

ORACLE®



ORACLE®

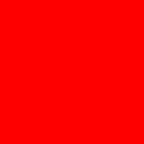


Implementing Replication with Oracle Streams

Ashish Ray

Director of Product Management

Database High Availability, Server Technologies

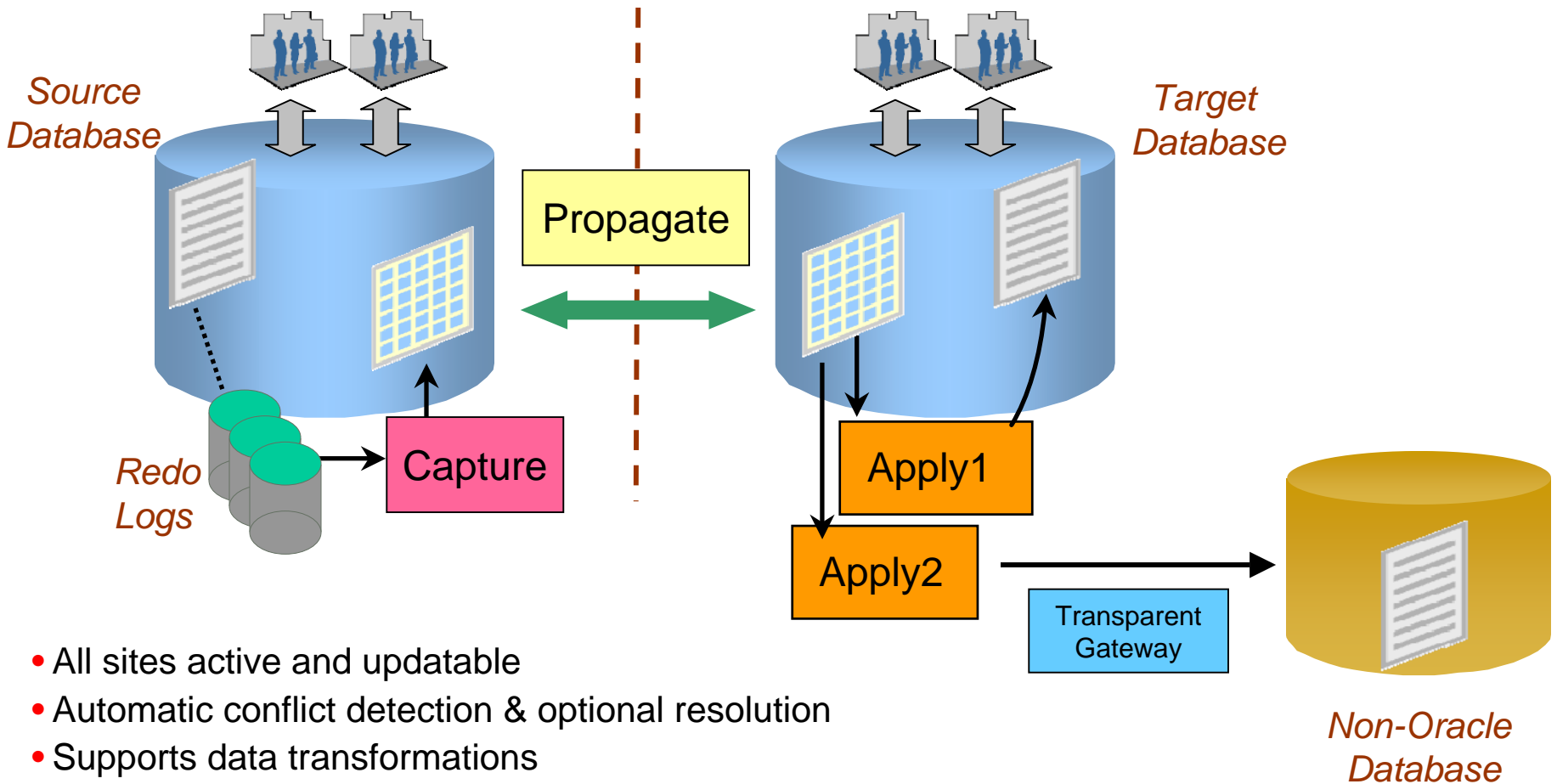


The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development,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

What is Oracle Streams?

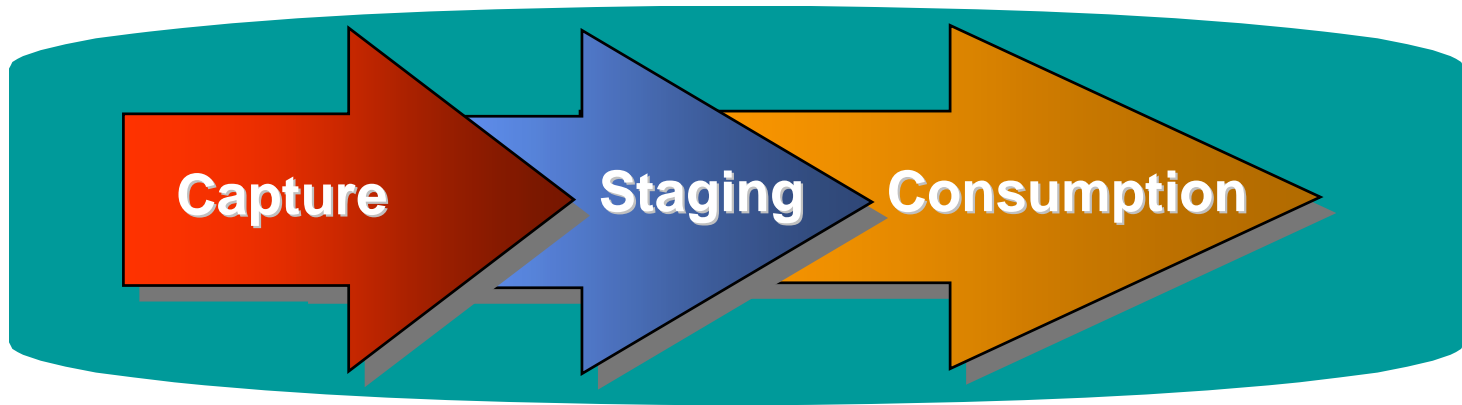
- Simple solution for information sharing
- Provides
 - replication
 - message queuing
 - data warehouse loading and simple ETL
 - database migration
 - application upgrade
 - event management and notification

Streams: Architecture Overview



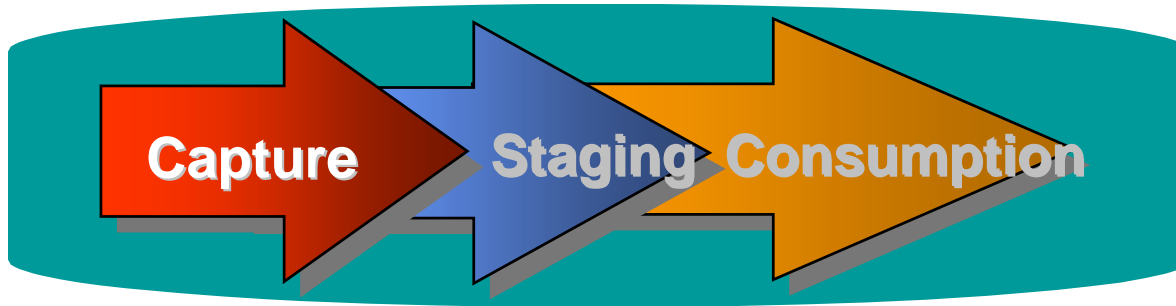
- All sites active and updatable
- Automatic conflict detection & optional resolution
- Supports data transformations
- Flexible configurations – n-way, hub & spoke, ...
- Database platform / release / schema structure can differ
- Provides HA for custom apps where update conflicts can be avoided or managed

Streams: Functional Components



Asynchronous Information Sharing Infrastructure

Capture

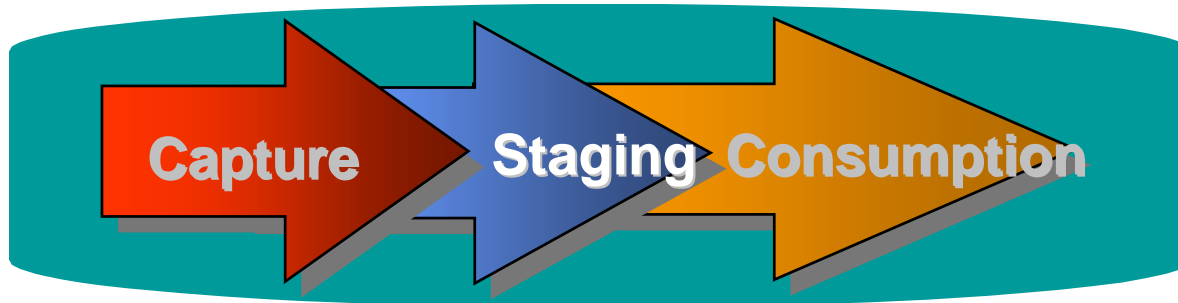


- Low overhead, low latency change capture
 - Changes to the database are written to the online redo log
 - Oracle Streams extracts changes from the log as it is written
 - Redo or Archived log files
 - Local or Downstream capture capability
 - Changes are formatted as a Logical Change Record (LCR),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

Logical Change Record (LCR)

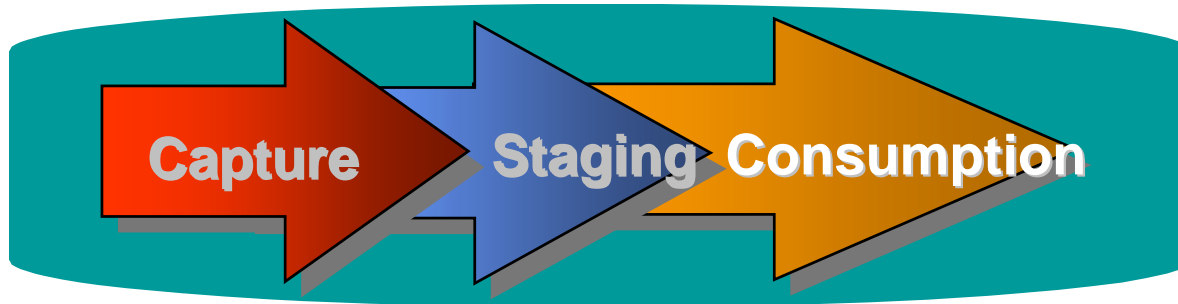
- Database change = LCR
 - DML
 - Row change = LCR
 - Object name, owner, Type of DML, SCN,...
 - OLD, NEW values
 - Optional attributes: username, session, thread,...
 - DDL
 - Object name, owner, Type of DDL, SCN,...
 - DDL text
 - Optional attributes: username, session, thread,...
 - LOB and LONG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

Staging



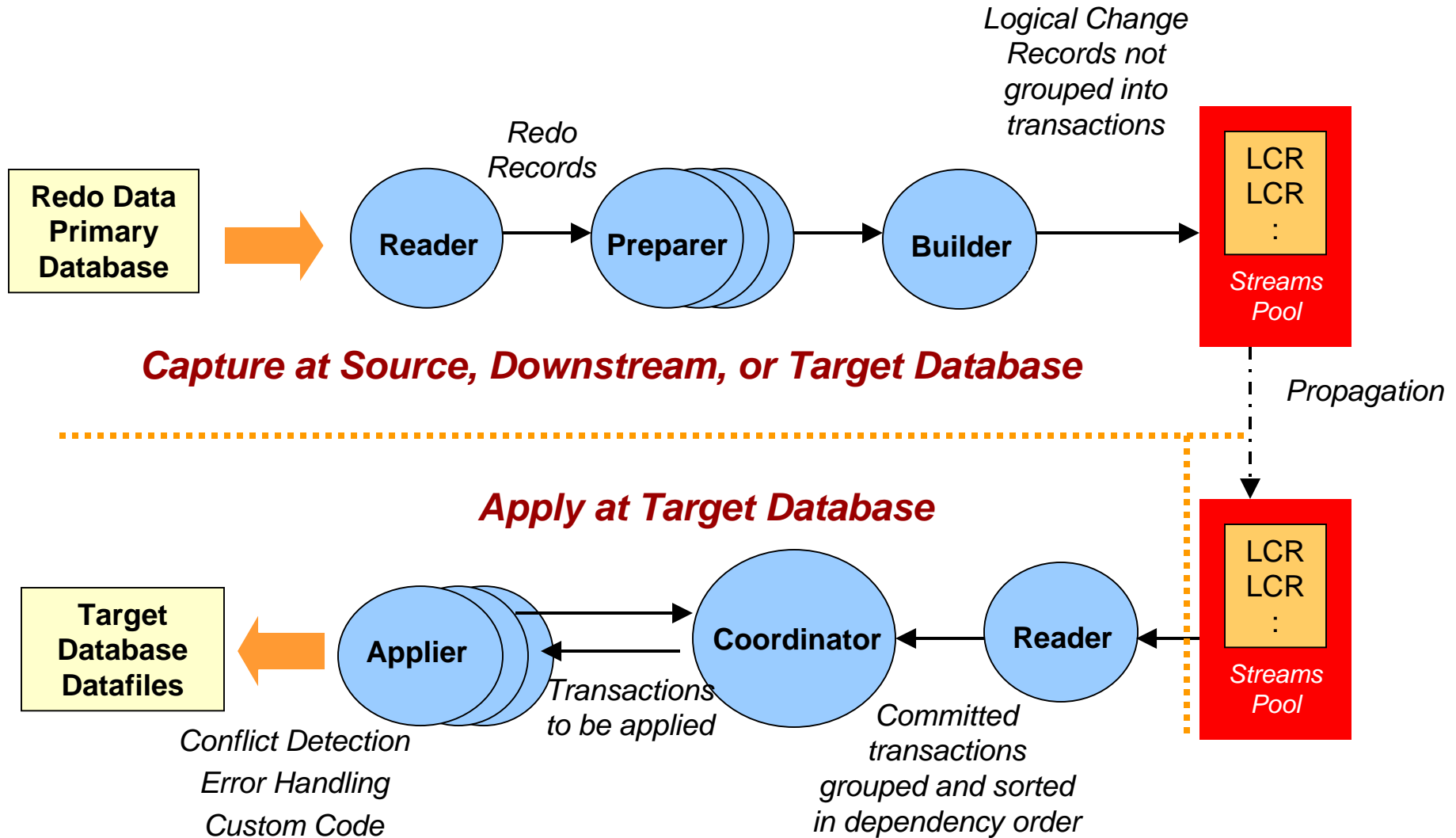
- Streams publishes captured events into an in-memory staging area
 - Implemented as ANYDATA queue
 - Streams Capture uses buffered ANYDATA queue
 - Streams Pool of SGA
 - Subscribers: other staging areas or Streams processes
 - Messages remain in staging area until consumed by all subscribers
- Propagation 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

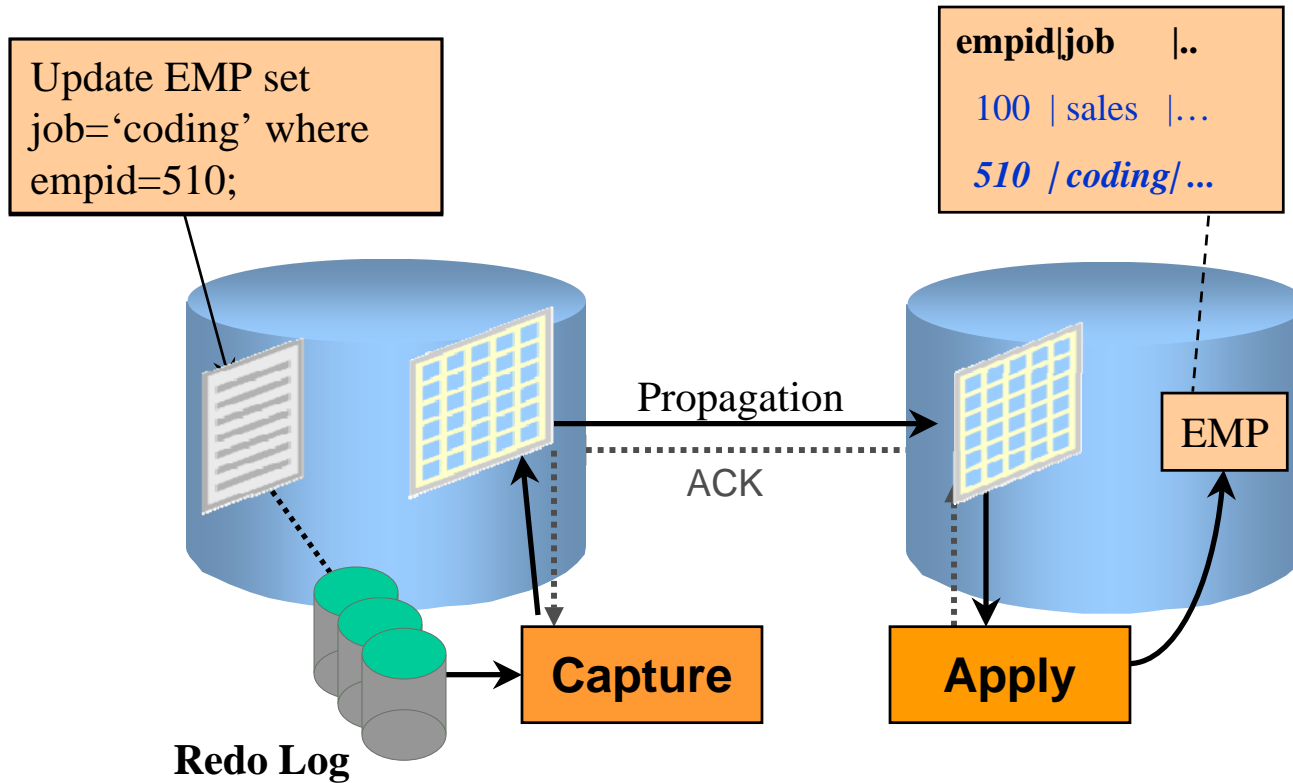


- The default apply engine will directly apply the DML or DDL represented in the LCR
 - Apply to local Oracle table
 - Apply via DB Link to non-Oracle table
- Automatic conflict detection with optional resolution
 - Unresolved conflicts placed in error queue
 - Transactions can be re-applied or deleted from error queue
- Parallel apply maximizes concurrency

Streams Process Architecture

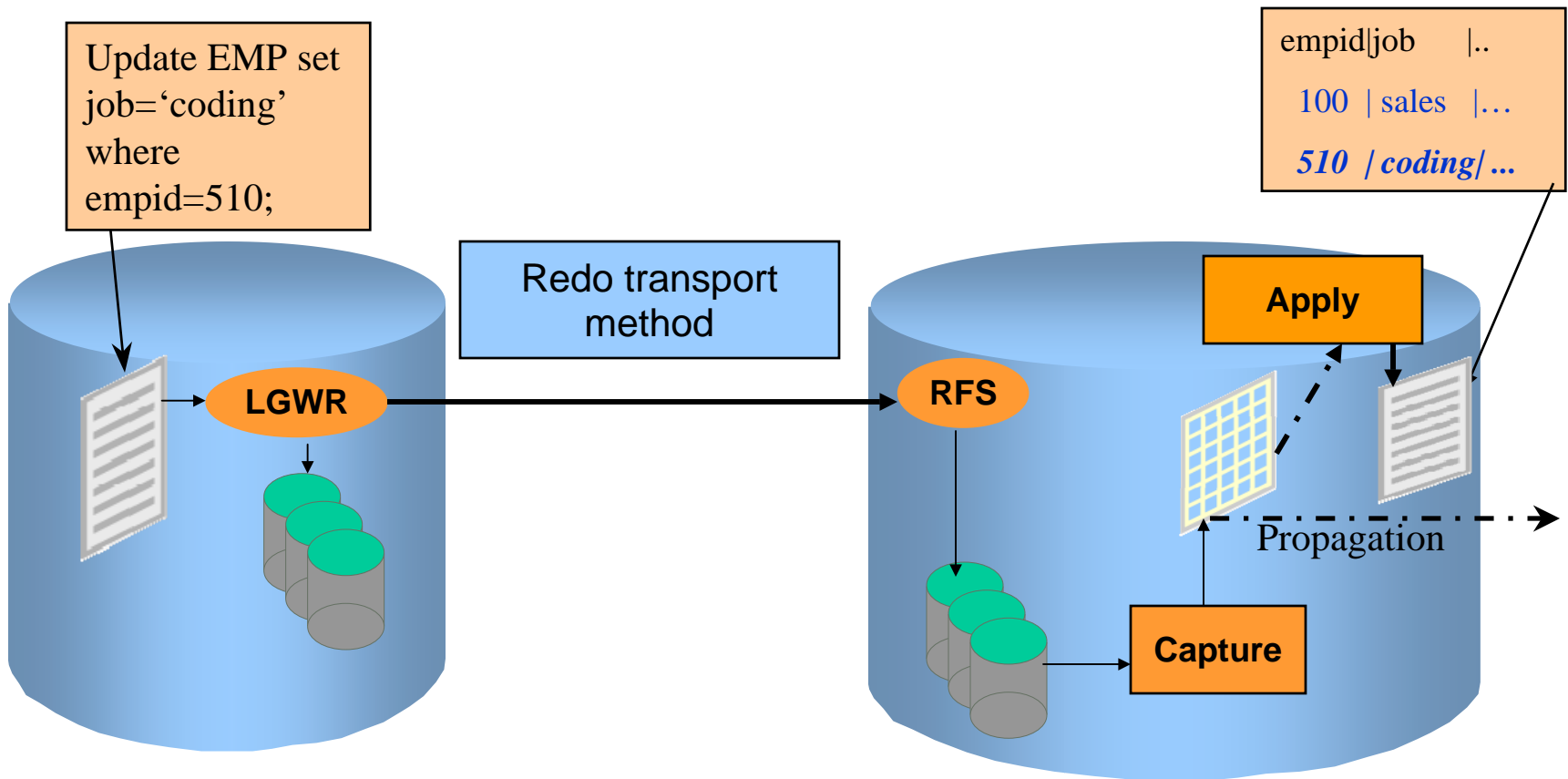


Streams Capture and Apply: Replication Configuration



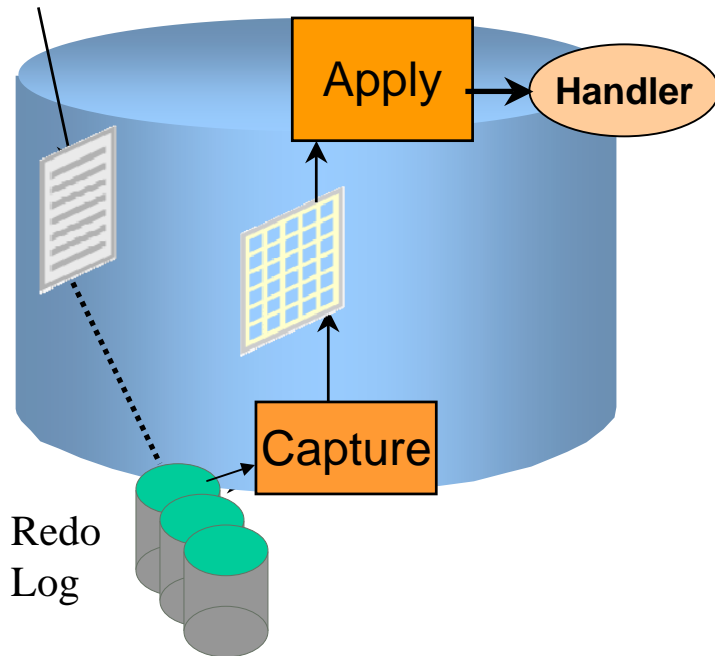
Local Capture

Downstream Capture



Streams Capture with Local Customized Apply

Update EMP set job=
'coding' where
empid= 510;



Local Capture

PL/SQL Package: **DBMS_APPLY_ADM**

LCR:

DML Handler

- Per table per operation (Insert/Update/Delete/Lob_Update)
- **SET_DML_HANDLER ()**

DDL Handler

- Defined for entire apply process
- **ALTER_APPLY ()**

Apply Directives

- **SET_ENQUEUE_DESTINATION ()**
- **SET_EXECUTE ()**

Transaction:

Pre Commit

- Defined for entire apply process
- **ALTER_APPLY ()**

Rule-based Configuration

- Consumers subscribe to published events
- Rule is expressed as SQL WHERE clause

```
dbms_rule_adm.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
- Tailored Replication API

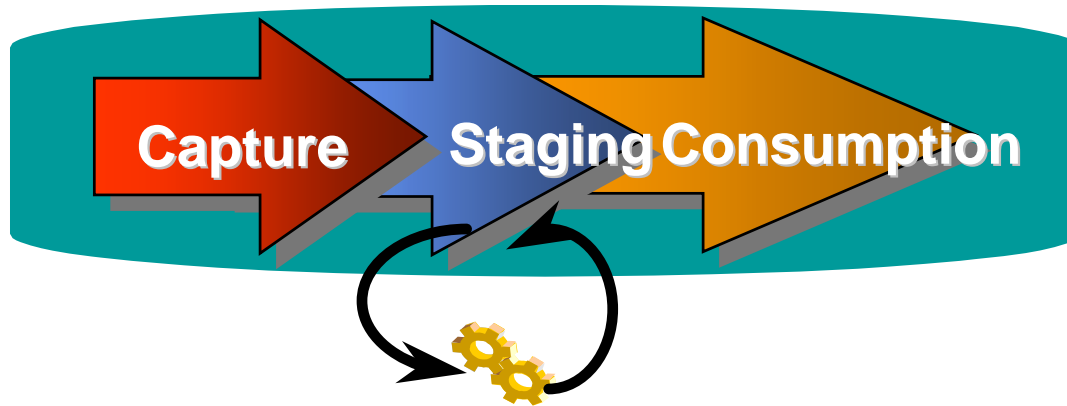
```
DBMS_STREAMS_ADM.MAINTAIN_*  
  {Tables | Schemas | Global | TTS}  
DBMS_STREAMS_ADM.ADD_*_RULES  
  {Table | Schema | Global | Subset}
```

10.2: Simple Configuration

Schema Replication

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

Rule-based Transformations



- Transformations can be performed:
 - As events enter the staging area
 - As events leave the staging area
 - As events propagate between staging areas
- Declarative Transformations
 - Rename schema, table, column
 - Add or delete column
- Custom Transformations
 - User-supplied PL/SQL function

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

Conflict Resolution

- Conflicts can occur if DML changes are allowed to same row of a table at multiple locations
- Conflicts always detected
- Apply attempts conflict resolution,if configured
 - Multiple supplied conflict resolution routines, or user-written
 - Conflicting columns identified by Primary Key or substitute key
 - Supplemental logging at source required for any columns involved in conflict resolution at destination

Handling Errors

- Error queue stores information about transactions that could not be successfully applied
 - Created automatically when queue created
 - Only contains information about local errors
 - Contains all LCRs for each error transaction
- **DBMS_APPLY_ADM.EXECUTE_ERROR()** has optional error handler capability
- Apply Error Management
 - **MESSAGE_NUMBER** column in **DBA_APPLY_ERROR**
 - Ref. scripts in Manual “*Oracle Streams Concepts and Administration*”, Chap 22 – “*Monitoring Streams Apply Processes*”, and Section – “*Displaying Detailed Information About Apply Errors*”

Error Transaction Retry

- Execute_error can specify a *user_procedure*

```
BEGIN DBMS_APPLY_ADM.EXECUTE_ERROR(  
    local_transaction_id => '5.6.924',  
    execute_as_user => false,  
    user_procedure => 'strmadmin.modify_emp_salary');  
END;  
/
```

- User Procedure signature

```
PROCEDURE strmadmin.modify_emp_salary(  
    in_any IN ANYDATA,  
    error_record IN DBA_APPLY_ERROR%ROWTYPE,  
    error_message_number IN NUMBER,  
    messaging_default_processing IN OUT BOOLEAN,  
    out_any OUT ANYDATA)
```

Example: user_procedure

- `Execute_error('1.1.1',user_procedure=>'MY_ERR_PROC');`

```
create or replace procedure strmadmin.my_err_proc (a in anydata,  
  ae in dba_apply_error%rowtype,  
  en in number,  
  messaging_default_processing in out boolean,  
  out_any out anydata) AS
```

```
  typenm  VARCHAR2(61);  
  rowlcr  SYS.LCR$_ROW_RECORD;  
  res     NUMBER;
```

```
BEGIN
```

```
  out_any := a;
```

```
  if (en = 107) then
```

```
    res := a.GETOBJECT(rowlcr);
```

```
    rowlcr.set_value('old', 'SALARY', Sys.AnyData.ConvertNumber(8000));
```

```
    messaging_default_processing := TRUE;
```

```
    out_any := Sys.AnyData.ConvertObject(rowlcr);
```

```
  end if;
```

```
END;
```

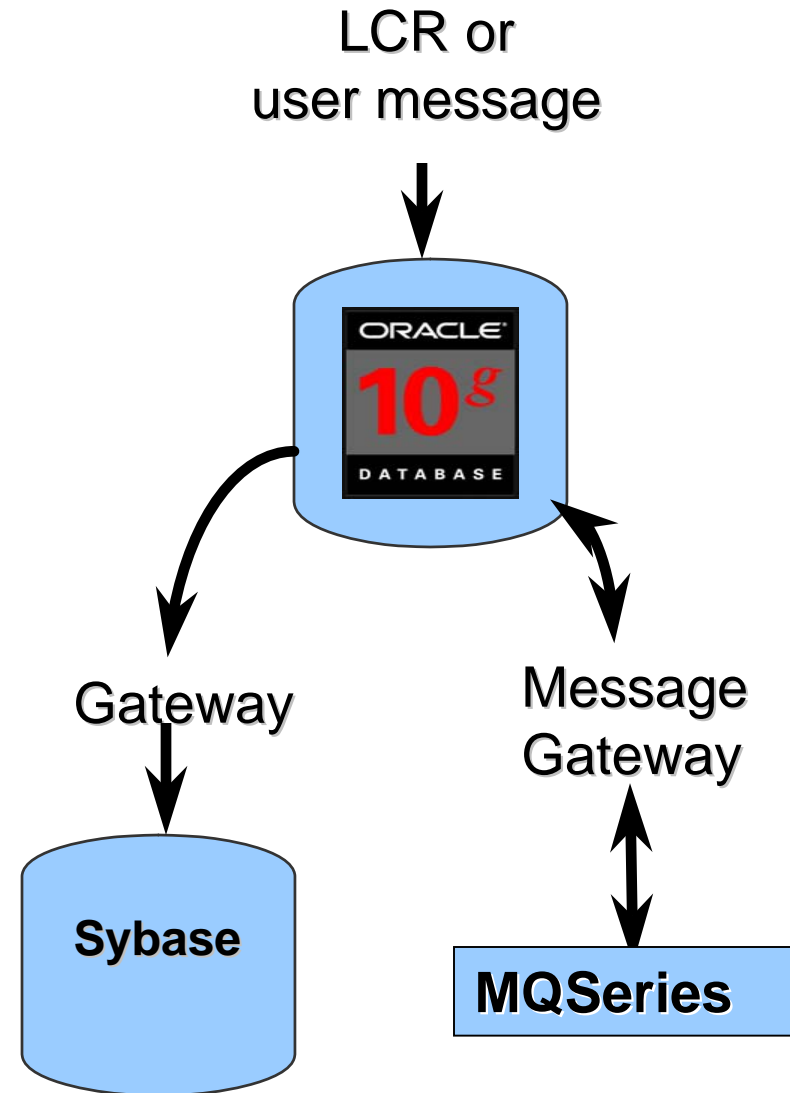
```
/
```

'en' is the lcr that failed:

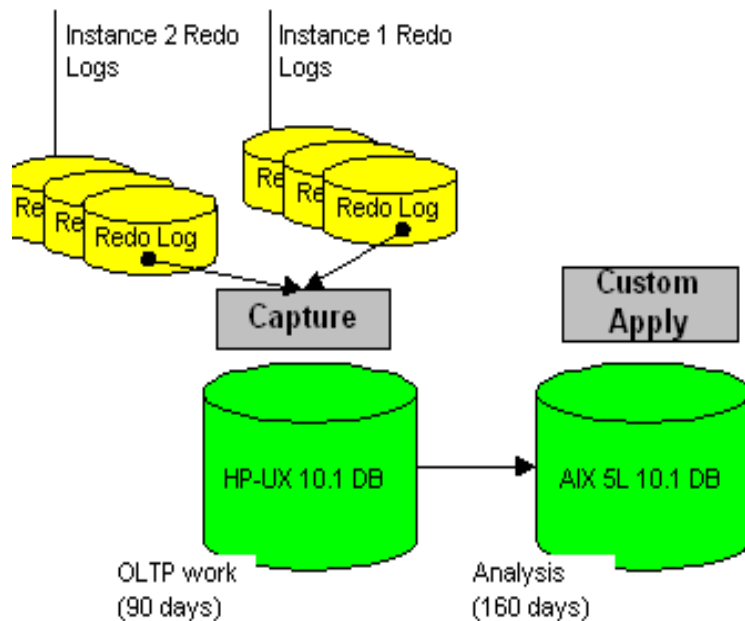
MESSAGE_NUMBER in DBA_APPLY_ERROR

Heterogeneous Support

- Oracle to non-Oracle Apply via gateway
 - Apply process on Oracle node applies change
- Non-Oracle to Oracle change capture supported via explicit enqueue of LCRs
- Message Gateways
 - MQSeries
 - Tibco



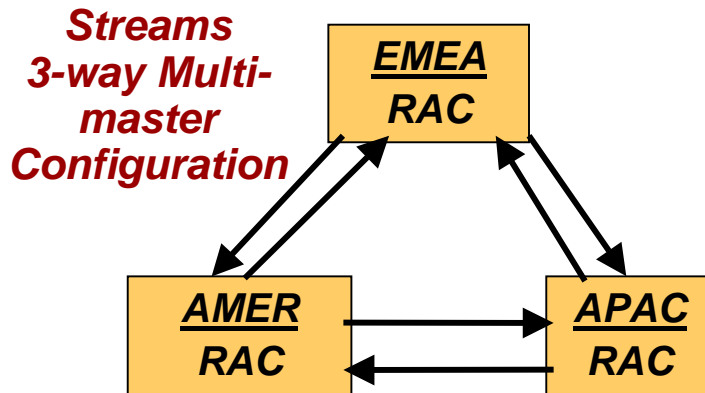
Case Study #1: 10g RAC->10g



- Large manufacturing factory
- Offload reporting
- Constant workload (manufacturing 24/7)
- Customized apply: track all transactions

Case Study #2

10g RAC<-->10gRAC



Each Database

3 queues: 1 for each process

1 capture (local changes)

2 apply processes

- changes from each of the other source databases

- Large investment bank:
 - Linux 10g RAC (2node) databases:
 - Geographically distinct regions (WAN)
 - *Follow the sun* workload
 - Customized Streams to convert LOBs to XMLType

Streams Replication Best Practices

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

General Configuration Tips

- Separate queue for each capture and apply, also for each source database
- init.ora:
 - `_job_queue_interval=1` (set at the source site, indicates scan rate interval (seconds) of job queue)
 - `streams_pool_size=200M` (set at source and destination, ref. to `V$STREAMS_POOL_ADVICE`)
- For WANs: (SQL*NET parameters)
 - Increase SDU (sqlnet.ora, tnsnames.ora, listener.ora)
 - Increase `send_buf_size`, `recv_buf_size`
 - For downstream capture, refer to MAA best practice paper: *Oracle Database 10g Release 2 Best Practices: Data Guard Redo Transport & Network Configuration*

10gR2 Streams Process Parameters

Capture (DBMS_CAPTURE_ADM package)

- Set retention time for capture checkpoints as needed
Alter_capture('captureName', checkpoint_retention_time=>7)
- Reduce the capture checkpoint frequency parameter
Set_parameter('captureName', '_checkpoint_frequency', '1000')

Propagation (DBMS_PROPAGATION_ADM package)

- Use queue_to_queue parameter (set to TRUE)
 - Source and target must be 10.2 or above

Apply (DBMS_APPLY_ADM package)

- Set_parameter('applyName', 'parallelism', '4')
- Set_parameter('applyName', '_hash_table_size', '10000000')
- Set_parameter('applyName', 'disable_on_error', 'N')

Apply Performance Tips

- **ALTER TABLE SYS.STREAMS\$_APPLY_PROGRESS
INITRANS 16 PCTFREE 10;**
 - **INITRANS** should match the apply parallelism
- Batch Processing
 - Frequent commits (transaction size < 1000 LCRs)
 - Consider procedural replication – replication of a PL/SQL procedure call, rather than the DML generated by the PL/SQL procedure – sample code available

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
- Use the negative ruleset (10g and above)
 - Ensure that rules do not allow objects with unsupported data types
- Avoid complex rules
 - LIKE
 - Functions
 - NOT (9.2)

Operational Tips

- Replicating DDL?
 - Avoid replicating system-generated constraint/index names
 - Modify manual hotbackup scripts to set an apply tag
 - Eg: `dbms_streams.set_tag('99')`
- Removing archived log files from disk?
 - Do not remove log files that may be needed for capture restart
 - Minimum `REQUIRED_CHECKPOINT_SCN`
 - Configure the source database to store archived redo log files in a location other than the flash recovery area – applies to downstream database also

Source Site: Heartbeat

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

Periodic Maintenance

- `DBMS_CAPTURE_ADM.BUILD()`
- `DBMS_CAPTURE_ADM.
PREPARE_GLOBAL_INSTANTIATION()`

Monitoring

- Views
- DBControl, Grid Control
- Healthcheck – MetaLink Note:273674.1 – *Configuration Report and Health Check Script*
- STRMMON – MetaLink Note:290605.1 – *Oracle Streams STRMMON Monitoring Utility*
- Alert Log – Look out for messages from the Capture process, logged when Capture sees at least 1 large or long-running transaction in its active transaction workload

Enterprise Manager

The screenshot shows the Oracle Enterprise Manager 10g Database Control interface for a database instance named 'streams'. The browser window title is 'Oracle Enterprise Manager (SYS) - Database Instance: streams - Microsoft Internet Explorer'. The interface includes a menu bar (File, Edit, View, Favorites, Tools, Help) and a toolbar with navigation icons. The main content area is titled 'Database Instance: streams' and is logged in as 'SYS'. A breadcrumb trail shows 'Home > Performance > Administration > Maintenance'. The 'Maintenance' tab is active, displaying several sections: 'High Availability' (Backup/Recovery, Backup/Recovery Settings, Oracle Backup), 'Data Movement' (Move Row Data, Move Database Files, Streams), and 'Software Deployments' (Installed Database Software, Database Software Patching). The 'Streams' link in the Data Movement section is highlighted with a yellow box. At the bottom, there are 'Related Links' for Advisor Central, Alert History, Alert Log Content, All Metrics, Blackouts, and iSQL*Plus.

Oracle Enterprise Manager 10g Database Control

Database Instance: streams

Home Performance Administration Maintenance

High Availability

- Backup/Recovery
 - Schedule Backup
 - Perform Recovery
 - Manage Current Backups
 - Manage Restore Points
 - Backup Reports
- Backup/Recovery Settings
 - Backup Settings
 - Recovery Settings
 - Recovery Catalog Settings
- Oracle Backup
 - Oracle Backup Device and Media
 - File System Backup and Restore

Data Movement

- Move Row Data
 - Export to Export Files
 - Import from Export Files
 - Import from Database
 - Load Data from User Files
 - Monitor Export and Import Jobs
- Move Database Files
 - Clone Database
 - Transport Tablespaces
- Streams
 - Setup
 - Management

Software Deployments

- Installed Database Software
 - Collection Status
- Database Software Patching
 - Apply Patch
 - View Patch Cache

Home Performance Administration Maintenance

Related Links

- Advisor Central
- All Metrics
- Alert History
- Blackouts
- Alert Log Content
- iSQL*Plus

Enterprise Manager

Oracle Enterprise Manager (SYSMAN) - Streams - Microsoft Internet Explorer

File Edit View Favorites Tools Help

ORACLE Enterprise Manager 10g
Grid Control

Setup Preferences Help Logout

Home Targets Deployments Alerts Policies Jobs Reports

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

Host: stadv08.us.oracle.com > Database Instance: inst2.net > Logged in As STRMADMIN

Streams

Overview Capture Propagation Apply Messaging

Page Refreshed May 10, 2005 1:58:02 PM PDT Refresh

View Data Manual Refresh

Capture

Capture Processes	1
Capture Processes Having Errors	0

Propagation

Propagation Jobs	1
Propagation Errors	0

Apply

Apply Processes	1
Apply Processes Having Errors	0

Messaging

Queue Tables	15
Queues	31
Total Propagation Errors	0

Overview

Oracle Streams enables information sharing. Oracle Streams can share database changes and other information in a stream, which can propagate events within a database or from one database to another. The specified information is routed to specified destinations. The result is a feature that provides greater functionality and flexibility than traditional solutions for capturing and managing information, and sharing the information with other databases and applications.

- A capture process is an Oracle background process that scans the database redo log to capture DML and DDL changes made to database objects. It formats these changes into events called logical change records (LCRs) and enqueues them into a queue.
- Propagations send events from one queue to another, and these queues can be in the same database or in different databases.
- An apply process is an Oracle background process that dequeues events from a queue and applies each event directly to a database object or sends events to apply handlers for custom processing.
- Oracle Streams Messaging, also called as Oracle Streams Advanced Queuing, provides database-integrated message queuing functionality.

Streams Summary

- Features:
 - Log-based Change Capture
 - Customizable Apply Engine
 - Schema Evolution
 - Transformations
 - Heterogeneous Support
- Versatile:
 - Replicate Data
 - Consolidate Information
 - Provide High Availability during database migration, upgrade

Streams Enhancements

- Source and Target data compare & converge
- Streams Performance Advisor
- Split/Merge of Streams for Hub & Spoke replication
 - Maintains high performance for all replicas
 - Automated, fast “catch-up” for unavailable replica
- Cross-database LCR tracking
 - Trace Streams messages from start to finish in single view
- Performance optimizations
- Streams Synchronous Capture
 - Available in all Editions of Oracle Database 11g
 - Efficient internal mechanism to immediately capture change

Resources

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



QUESTIONS
ANSWERS



For More Information

search.oracle.com



or

oracle.com

ORACLE®



ORACLE IS THE INFORMATION COMPANY