

ORACLE®




ORACLE
OPEN
WORLD

Your. Open. World.

Implementing Oracle Streams Replication:
Tips and Techniques from Development

Patricia McElroy
Principal Product Manager
Oracle Streams



The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Agenda

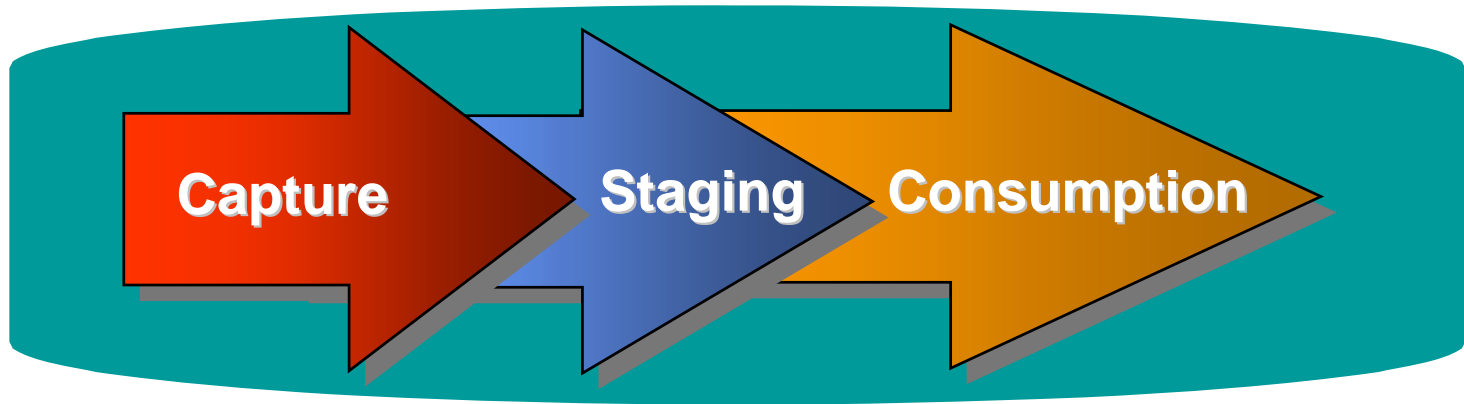
- Streams Overview
- Best Practices for Implementing Streams
- Streams Active/Active Replication
- Custom Streams Processing
- Rolling Upgrade with Streams
- Summary



Oracle Streams

- Simple solution for information sharing
- Provides
 - Uniquely flexible information sharing
 - Active/Active replication
 - Real-time ETL for data warehouses
 - Availability during database migration
 - Availability during application upgrade
 - Message queuing
 - Event management and notification

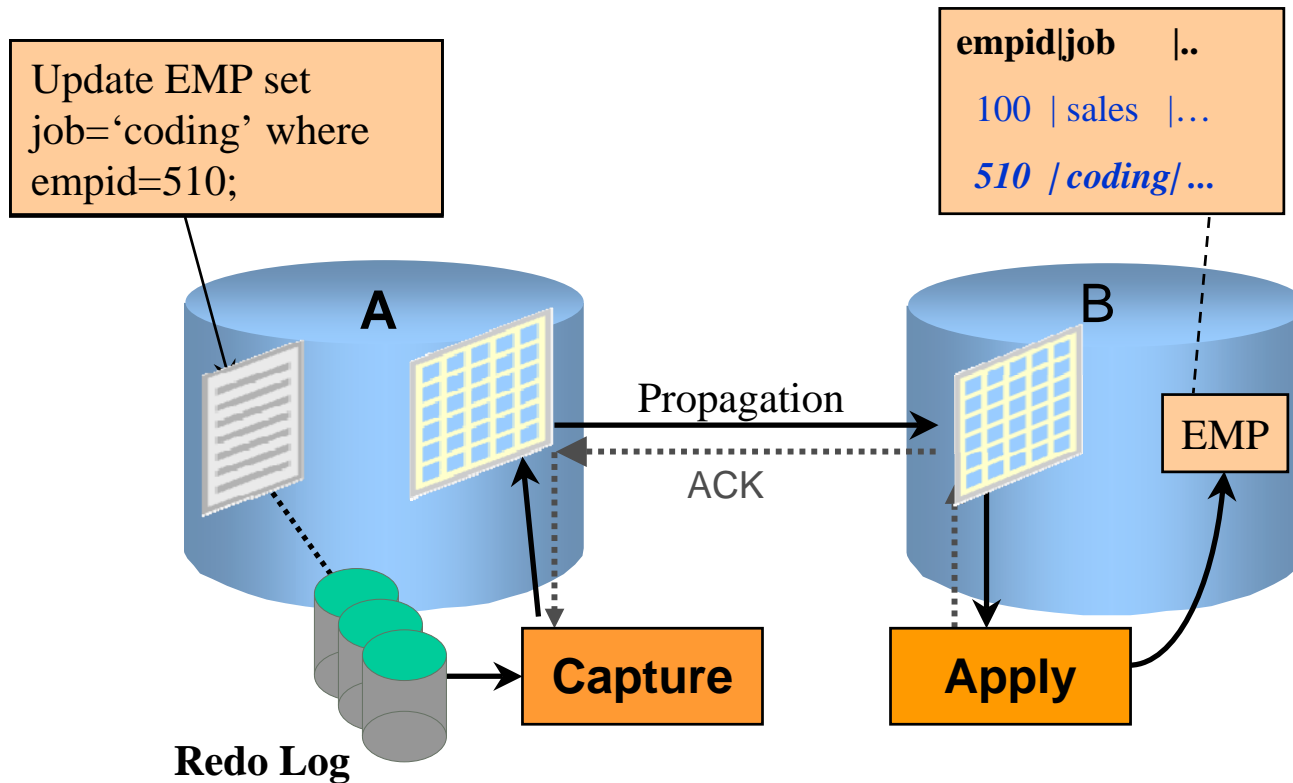
Streams: Functional Components



Asynchronous Information Sharing Infrastructure

Streams Capture and Apply

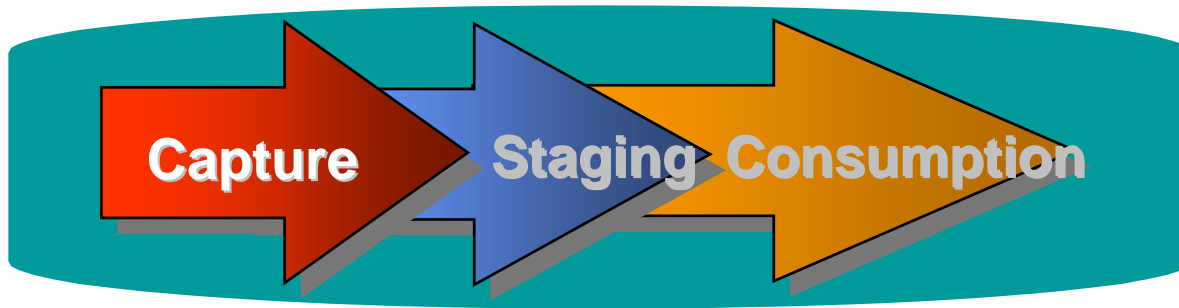
Replication Configuration



Local (Upstream) Capture

If subset of data, network bandwidth requirements reduced significantly

Capture

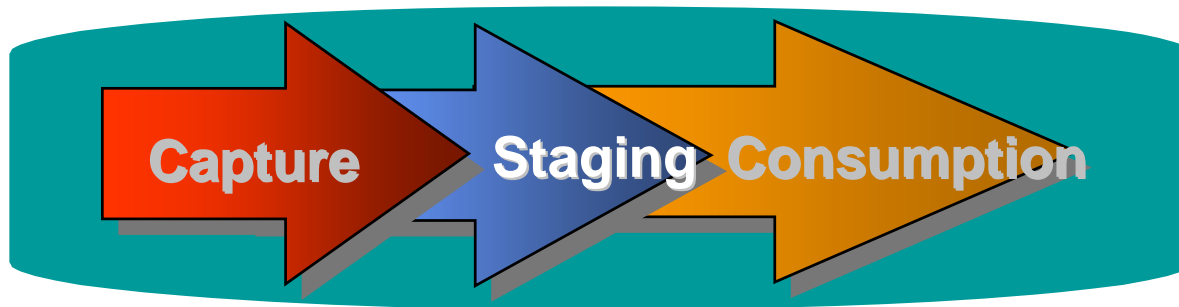


- Low overhead, low latency change capture
 - Changes to the database are written to the online redo log
 - Oracle Streams extracts changes from the log as it is written
 - Log Buffer (11g), Redo, or Archived log files
 - Local or Downstream capture capability
 - Changes are formatted as a Logical Change Record (LCR), a representation of the change
- Background process
 - Auto restart (instance restart or RAC instance failover)
 - Automatic flow control

Logical Change Record (LCR)

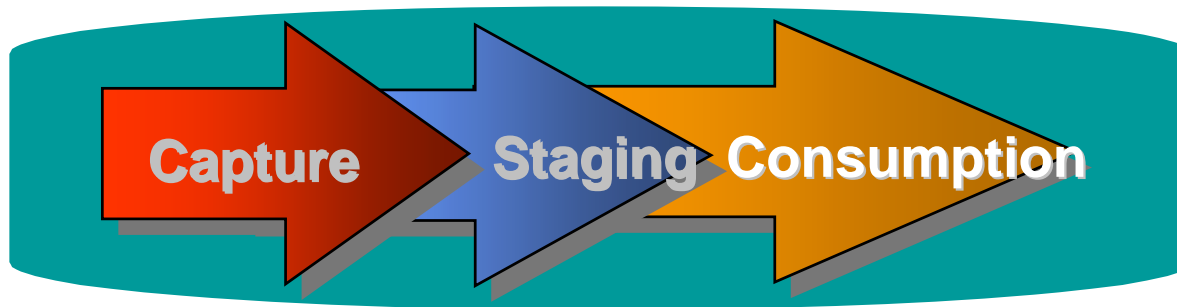
- **Database change = LCR**
 - **DML**
 - Row change = LCR
 - Object name, owner, Type of DML, SCN,...
 - OLD, NEW values
 - Optional attributes: username, session, thread,...
 - **DDL**
 - Object name, owner, Type of DDL, SCN,...
 - DDL text
 - Optional attributes: username, session, thread,...
 - **LOB and LONG columns**
 - Multiple LCRs per LOB or LONG column
 - Piecewise chunks

Staging



- Redo-based capture publishes messages into an in-memory staging area
 - Streams Pool of SGA
- Subscribers: other staging areas or Streams processes
- Messages remain in staging area until consumed by all subscribers
- Propagation between staging areas is scheduled through a propagation job
 - LCRs propagated from a queue in the local database to a queue in the target database

Apply



- The default apply engine will directly apply the DML or DDL represented in the LCR
 - Apply to local Oracle table or to non-Oracle table via dbLink
- Automatic conflict detection with optional resolution
 - Unresolved conflicts placed in error queue
 - Transactions can be re-applied or deleted from error queue
- Parallel apply maximizes concurrency
- Customizable apply processing
 - DML, DDL, Pre-commit, and Error handlers modify Apply processing

Rule-based Configuration

- Rule is expressed as SQL WHERE clause

```
dbms_rule_admin.create_rule(  
  rule_name=>'scott.rule1',  
  condition=>':dml.get_object_owner() = "SCOTT" AND  
              :dml.get_object_name()="EMP"');
```

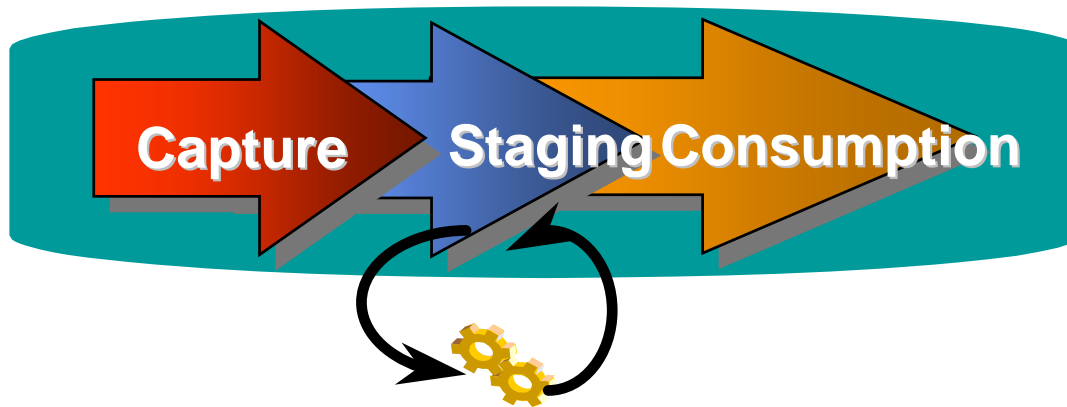
- Rule sets govern capture, staging, and apply

- Inclusion
- Negative

- Dynamic rule maintenance

```
DBMS_STREAMS_ADM.ADD_TABLE_RULES  
DBMS_STREAMS_ADM.ADD_SCHEMA_RULES  
DBMS_STREAMS_ADM.ADD_GLOBAL_RULES
```

Rule-based Transformations



- Implicit datatype conversion during Apply (11g)
- Declarative Transformations
 - Rename schema, table, column
 - Add or delete column
- Custom Transformations
 - User-supplied PL/SQL function
- Specified on a rule for capture, propagation, or apply
- A single rule can have multiple transformations

Declarative Transformations

BEGIN

```
DBMS_STREAMS_ADM.RENAME_SCHEMA(  
rule_name => 'STRMADMIN.HR51',  
from_schema_name => 'HR',  
to_schema_name => 'HR_REPL',  
step_number => 0,  
operation => 'ADD');
```

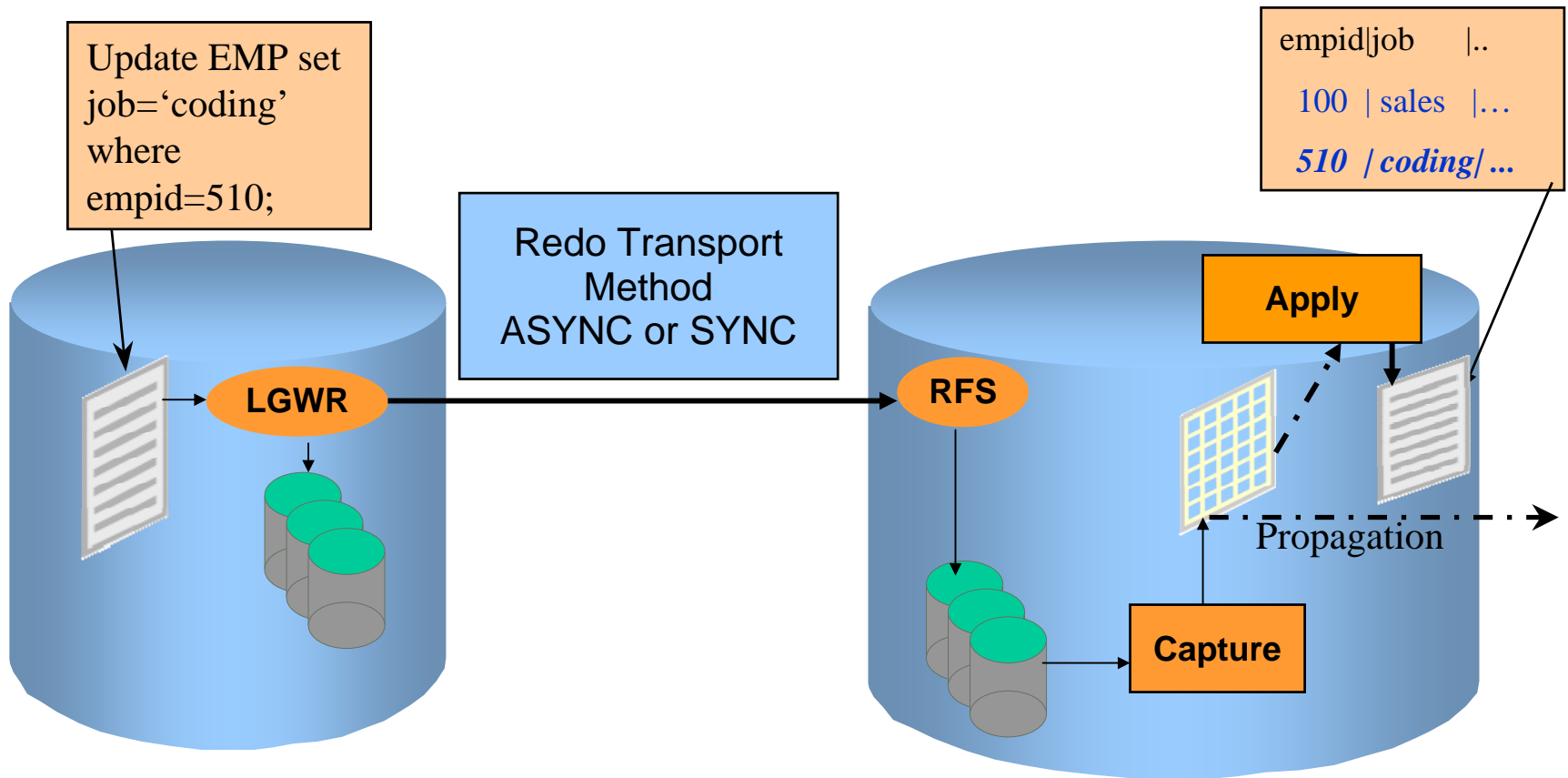
END;

/

```
SELECT rule_owner||'.'||rule_name rule,transform_type,from_schema_name,to_schema_name from  
DBA_STREAMS_TRANSFORMATIONS;
```

RULE	TYPE	FROM	TO
STRMADMIN.HR51	DECLARATIVE TRANSFORMATION	HR	HR_REPL

Downstream Capture



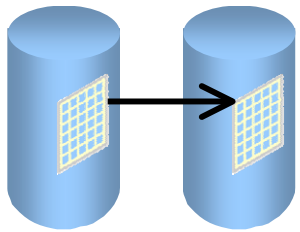
Offload Capture Processing From Production Database

Simple Configuration

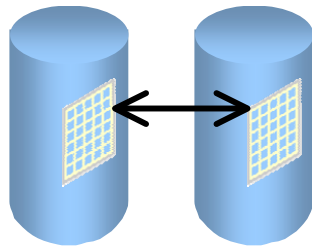
```
BEGIN dbms_streams_adm.MAINTAIN_SCHEMAS(  
    SCHEMA_NAMES          => 'HR,SCOTT',  
    SOURCE_DATABASE       => NULL,      -- local database  
    DESTINATION_DATABASE  => 'TARGET_global_name',  
    PERFORM_ACTIONS       => TRUE,  
    BI_DIRECTIONAL        => TRUE,  
    INSTANTIATION         => DBMS_STREAMS_ADM.INSTANTIATION_SCHEMA_NETWORK,  
    SOURCE_DIRECTORY_OBJECT => null,  
    DESTINATION_DIRECTORY_OBJECT => null,  
    SCRIPT_DIRECTORY_OBJECT => 'SCRIPT_DIR',  
    SCRIPT_NAME           => 'generated_schemas_script.sql'  
);  
END;  
/
```

Example Configurations using MAINTAIN_SCHEMAS

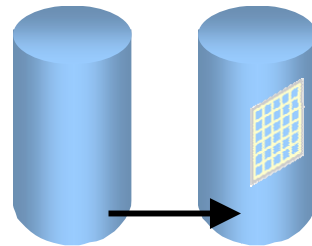
Reporting



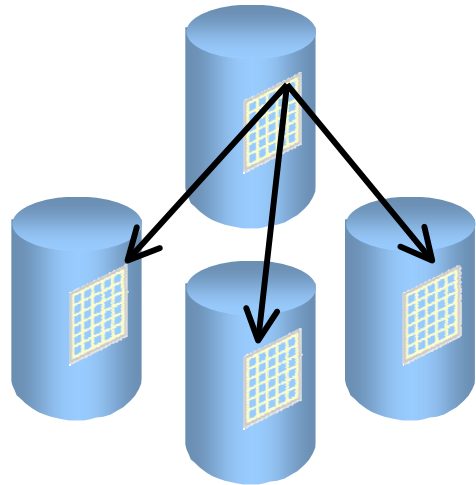
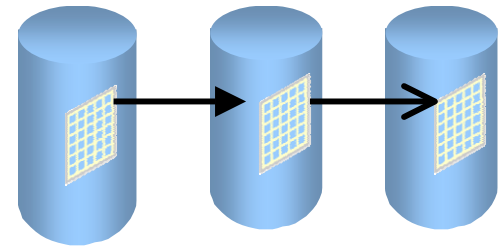
Updatable



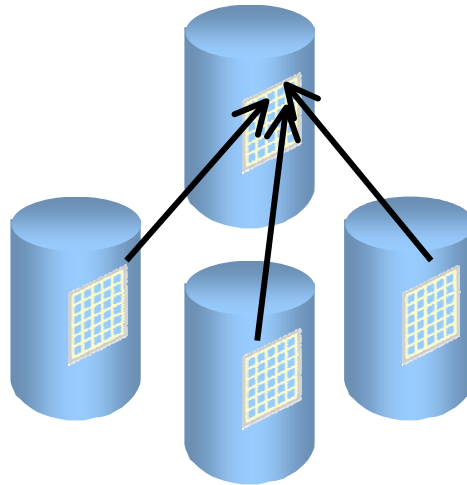
Offload Prod



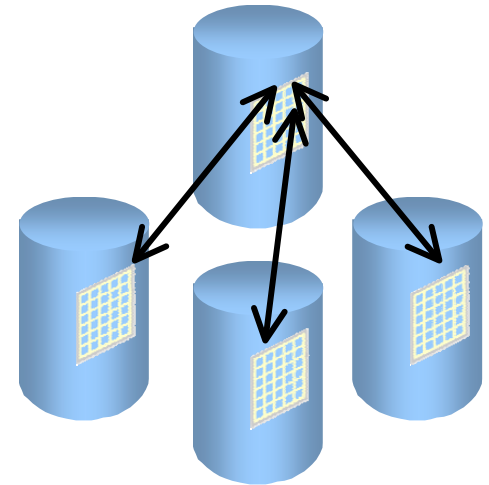
Cascade



Disseminate



Consolidate



Bidirectional

Streams in Oracle Database 11g

Improved Performance, Manageability

- Streams 11g Performance paper available on OTN
 - Demonstrates high performance throughput even across the WAN
- Performance Advisor identifies topology and analyzes end-to-end throughput.
- Split/Merge of Streams for Hub & Spoke replication
 - Maintains high performance for all replicas
 - Automated, fast “catch-up” for unavailable replica
- Table data comparison package
 - Compare data between tables in different databases
 - Identify differences by rowid
 - Converge differing rows
- Documentation: 2Day+ Data Replication & Integration

Streams Best Practices



***Ref. Chap 15 - “Best Practices for Streams Replication Databases”,
Oracle Streams Replication Administrator’s Guide 11g .***

***Ref. Chap 14 - “Best Practices for Streams Replication Databases”,
Oracle Streams Replication Administrator’s Guide 10g Release 2 (10.2)***

.

10.2 Software Updates

- Use latest patchset of Oracle Database 10g Release 2 (10.2.0.4)
- MAA paper on *Oracle Streams Configuration Best Practices*
- Metalink article [418755.1](#) lists recommended patches and parameter settings for Oracle Streams
- Streams management via Enterprise Manager
 - MAINTENANCE tab of Enterprise Manager (10.2)

Watch OTN Streams website for custom software

- <http://otn.oracle.com/products/dataint/content.html>

11.1 Software Updates

- Use latest patchset of Oracle Database 11g Release 1 (11.1.0.7)
- Streams management via Enterprise Manager
 - DATA MOVEMENT tab of Enterprise Manager

Watch OTN Streams website for custom software

- <http://otn.oracle.com/products/dataint/content.html>

General Configuration Tips

- Keep LCRs from each source database separate
- Separate queue for each capture/apply stream
- init.ora:
 - `_job_queue_interval=1` (set at the source site, indicates scan rate interval (seconds) of job queue) (**10g recommendation**)
 - `streams_pool_size=200M` (set at source and destination, ref. to `V$STREAMS_POOL_ADVICE`)
 - `aq_tm_processes` should not be set to 0 or 10 in init.ora, spfile or by ALTER SYSTEM commands.
- For WANs: (SQL*NET parameters)
 - Increase SDU (sqlnet.ora, tnsnames.ora, listener.ora)
 - Increase send_buf_size, recv_buf_size
 - Refer to MAA best practice paper: [Oracle Streams Configuration Best Practices: Oracle Database 10g Release 2 Best Practices](#)

Rule Tips

- Spelling counts!
- `source_database_name = GLOBAL_NAME` of source
 - Must be correctly specified for each process
- Eliminate duplicate or overlapping rules in a rule set
- Ensure that capture rules do not include objects with unsupported datatypes
 - `MAINTAIN_SCHEMAS` (and `GLOBAL`) avoid unsupported datatypes automatically
 - Use the negative rule set to eliminate specific tables or schemas
 - Use the `and_condition` clause on `ADD_SCHEMA_RULES` (or `GLOBAL`) to avoid tables with unsupported datatypes in 10.2 (`inclusion_rules=FALSE`)
`and_condition => ':lcr.get_compatible() > dbms_streams.compatible_10_2'`
- Avoid complex rules
 - `LIKE`
 - Functions
 - `NOT` (9.2)

Recommended Streams Parameters

CAPTURE

- Reduce the value for **checkpoint_retention_time**
 - Minimize metadata stored within database
 - Automates move of FIRST_SCN of capture
 - Managed with ALTER_CAPTURE procedure

APPLY

- Modify parameters with **SET_PARAMETER** procedure
 - Enable **parallelism** to process multiple transactions at a time (based on transaction dependencies)
 - After initial setup and validation testing complete, allow apply to continue processing when user transaction fails with **DISABLE_ON_ERROR = N**

Managing Apply Errors

Enterprise Manager

ORACLE Enterprise Manager 10g
Grid Control

Home **Targets** Deployments Alerts Compliance Jobs Reports

Hosts | Databases | Web Applications | Services | Systems | Groups | All Targets

Database Instance: patdb-viewa > Streams > **Apply Errors: APPLY\$_STRM_3** Logged in As STRMADMIN

[Common Error Messages](#) Retry all Errors Delete All

[Retry Error](#) [Delete](#)

[Select All](#) | [Select None](#)

Select	Local Transaction ID	Message Number	Message Count	Commit SCN	Source Database	Error Number	Error Message	View Error LCRs
<input type="checkbox"/>	5.17.1327	1	1	2244539	STRM.WORLD	26786	ORA-26786: A row with key ("ORDER_NUMBER") = (1028) exists but has conflicting column(s) "STATUS" in table DEMO.ORDERS ORA-01403: no data found ORA-06512: at "SYS.LCR\$_ROW_RECORD", line 440 ORA-06512: at "DEMO.ORDERS_DML_HANDLER", line 31 ORA-06512: at line 1	

Home | **Targets** | Deployments | Alerts | Compliance | Jobs | Reports | Setup | Preferences | Help | Logout

ORACLE Enterprise Manager 10g
Grid Control

Home **Targets** Deployments Alerts Compliance Jobs Reports

Hosts | Databases | Web Applications | Services | Systems | Groups | All Targets

Database Instance: patdb-viewa > Streams > [Apply Errors](#) > [View Error LCRs](#) > **Compare Values: APPLY\$_STRM_3** Logged in As STRMADMIN

- Old Value shows the old column value in the row LCR. This is the column value for the row before the DML change.
- New Value shows the new column value in the row LCR. This is the column value for the row after the DML change.
- Current Value shows the column value in the row at the destination database. This is the row that will be modified by the row LCR when the apply process applies the change.

Column Name	Column Type	Old Value	New Value	Current Value
STATUS	SYS.VARCHAR2	PENDING	Accepted	MAKEITFAIL
ORDER_NUMBER	SYS.VARCHAR2	1028	1028	1028
DELIVERY_DATE	SYS.DATE		20-SEP-08	
PART_NUMBER	SYS.VARCHAR2	4455	4455	4455

Home | **Targets** | Deployments | Alerts | Compliance | Jobs | Reports | Setup | Preferences | Help | Logout

Streams Monitoring*

- Runtime views for statistics and process state
- STRMMON – MetaLink Note:[290605.1](#) – *Oracle Streams STRMMON Monitoring Utility*
- UTLSPADV – 11g STRMMON replacement
 - Package in `$ORACLE_HOME/rdbms/admin` directory
- Healthcheck – MetaLink Note:[273674.1](#) – *Configuration Report and Health Check Script*
- Alert Log –
 - Be aware of messages from the Capture process, logged when Capture sees at least 1 large or long-running transaction in its active transaction workload

* S298778 - Performance Analysis of Oracle Streams Configurations: Maximum Availability Architecture Best Practices, 1:30 - 2:30 pm, Moscone South Room 102

Message Tracking: *Source Database*

- Enable Message Tracking in Session
- Execute DML to monitor
- Monitor View from any session
 - **Action** identifies processing within component
 - **Action_Details** provides further information about that processing

```
Run SQL Command Line
no rows selected
SQL> exec dbms_streams_adm.set_message_tracking('P1');
PL/SQL procedure successfully completed.
SQL> insert into demo.orders values ('1','1','PENDING',null);
1 row created.
SQL> commit;
Commit complete.
SQL> select component_name,action,action_details,object_name,command_type,xid,tracking_id from v$streams_message_tracking;
COMPONENT_ ACTION          ACTION_DETAILS          OBJECT_NAM COMMAND_T XID
-----
TRACKING_ID
-----
ORCL$CAP Create          ORDERS          INSERT  7.7.504
533375CE2BC3ACD8E040578CD7065FD3
ORCL$CAP Rule evaluation positive rule set: "
                    $IRADMIN"."DEMO7"
533375CE2BC3ACD8E040578CD7065FD3
ORDERS          INSERT  7.7.504
ORCL$CAP Capture send          ORDERS          INSERT  7.7.504
533375CE2BC3ACD8E040578CD7065FD3
ORCL$CAP Create          COMMIT  7.7.504
533375CE2BC4ACD8E040578CD7065FD3
ORCL$CAP Rule evaluation positive rule set: "
                    $IRADMIN"."DEMO7"
533375CE2BC4ACD8E040578CD7065FD3
COMMIT  7.7.504
ORCL$CAP Capture send          COMMIT  7.7.504
533375CE2BC4ACD8E040578CD7065FD3

6 rows selected.
SQL>
SQL>
```

Create A Heartbeat Table

- Quick status for DBA
- Generates activity in Database
- Implement periodic Job to update “heartbeat” table
 - 2 columns: Dbname, timestamp
 - Configure Streams capture/apply for “heartbeat” table
 - Configure Conflict Resolution method
MAXIMUM or OVERWRITE

```
Run SQL Command Line
SQL> create table demo.heartbeat (dbname varchar2(20), last_update timestamp, primary key (dbname));
Table created.
SQL> insert into demo.heartbeat@orcl.world values ('ORCL.WORLD', systimestamp);
1 row created.
SQL> BEGIN
2   DBMS_SCHEDULER.CREATE_JOB (
3     job_name => 'HEARTBEAT_UPDATE'
4     ,job_type => 'PLSQL_BLOCK'
5     ,job_action => 'BEGIN Update demo.heartbeat set last_update=systimestamp where dbname = (select global_name from global_name);END;'
6     ,repeat_interval => 'FREQ=MINUTELY'
7     ,enabled => TRUE
8     ,comments => 'Refreshes the Heartbeat table every second'
9   );
10 END;
11 /
PL/SQL procedure successfully completed.
SQL> connect strmadmin/strmadmin@strm.world
Connected.
SQL> DECLARE
2   cols DBMS_UTILITY.NAME_ARRAY;
3 BEGIN
4   cols(1) := 'dbname';
5   cols(2) := 'last_update';
6   DBMS_APPLY_ADM.SET_UPDATE_CONFLICT_HANDLER(
7     object_name => 'demo.heartbeat',
8     method_name => 'MAXIMUM',
9     resolution_column => 'last_update',
10    column_list => cols);
11 END;
12 /
PL/SQL procedure successfully completed.
```

Periodic Maintenance

- **DBMS_CAPTURE_ADM.BUILD()**
- **DBMS_CAPTURE_ADM.
PREPARE_GLOBAL_INSTANTIATION()**
 - Default prepare enables supplemental logging
 - To avoid supplemental logging configuration, use 'NONE' instead of default 'KEYS'

Streams Active/Active Replication



Active/Active Replication

Multiple databases replicating data from shared tables and all replicas updatable

- Active/Active enables
 - Full Utilization of Hardware Investment
 - Site Autonomy & Distribution of Load
 - Geographically diverse locations
- Active/Active Considerations
 - Potential for Data Conflicts due to multiple updates to the same row at the same time at different databases
 - Appearance of Data loss (latency)
 - Application dependencies

Active/Active Replication = Streams

- Automatic Conflict Detection
- Optional Automatic Conflict Resolution (Oracle or user-supplied)
- Optimized for near real-time replication across WAN

• S300460 - Deploying Active-Active Data Centers Using Oracle Database Solutions, 9:00 - 10:00 am, Moscone South Room 103

Conflict Detection and Resolution

Deployment Tips

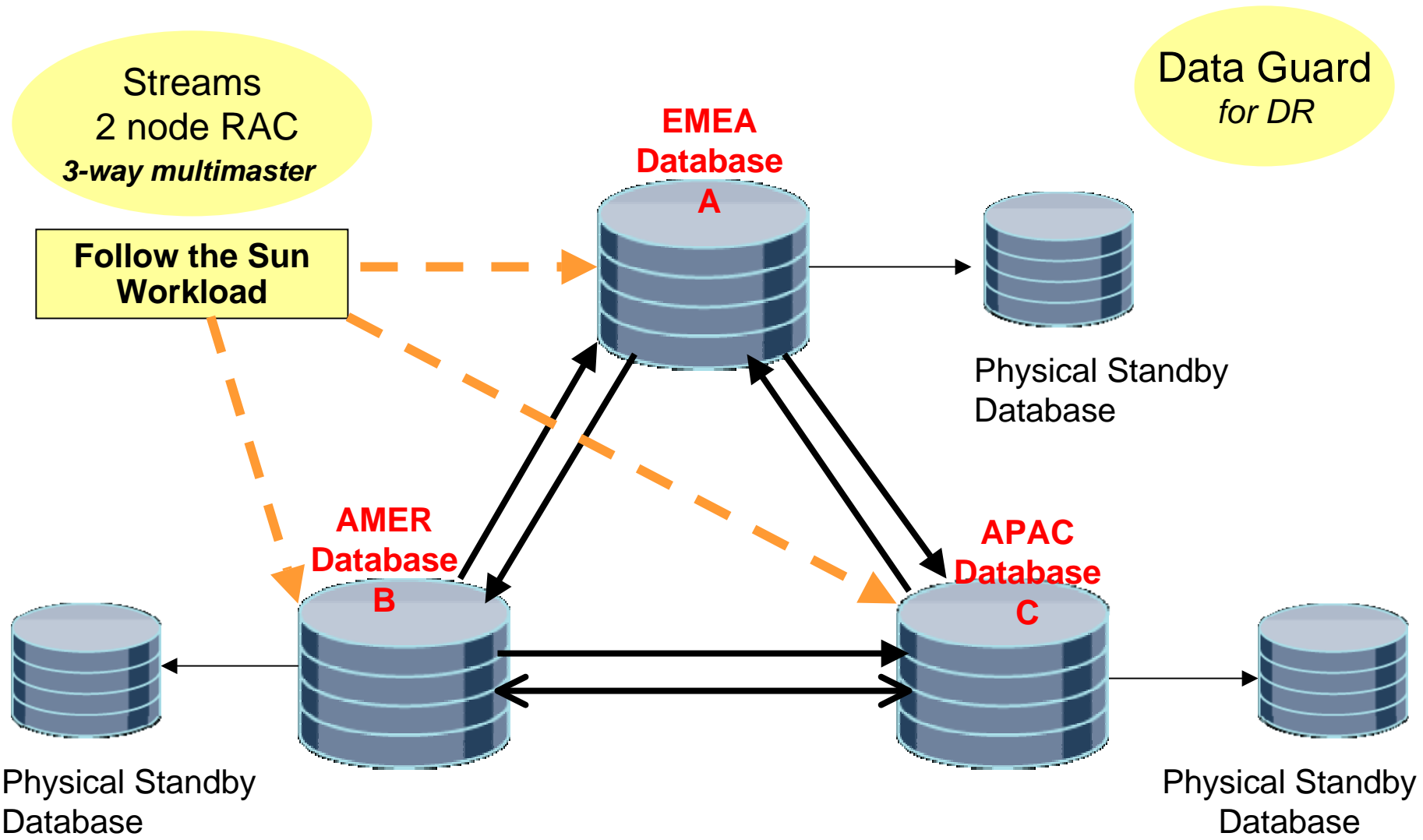
- **Potential for updates to same row at same time in multiple locations**
 - Compare “old” values in LCR to existing value at target
 - Supplemental logging for “old” values
- **Avoid conflicts, as much as possible**
 - Locally partition workload across geographic regions (ex: Follow the sun workload)
 - Partition applications across databases (ex: DB1 updates schema ABC, DB2 updates schema XYZ)
 - Ensure that user sessions are “sticky” (user affinity)
- **Implement conflict resolution methods and handlers to automate error handling**
 - For ACTIVE/ACTIVE use MAXIMUM with a TIMESTAMP column
 - MINIMUM,OVERWRITE,DISCARD possible for other scenarios
 - Error Handlers for more complex business requirements

Streams Active/Active

Deployment Tips

- **If replicating DDL, perform DDL from single database**
 - Consider impact of specific DDL across multiple databases
 - Example: Compiling procedures, functions
- **For key columns generated from sequences:**
 - Manage separate sequence ranges across databases
 - Change start and increment by settings
 - Odd/even
 - Last digit indicates database (modulo)
 - Use unique string in combination with sequence as key
- **Avoid replication cycles by setting Streams Tag**
 - Tag is null by default
 - Changes made by Streams Apply process have non-NULL tags
 - Rules indicate if tagged LCRs are replicated
 - Tag can be set in local user session to avoid replication of specific session changes.

Financial Services Company Using Streams, RAC, and Data Guard



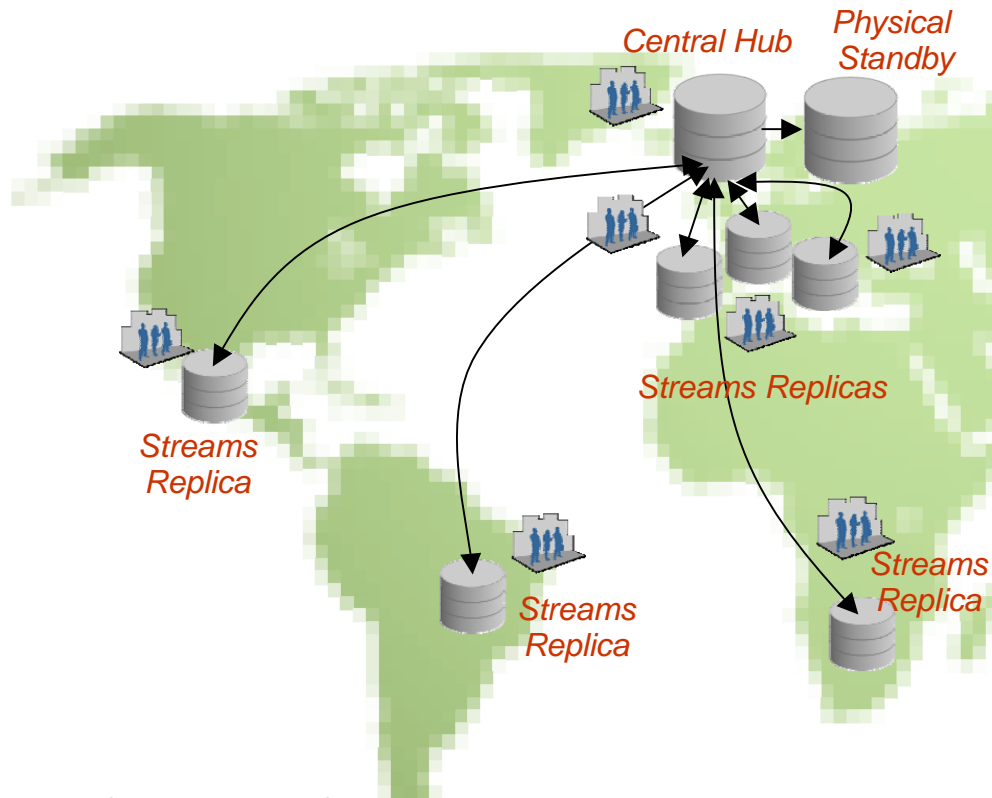
Streams and Real Application Clusters

Deployment Considerations

- **Streams Processes (Capture/Apply)**
 - Processes run on “*owning instance*” of associated queue
 - If owning instance unavailable, queue ownership automatically moves to another instance and processes restart
 - Ownership can be assigned to specific instances (primary and secondary)
 - If multiple queue tables exist within database, each can run on separate instance
- **Propagation between databases**
 - **Queue_to_queue** (set to TRUE) propagation automates delivery to specific “*owning instance*” of target queue, using registered database service
 - Database service name based on queue name and global name of database
 - Oracle Net configuration expects global name of database in `CONNECT_DATA service_name` clause

Automotive Manufacturer

Oracle Streams Hub and Spoke Configuration



- 1 TB central engineering repository (hub) maintained at HQ
- Replicas (spokes) at factories worldwide for fast, reliable, local access
- Replicas synchronized with Streams
- Physical standby for protecting central repository

- \$100,000 - \$200,000/hr per site savings in downtime costs
- Fully bi-directional, automatic conflict detection and resolution
- Minimum subset of data replicated across WAN (about 1/3 out of 200 tables)

Streams Rolling Upgrade



Database Maintenance with Streams

- Migrating between Platforms or Character Set
- Interoperate between Database Releases
- Application Upgrades
- Database Customization

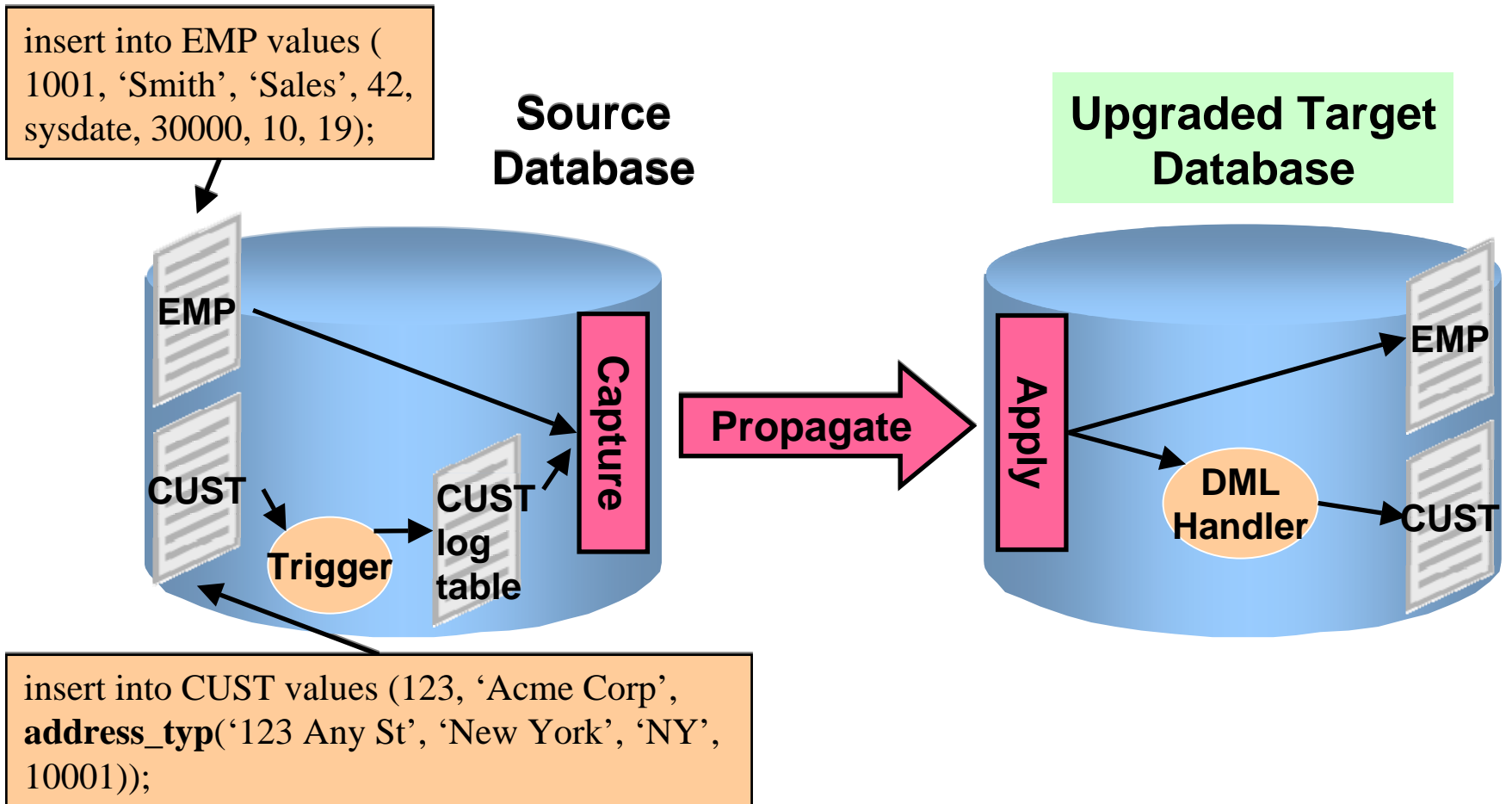
- Streams Configuration procedures
 - PRE_INSTANTIATION
 - POST_INSTANTIATION
 - CLEANUP
- Initial target database instantiation via RMAN, Physical Standby, or full exp/imp or Datapump

Extended Datatype Support (EDS)

- EDS package generates Streams configuration, table triggers and Dml Handlers to synchronize tables having these types
- For tables containing the following unsupported types:
 - Object columns with simple object types
 - Object tables
 - XMLtype
 - Varrays
 - Partial Spatial types (SDO_GEOMETRY)
- Metalink Article 556742.1

Streams Rolling Upgrade

Extended Datatype Support



Using EDS - Streams Rolling Upgrade

- Prior to Rolling Upgrade:
 - Load EDS package into Streams Administrator schema
 - Use the appropriate source setup procedure of EDS package
 - Generates scripts to create the logging table, base table trigger, and logging table trigger, and Streams directives on the source database
- During Rolling Upgrade:
 - Run generated scripts in documented order
 - Activate the physical standby as independent database (target)
 - Disable jobs at target and create database link to source database
 - Run the EDS destination setup procedure on source database
 - Perform generated scripts in documented order
 - Start Streams processes after upgrade/migration tasks complete
 - Switch users to new database when “caught up”

Managing Application Connections

- Create a service that is not managed via service_names init.ora parameter
- Configure application to use the service
- Manage service so that it is only started on the source database
- **For reference:** [Client Failover in Data Guard Configurations for Highly Available Oracle Databases](http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_ClientFailoverBestPractices.pdf)
http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_ClientFailoverBestPractices.pdf

Streams Custom Processing



Customizing the Apply Process

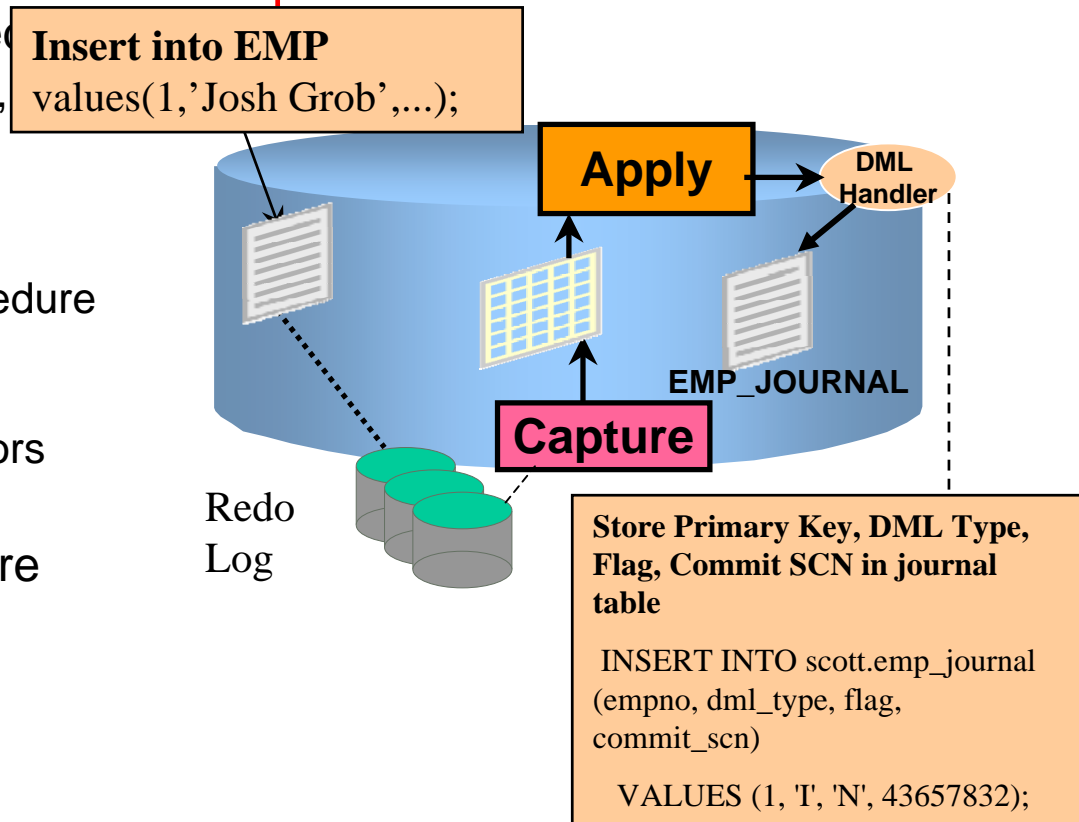
- Configure Streams for EMP table. **Example:** Journal Table

- Create custom PL/SQL procedure

- Can be used to execute SQL, perform lookups, modify data
- LCR.EXECUTE performs the change in the LCR
- Do not COMMIT within procedure (APPLY will commit at end of transaction)
- Procedure should handle errors

- Register customized procedure with APPLY as DML Handler

- Register for EMP table
- Error_handler=>FALSE
- Table and operation specific



DML Handler Procedure

```
streams_sample - Microsoft Word
File Edit View Insert Format Tools Table Window Help
1 2 3 4 5 6
-- DML handler inserts rows into journal table according to command type
-- of the LCR
-- 1) insert
-- (key columns, 'I', 'N', cscn)
-- 2) update to non-key columns
-- (key columns, 'U', 'O', cscn)
-- 3) update to key columns
-- (old key columns, 'D', 'O', cscn)
-- (new key columns, 'I', 'N', cscn)
-- 4) Delete
-- (key columns, 'D', 'O', cscn)
--
CREATE OR REPLACE PROCEDURE emp_handler(in_any IN SYS.ANYDATA)
IS
  lcr      SYS.LCR$ROW_RECORD;
  ret      PLS_INTEGER;
  ad       SYS.ANYDATA;
  cmdtyp   VARCHAR2(6);
  cscn     NUMBER;
  n$empno  NUMBER(4);
  o$empno  NUMBER(4);
  key_chgd BOOLEAN;
  o$dml_type VARCHAR2(1);
BEGIN
  -- get LCR
  ret := in_any.getobject(lcr);

  -- get command type
  cmdtyp := lcr.get_command_type;
```

Page 2 Sec 1 2/6 At 2.5" Ln 11 Col 23 REC TRK EXT OVR

Logical Change
Record Methods

```
streams_sample - Microsoft Word
File Edit View Insert Format Tools Table Window Help
Can't Undo
1 2 3 4 5 6
-- get commit scn
cscn := lcr.get_commit_scn;

--
-- populate journal table according to command type
--
IF cmdtyp = 'INSERT' THEN
  -- get new primary key values. New primary key can not be NULL
  ad := lcr.get_value('NEW', 'empno');
  IF (ad IS NOT NULL) THEN
    ret := ad.getnumber(n$empno);
  ELSE
    raise_application_error(-20000, 'New primary key value can not be NU
  END IF;

  INSERT INTO scott.emp_journal (empno, dml_type, flag, commit_scn)
  VALUES (n$empno, 'I', 'N', cscn);

ELSIF cmdtyp = 'DELETE' THEN
  -- get old primary key. Old primary key can not be NULL
  ad := lcr.get_value('OLD', 'empno');
  IF (ad IS NOT NULL) THEN
    ret := ad.getnumber(o$empno);
  ELSE
    raise_application_error(-20000, 'Old primary key value can not be NU
  END IF;

  INSERT INTO scott.emp_journal (empno, dml_type, flag, commit_scn)
  VALUES (o$empno, 'D', 'O', cscn);
```

Page 3 Sec 1 3/6 At 7.1" Ln 40 Col 1 REC TRK EXT OVR

DML Handler Procedure (cont)

```
streams_sample - Microsoft Word
File Edit View Insert Format Tools Table Window Help
1 2 3 4 5
ELSIF cmdtyp = 'UPDATE' THEN
  -- get old primary key. Old primary key can not be NULL
  ad := lcr.get_value('OLD','empno');
  IF (ad IS NOT NULL) THEN
    ret := ad.getnumber(o$empno);
  ELSE
    raise_application_error(-20000, 'Old primary key value can not be NULL');
  END IF;

  -- get new primary key values. For updates to non-primary key
  -- new primary key is null
  ad := lcr.get_value('NEW','empno', 'N');
  IF (ad IS NOT NULL) THEN
    ret := ad.getnumber(n$empno);
    key_chgd := TRUE;
  ELSE
    key_chgd := FALSE;
  END IF;

  IF key_chgd THEN
    -- if key changed, treat as delete/insert pairs
    o$dml_type := 'D';
  ELSE
    o$dml_type := 'U';
  END IF;

  INSERT INTO scott.emp_journal (empno, dml_type, flag, commit)
  VALUES (o$empno, o$dml_type, 'O', cscn);

  IF key_chgd THEN
    INSERT INTO scott.emp_journal (empno, dml_type, flag, commit)
    VALUES (n$empno, 'I', 'N', cscn);
  END IF;

```

Populate Journal Table

```
streams_sample - Microsoft Word
File Edit View Insert Format Tools Table Window Help
1 2 3 4 5
END IF;
END;
/
show errors;

-- set apply dml handler for scott.emp
BEGIN
  dbms_apply_adm.set_dml_handler(
    object_name => 'scott.emp',
    object_type => 'TABLE',
    operation_name => 'DEFAULT',
    error_handler => false,
    user_procedure => 'emp_handler',
    apply_database_link => NULL);
END;
/

select * from dba_apply_dml_handlers;

-- set instantiation SCN for scott.emp
DECLARE
  iscn NUMBER;
BEGIN
  iscn := dbms_flashback.get_system_change_number();
  dbms_apply_adm.set_table_instantiation_scn(
    source_object_name => 'scott.emp',
    source_database_name => :db_name,
    instantiation_scn => iscn);
END;
/

select * from dba_apply_instantiated_objects;

```

Register Handler with Apply

Think Outside the Box

Combine multiple features with Streams

- Triggers
- Dynamic SQL
- Flashback
- LCR modification
- DML & DDL handlers, Pre-commit handlers

Sample code available

- Procedural Replication
- Asynchronous Triggers
- Streams driven Materialized View Refresh
- Streams driven Continuous Query
- Streams Capture and Streams AQ

Summary



Oracle Streams

- Integrated Feature of the database
- Active/Active capabilities
 - Full Utilization of Hardware Investment
 - Site Autonomy & Distribution of Load
 - Geographically diverse locations
- Information Sharing between applications and databases
 - N-way multimaster (bi-directional replication), hub & spoke, many to one replication, custom capture/apply
 - Support data transformations, subsetting, custom apply functions
 - Support heterogeneous platforms and different character sets
 - Interoperate between database releases
- Reduce planned downtime
 - During database platform migrations
 - During application upgrades, when the logical structure of the database is changed

Resources

- Collateral, best practices, sample code for both Streams replication and Advanced Queuing:
<http://otn.oracle.com/products/dataint/>
- MetaLink Note [418755.1](#) – 10.2 Streams Recommendations

HA Sessions, Labs, Demos From Oracle Development

Mon, Sep 22

- 2:30 pm - Database 11g: Next-Gen HA, Moscone South 103

Tue, Sep 23

- 9:00 am - Active-Active Data Centers, Moscone South 103
- 11:30 am - Sharding with Oracle, Moscone South 302
- 11:30 am - HA with Oracle VM, Moscone West 3024
- 1:00 pm - Active Data Guard, Moscone South 104

Wed, Sep 24

- 9:00 am - Fusion Middleware Grid HA, Marriott Nob Hill AB
- 11:30 am - RMAN Best Practices, Moscone South 103
- 5:00 pm - Data Guard & Real Application Testing, Moscone 102
- 5:00 pm - EM in Secure MAA, Moscone West 2001

Wed, Sep 24 (contd.)

- 5:00 pm - E-Business Suite HA, Moscone West 2002/04

Thu, Sep 25

- 9:00 am - Oracle Secure Backup, Moscone South 102
- 10:30 am - Streams Replication, Moscone South 102
- 12:00 pm - Rolling Database Upgrades, Moscone South 103
- 1:30 pm - Streams Performance, Moscone South 102
- 3:00 pm - Oracle Grid Computing, Moscone South 303
- 3:00 pm - E-Business Suite R12 MAA, Moscone West 2007
- 3:00 pm - Siebel MAA, Moscone South 308
- 3:00 pm - Fusion SOA HA & Scalability, Marriott Salon 14/15

Hands On Labs - Thu, Sep 25

- 10:30 - 11:30 am, 12:00 - 1:00 pm - Active Data Guard, Marriott Golden Gate A3

DEMOgrounds, Mon-Thu

- Active Data Guard, Streams, Oracle Secure Backup, RMAN/Flashback, MAA

For More Information

search.oracle.com



or

oracle.com

ORACLE®