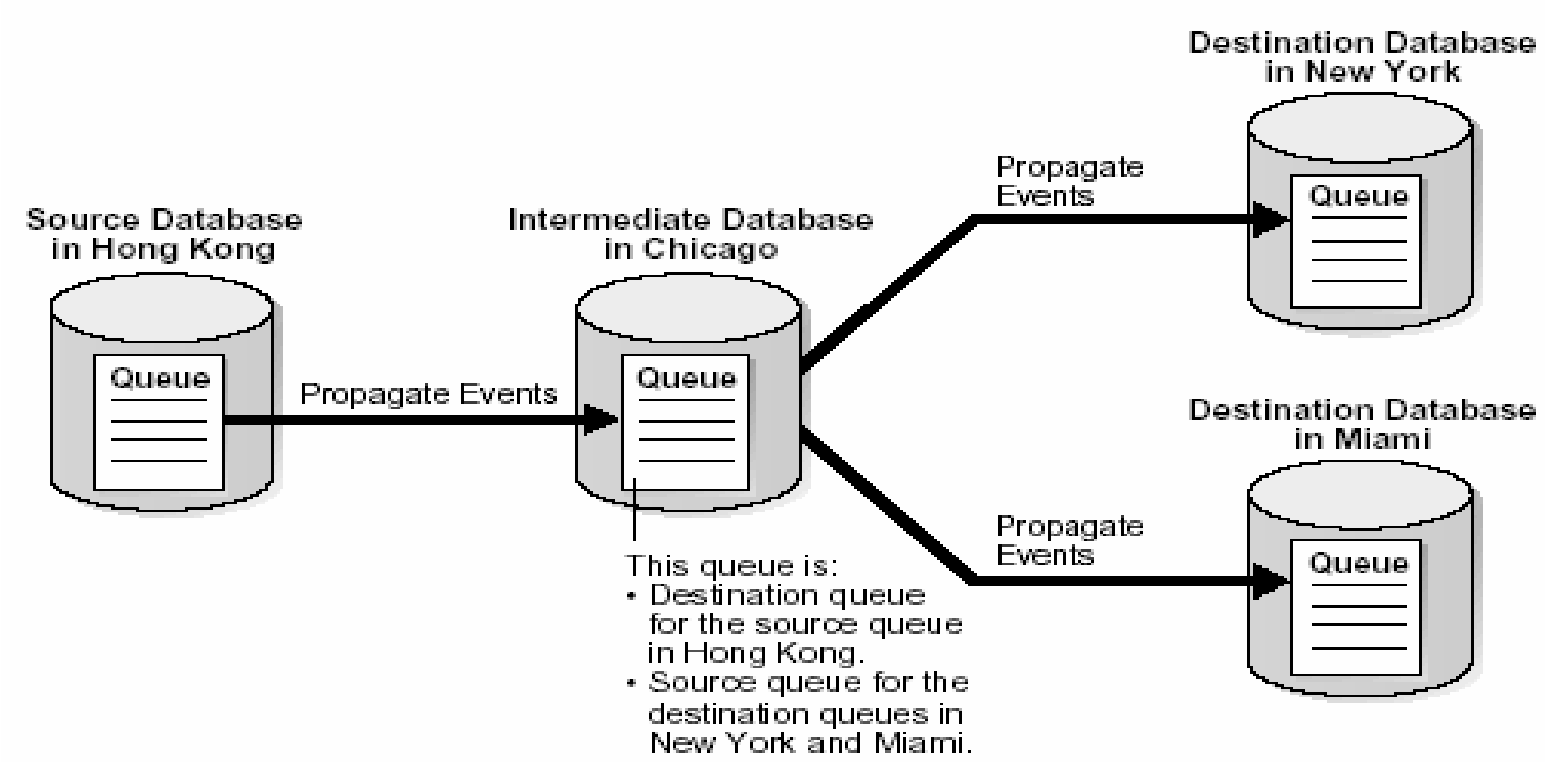
PROPAGATION - Creation

```
BEGIN
    Dbms_Streams_Adm.Add_Table_Propagation_Rules(
    table_name => 'hr.departments',
    streams_name => 'strm01_propagation',
    source_queue_name => 'strmadmin.strm01_queue',
    destination_queue_name =>
       'strmadmin.strm02_queue@dbs2.net',
    include_dml => true,
    include_ddl => true,
    include_tagged_lcr => false,
    source_database => 'dbs1.net' );
END;
```

PROPAGATION RULES

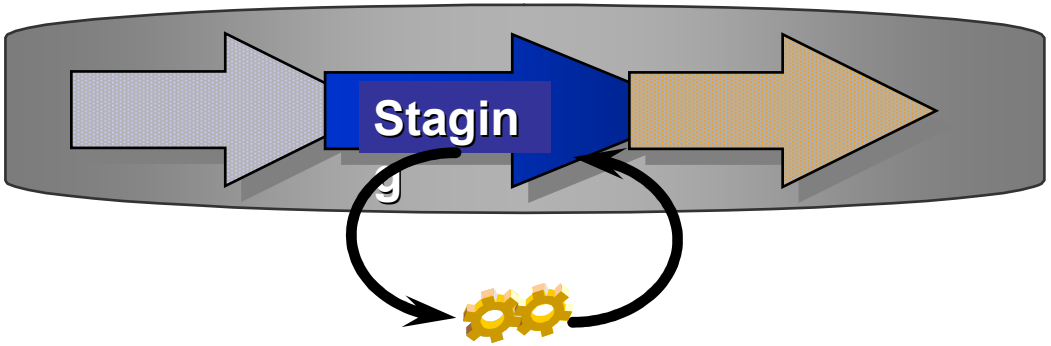
- You can specify propagation rules for LCR events at the following levels:
  - A table rule propagates either DML or DDL changes to a particular table.
  - A schema rule propagates either DML or DDL changes to the database objects in a particular schema.
  - A global rule propagates either all DML or all DDL changes in the source queue.
- For non-LCR events, you can create your own rules to control propagation.

Directed Network



- **Queue Forwarding and Apply Forwarding**

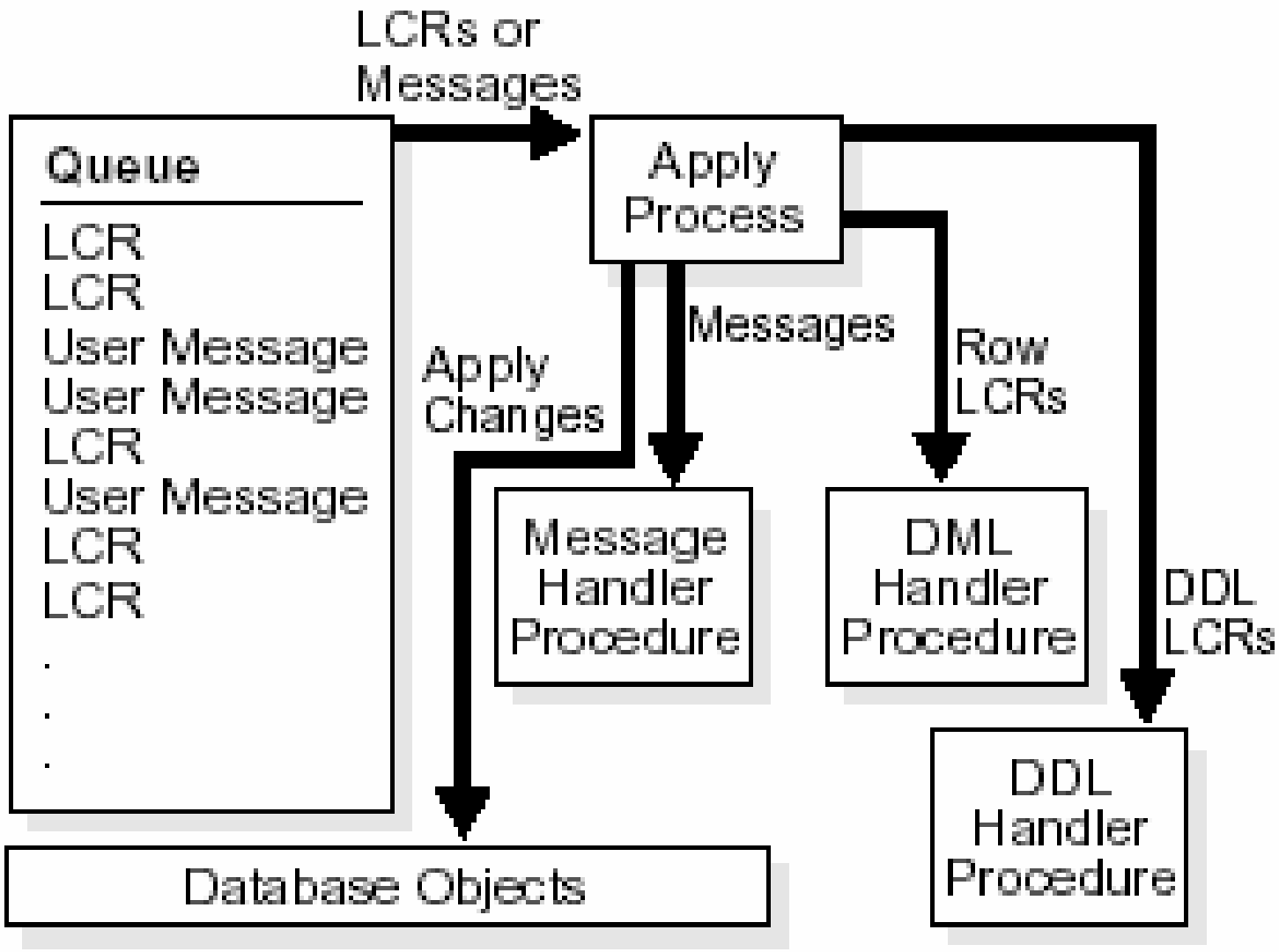
TRANSFORMATION



- Transformations can be performed
  - as events enter the staging area
  - as events leave the staging area
  - as events propagate between staging areas
- Transformation examples
  - change format, data type, column name, table name

APPLY PROCESS

- An apply process is an optional Oracle background process [ap nn] that dequeue logical change records (LCRs) and user messages
- For non-LCR messages, the apply servers pass the events to the message handler.
- Automatic conflict detection with optional resolution →unresolved conflicts placed in exception queue



APPLY PROCESS

- **Types of DDL Changes Ignored by an Apply Process**

  CREATE/ALTER/DROP MATERIALIZED VIEW

  CREATE/ALTER/DROP MATERIALIZED VIEW LOG

  CREATE/DROP DATABASE LINK

  CREATE SCHEMA AUTHORIZATION

  ALTER DATABASE/SESSION/SYSTEM

  RENAME

- Apply process records information about it in the trace file for the apply process.

Apply Process Components

- A reader server that dequeue events and returns the assembled transactions to the coordinator,
- A coordinator process that gets transactions from the reader and passes them to apply servers.
- One or more apply servers that apply LCRs to database objects as DML or DDL statements or that pass the LCRs to their appropriate handlers.

- The parallelism specifies the number of apply servers that may concurrently apply transactions.e.g, if parallelism is set to 5, then an apply process uses a total of five apply servers.

NOTE : Make sure the PROCESSES and PARALLEL_MAX_SERVERS initialization parameters are set appropriately.

APPLY PROCESS - Creation

```
BEGIN
    DBMS_STREAMS_ADM.ADD_TABLE_RULES(
    table_name => 'hr.employees',
    streams_type => 'apply',
    streams_name => 'apply_emp',
    queue_name => 'strmadmin.streams_queue',
    include_dml => true,
    include_ddl => false,
    source_database => 'cpap.net');
END;
```

APPLY PROCESS - Start

```
BEGIN
      DBMS_APPLY_ADM.SET_PARAMETER(
      apply_name => 'apply_emp',
      parameter => 'disable_on_error',
      value => 'n');
END;
/
BEGIN
      DBMS_APPLY_ADM.START_APPLY(
      apply_name => 'apply_emp');
END;
```

APPLY PROCESS - Commit

Apply servers may apply transactions at the destination in an order that is different from the commit order at the source

```
BEGIN
    DBMS_APPLY_ADM.SET_PARAMETER(
    apply_name => 'strm01_apply',
    parameter => 'commit_serialization',
    value => 'none');
END;
```

Commit Serialization has the following value:

- full: Default and order is same as at source database
- none: Commit transactions in any order. Performance is best if you specify this value.

More Facts

Missing Columns at the Destination Database

Apply process raises an error and moves the transaction into an exception queue.

Fix :- Creating a rule-based transformation or DML handler that eliminates the missing columns from the LCRs before they are applied.

Column Datatype Mismatch

Apply process places transactions into an exception queue.

Fix :- Create a rule-based transformation or DML handler that converts the datatype.

APPLY PROCESS - More Facts

Extra Columns at the Destination Database

- Apply process check for Dependency for the column and If the extra columns are not used for dependency computations, then applies changes to the destination table.
- If column defaults exist for the extra columns, then these defaults are used for these columns for all inserts.

Avoid System Generated Names

- For example, DDL statement at a source database:

  CREATE TABLE EMP (n1 NUMBER NOT NULL);

- This results in a NOT NULL constraint with a system-generated name. For example, sys_001500.
- When DDL is applied at a destination , the system-generated name for this constraint may be sys_c1000.
- Again DDL statement at the source database:

  ALTER TABLE EMP DROP CONSTRAINT sys_001500;

- It fails at the destination database during the apply process and so Fix is

CREATE TABLE EMP

(n1 NUMBER CONSTRAINT emp_null_nn NOT NULL);

Streams Application

- Streams can be deployed to meet a variety of information sharing requirements
  - Replication
  - Data Warehouse Loading
  - Event Notification
  - Message Queuing

Replication

- Streams asynchronously replicate the data to large number of servers via automatic apply
- Streams automatically captures, propagates, and applies DML as well as DDL changes
- Detects and optionally resolves conflicts for the Update using Max,Min,Overwrite,Discard etc method
- Supports flexible data movement and subsetting
- Gateways and APIs for heterogeneous support
- You can configure a Streams environment to make it bi-directional.

Streams Replication Benefits

- No quiesce for DDL changes
- Lower overhead on production system
- Flexible configurations
- Reduced network traffic

Advance Replication uses 2-phase commit protocol distributed transactions which involves a great deal of waiting for acknowledgements, thus reducing performance.

Streams uses a distributed messaging protocol that send data with minimal acknowledgements from the remote database, drastically improving performance and guaranteeing once-and-only-once transactional delivery of messages.

Data Warehouse Loading

- Streams can load data warehouse staging areas and Operational Data Stores by updates captured directly from a production system Redo log files.
- Supports continuous or batch loading
- Support for data transformations and user-defined apply procedures allows the necessary flexibility to reformat data or update warehouse-specific data fields as data is loaded.

Data Warehousing Loading benefits

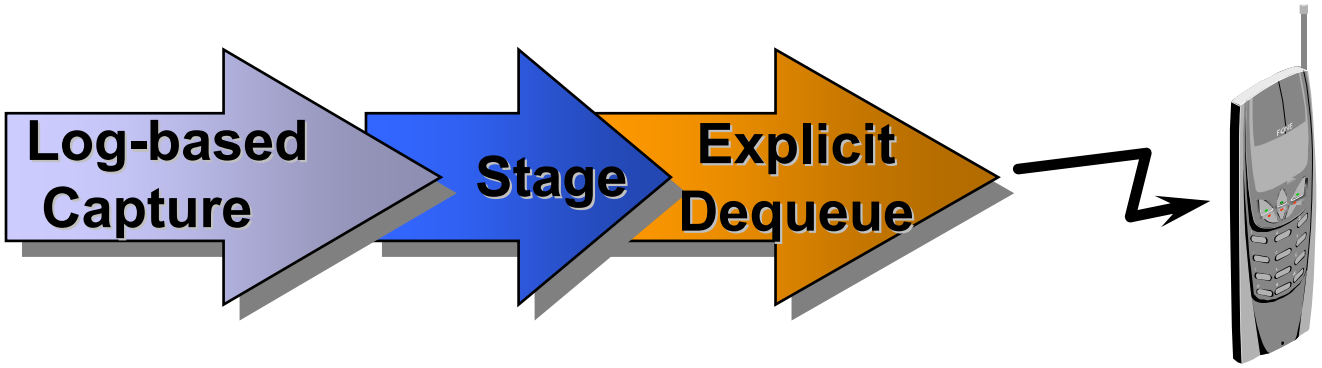
- low overhead due to Direct Redo log file reads as well as Reduced Network Traffic
- Automatic transformation with no Extra steps
- Near real-time loading of operation data stores

Event Notification

- Streams can notify subscribers that events of interest have occurred
  - News Alerts from several News provider
  - Pager notification of flight delays
  - Notification of price drops
- Streams can evaluate DML events and send notifications to applications that send emails, page users, etc

Event Notification Benefits

- Scalable as users can be increased
- Reduced custom development



DSI Data Softech Inc.
Complete Database Solution Provider

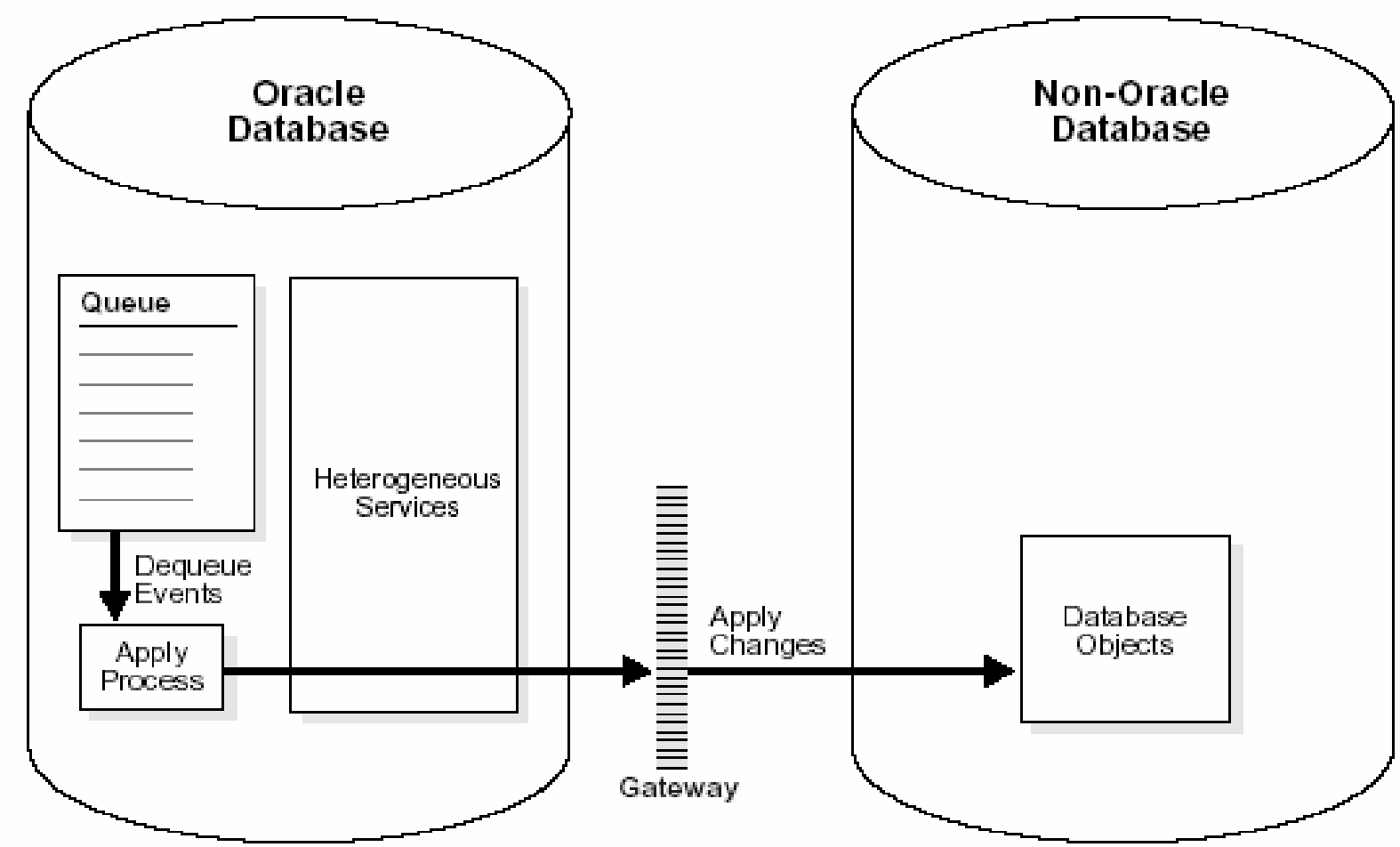
Message Queuing

- Streams allows user applications to
  - Enqueue messages of different types
  - Propagate the messages to subscribing queues,
  - Notify user applications that messages are ready for consumption, and
  - Dequeue messages at the destination database.
- Streams supports all standard features of message queuing systems, including
  - Multiconsumer queues, publishing and subscribing,
  - Content-based routing, internet propagation, transformations, and
  - Gateways to other messaging subsystems.
  - Automatic transform DML/DDL into messages

Messaging Queuing Benefits

- Reduced development costs
- Easy database integration
- Single development, operational, security model

Oracle → Non Oracle Replication

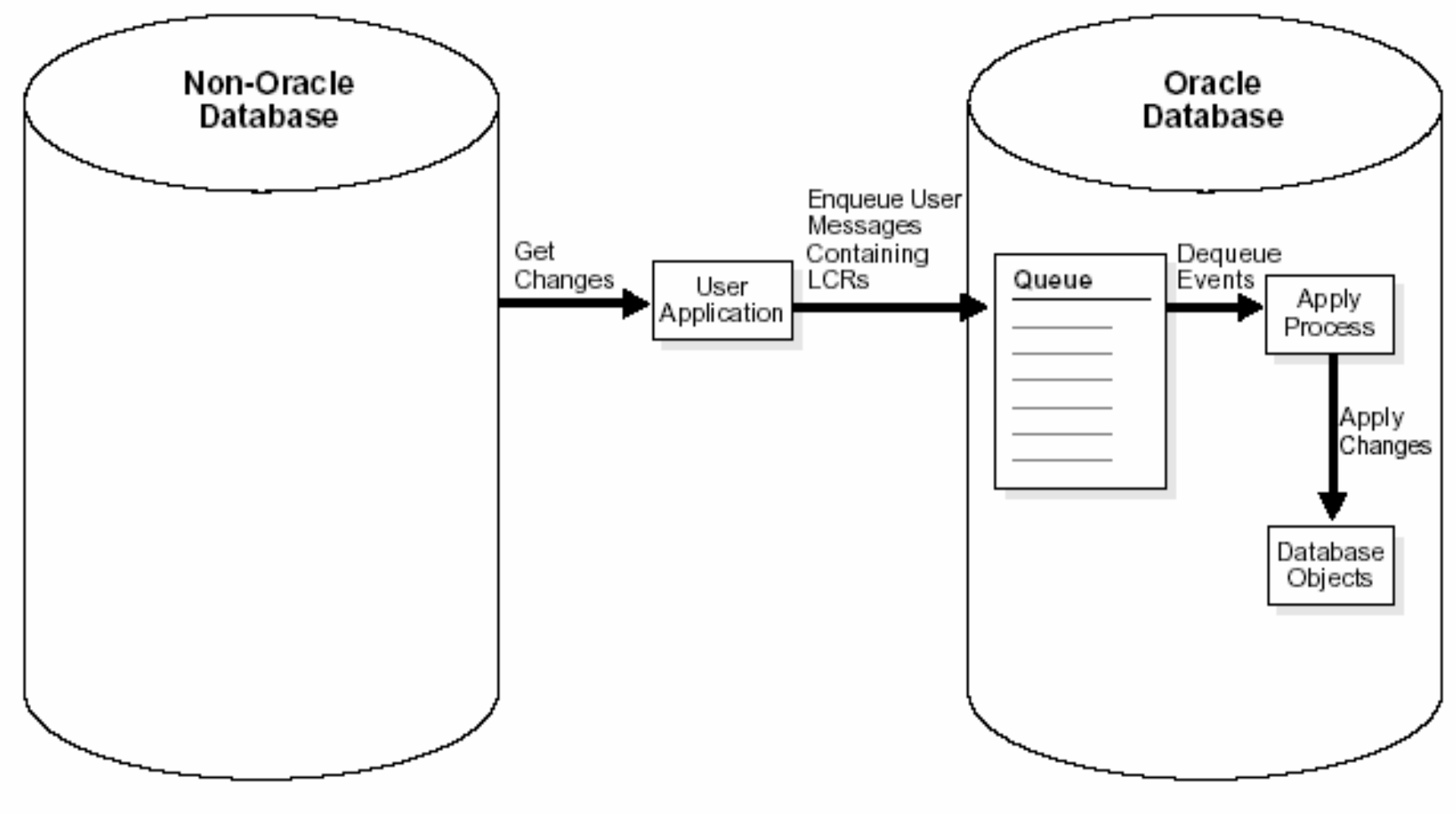


DSI DATA SOFTECH INC.
Complete Database Solution Provider

Oracle → Non Oracle Replication

- Parallel apply to non-Oracle databases is not supported.
- Error handlers and conflict handlers are not supported
- If an apply error occurs, then the transaction moved into an exception queue in the Oracle database.
- The apply process detects data conflicts but automatic conflict resolution is not supported.Therefore, any data conflicts encountered are treated as apply errors.
- The apply process cannot apply DDL changes at non-Oracle databases.

Non-Oracle → Oracle Replication



DSI DATA SOFTECH INC.
Complete Database Solution Provider

Streams Vs Clusters [RAC etc]

- Streams protect from user error, media failure, or disasters. These types of failures require redundant copies of the database which is not the case in clustering.
- Streams provides better protection from corruptions. Because data is logically captured and applied, it is very unlikely a physical corruption can propagate to the logical copy of the database.
- Clustering are also bound to the system which are close to each other which is not the case in Streams
- Clusters is the preferred method for protecting from an instance or system failure.

Streams Vs Physical Standby DB

- Streams database can be updated
- A logical copy can have a different physical layout For example, it can contain additional indexes, etc
- A logical copy provides better protection from corruptions. Because data is logically captured and applied
- The production and the replica need not be running on the exact same platform as required in Standby Database
- Streams replicas can use different character sets than the production database. which is extremely important for global operations

Streams Vs Physical Standby Contd.

- Streams can lag the production database by no more than a few seconds, as the changes can be read from the online redologs as the logs are written
- Streams supports unlimited numbers of replicas. While standby databases configured with Data Guard use the LOG_ARCHIVE_DEST_ n parameter which limit to ten copies
- Streams replicas are able to instantly resume processing.
- Customers need to invest in the expertise and planning required to make a Streams-based solution robust in contrast to Oracle Supply Data Guard used in Standby maintenance
- Streams can be used only Oracle 9.0.2 or higher

Streams Vs Advanced Replication

- No Quiescing in Streams yield higher Availability and greater ease of management in propagating DDL changes
- Streams are a Non-trigger based solution and hence is fast information sharing technology
- Row Subsetting is possible which is not possible in Advance replication
- Source and Destination database can have different column name, Datatype to replicate data which is not allowed in Advance Replication

Pre-requisite for Streams

- Oracle Software Version 9.2.0.3 or higher
- Database should be in ARCHIVELOG mode
- Override Nologging operations by using

  Alter Database/Tablespace Force Logging;
- Following init.ora parameter setting
  - AQ_TM_PROCESSES to be at least 1
  - COMPATIBLE to be 9.2.0 or higher
  - GLOBAL_NAMES=true for sharing information between databases
  - JOB_QUEUE_PROCESSES to be at least 2
  - SHARED_POOL_SIZE increase by 10Mb

Important DD Views - CAPTURE

DBA_CAPTURE

DBA_CAPTURE_PARAMETERS

DBA_CAPTURE_PREPARED_DATABASE

DBA_CAPTURE_PREPARED_SCHEMA

DBA_CAPTURE_PREPARED_TABLES

V$STREAMS_CAPTURE

Important DD Views - STAGE

DBA_QUEUES

DBA_QUEUE_PUBLISHERS

DBA_QUEUE_TABLES

AQ$<queue Table Name> - Enqueue & Dequeue Information

Important DD Views - PROPAGATE

DBA_DB_LINKS

DBA_JOBS

DBA_JOBS_RUNNING

DBA_PROPAGATION

DBA_QUEUE_SCHEDULES

Important DD Views - APPLY

DBA_APPLY
DBA_APPLY_PROGRESS
DBA_APPLY_PARAMETERS
DBA_APPLY_CONFLICT_COLUMNS
DBA_APPLY_DML_HANDLERS
DBA_APPLY_ERROR
DBA_APPLY_INSTANTIATED_OBJECTS
DBA_APPLY_KEY_COLUMNS
V$STRAMS_APPLY_COORDINATOR
V$STRAMS_APPLY_READER
V$STRAMS_APPLY_SERVER

Questions
&
Answers

DATA SOFTECH INC.
Complete Database Solution Provider

Session id: 36636

Oracle 9iAS Forms Deployment : Multiple OC4J Instance

Inderpal S. Johal
Principal Consultant,
Data Softech Inc.