**Oracle GoldenGate Sequence Replication: ALS to ENTITYDEV**

**Specific Configuration for TINSIDCNT Sequence**

**Prerequisites**

* GoldenGate installed on both source (ALS) and target (ENTITYDEV)
* Database supplemental logging enabled
* Appropriate privileges configured
* Network connectivity established

**Step 1: Enable Sequence Supplemental Logging (Source - ALS)**

**Connect to source database (ALS) as SYSDBA**

-- Enable database-level supplemental logging

ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;

-- Enable sequence supplemental logging

ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;

-- Verify supplemental logging is enabled

SELECT supplemental\_log\_data\_min, supplemental\_log\_data\_all

FROM v$database;

-- Check current sequence status on ALS

SELECT sequence\_name, last\_number, increment\_by, cache\_size, cycle\_flag, order\_flag

FROM dba\_sequences

WHERE sequence\_name = 'TINSIDCNT';

**Step 2: Configure Extract Process (Source - ALS)**

**Create/Modify Extract Parameter File for TINSIDCNT**

-- Example: GGSCI> EDIT PARAM EXT\_TINS

EXTRACT EXT\_TINS

USERID ggadmin, PASSWORD ggadmin\_password

RMTHOST entitydev\_host, MGRPORT 7809

RMTTRAIL ./dirdat/ti

-- Enable DDL support for sequences

DDL INCLUDE MAPPED

-- Capture sequence operations specifically for TINSIDCNT

SEQUENCE \*.TINSIDCNT;

-- If you need to capture from specific schema owner

-- Replace \* with actual schema name if known

-- SEQUENCE SCHEMA\_NAME.TINSIDCNT;

-- Additional logging options

TRANLOGOPTIONS DBLOGREADER

DBOPTIONS ALLOWUNUSEDCOLUMN

-- Exclude regular table operations if only sequences needed

-- TABLEEXCLUDE \*.\*;

**Alternative Extract Configuration for Sequences Only**

EXTRACT EXT\_SEQ\_ONLY

USERID ggadmin, PASSWORD ggadmin\_password

RMTHOST exadata\_host, MGRPORT 7809

RMTTRAIL ./dirdat/sq

-- Focus only on sequences

DDL INCLUDE MAPPED OBJNAME schema\_name.\*;

SEQUENCE schema\_name.\*;

-- Exclude regular table data if only sequences needed

--TABLEEXCLUDE schema\_name.\*;

**Step 3: Configure Replicat Process (Target - ENTITYDEV)**

**Create/Modify Replicat Parameter File for TINSIDCNT**

-- Example: GGSCI> EDIT PARAM REP\_TINS

REPLICAT REP\_TINS

USERID ggadmin, PASSWORD ggadmin\_password

ASSUMETARGETDEFS

DISCARDFILE ./dirrpt/rep\_tins.dsc, PURGE

-- Handle DDL operations

DDL INCLUDE MAPPED

-- Map the TINSIDCNT sequence specifically

SEQUENCE \*.TINSIDCNT, TARGET \*.TINSIDCNT;

-- Handle potential conflicts

HANDLECOLLISIONS

DBOPTIONS SUPPRESSTRIGGERS

-- Allow sequence gaps if needed

DBOPTIONS ALLOWSEQUENCEGAPS

**Step 4: Synchronize Current TINSIDCNT Value**

**Get Current Value from Source (ALS)**

-- Connect to ALS database

SELECT sequence\_name, last\_number, cache\_size, increment\_by

FROM dba\_sequences

WHERE sequence\_name = 'TINSIDCNT';

-- Note: From your screenshot, TINSIDCNT shows:

-- Last Cached Value: 254223193

-- Cache Size: 20

-- Increment By: 1

**Set Initial Value on Target (ENTITYDEV)**

-- Connect to ENTITYDEV database

-- Check if sequence exists

SELECT sequence\_name, last\_number, cache\_size, increment\_by

FROM dba\_sequences

WHERE sequence\_name = 'TINSIDCNT';

-- If sequence doesn't exist, create it

CREATE SEQUENCE TINSIDCNT

START WITH 254223194 -- Start with next value after current

INCREMENT BY 1

MAXVALUE 9999999999

CACHE 20

NOCYCLE

NOORDER;

-- If sequence exists but value is different, sync it

-- Note: You may need to drop and recreate or use ALTER SEQUENCE

-- depending on your Oracle version

-- For Oracle 18c and later:

-- ALTER SEQUENCE TINSIDCNT RESTART START WITH 254223194;

-- For earlier versions:

DROP SEQUENCE TINSIDCNT;

CREATE SEQUENCE TINSIDCNT

START WITH 254223194

INCREMENT BY 1

MAXVALUE 9999999999

CACHE 20

NOCYCLE

NOORDER;

**Step 4: Advanced Sequence Configuration**

**For Complex Sequence Handling**

-- In Extract parameter file

EXTRACT EXT\_ADVANCED

USERID ggadmin, PASSWORD ggadmin\_password

RMTHOST exadata\_host, MGRPORT 7809

RMTTRAIL ./dirdat/ad

-- Capture sequence changes with specific options

TRANLOGOPTIONS DBLOGREADER

TRANLOGOPTIONS MINEFROMACTIVEDG

-- Sequence handling with custom logic

TABLE schema\_name.\*, TOKENS (

TK-SEQ-VAL = @GETENV('TRANSACTION', 'CSN')

);

SEQUENCE schema\_name.\*, TOKENS (

TK-SEQ-LASTVAL = @GETENV('SEQUENCE', 'LAST\_NUMBER')

);

**Step 5: Handle Sequence Synchronization Issues**

**Initial Sequence Value Synchronization Script**

-- Run on target (Exadata) to sync initial values

-- Generate sync script for all sequences

SET PAGESIZE 0

SET LINESIZE 200

SET FEEDBACK OFF

SET HEADING OFF

SPOOL sync\_sequences.sql

SELECT 'ALTER SEQUENCE ' || sequence\_owner || '.' || sequence\_name ||

' RESTART START WITH ' || last\_number || ';'

FROM dba\_sequences@source\_db\_link

WHERE sequence\_owner IN ('SCHEMA1', 'SCHEMA2') -- Adjust as needed

AND sequence\_name NOT LIKE 'SYS\_%';

SPOOL OFF

-- Execute the generated script

@sync\_sequences.sql

**Step 6: GGSCI Commands to Implement**

**On Source (M7)**

# Connect to GGSCI

ggsci

# Add extract

ADD EXTRACT EXT\_SEQ, TRANLOG, BEGIN NOW

ADD EXTTRAIL ./dirdat/lt, EXTRACT EXT\_SEQ

# Start extract

START EXTRACT EXT\_SEQ

# Check status

INFO EXTRACT EXT\_SEQ, DETAIL

**On Target (Exadata)**

# Connect to GGSCI

ggsci

# Add replicat trail

ADD REPLICAT REP\_SEQ, EXTTRAIL ./dirdat/lt

# Start replicat

START REPLICAT REP\_SEQ

# Check status

INFO REPLICAT REP\_SEQ, DETAIL

**Step 7: Monitoring and Verification**

**Check Sequence Replication Status**

-- On source

SELECT sequence\_owner, sequence\_name, last\_number, increment\_by

FROM dba\_sequences

WHERE sequence\_owner = 'YOUR\_SCHEMA'

ORDER BY sequence\_name;

-- On target

SELECT sequence\_owner, sequence\_name, last\_number, increment\_by

FROM dba\_sequences

WHERE sequence\_owner = 'YOUR\_SCHEMA'

ORDER BY sequence\_name;

**Monitor GoldenGate Lag**

# In GGSCI

LAG EXTRACT EXT\_SEQ

LAG REPLICAT REP\_SEQ

# View statistics

STATS EXTRACT EXT\_SEQ

STATS REPLICAT REP\_SEQ

**Step 8: Troubleshooting Common Issues**

**Handle Sequence Gaps**

-- In Replicat parameter file

REPLICAT REP\_SEQ

...

-- Allow sequence gaps

DBOPTIONS SUPPRESSTRIGGERS, ALLOWSEQUENCEGAPS

**Handle Sequence Cache Issues**

-- Adjust sequence cache on target if needed

ALTER SEQUENCE schema\_name.sequence\_name CACHE 1000; -- Adjust as needed

**View GoldenGate Error Logs**

# Check extract errors

VIEW REPORT EXT\_SEQ

# Check replicat errors

VIEW REPORT REP\_SEQ

# Check manager log

VIEW GGSEVT

**Important Considerations**

1. **Initial Load**: If sequences are significantly out of sync, perform initial synchronization before starting replication
2. **Cache Settings**: Consider sequence cache settings on both source and target
3. **Performance**: Sequence replication can impact performance; monitor carefully
4. **Bi-directional**: If implementing bi-directional replication, handle sequence conflicts carefully
5. **Testing**: Always test in non-production environment first

**TINSIDCNT Specific Validation Script**

-- Comprehensive comparison script for TINSIDCNT

-- Run this to validate replication is working

-- Create database link from ENTITYDEV to ALS (if not exists)

-- CREATE DATABASE LINK als\_link

-- CONNECT TO username IDENTIFIED BY password

-- USING 'ALS\_TNS\_ALIAS';

-- Compare TINSIDCNT between systems

SELECT

'TINSIDCNT Comparison' as check\_type,

als.last\_number as als\_last\_number,

entitydev.last\_number as entitydev\_last\_number,

(als.last\_number - entitydev.last\_number) as difference,

CASE

WHEN ABS(als.last\_number - entitydev.last\_number) <= 20 THEN 'OK (within cache range)'

WHEN (als.last\_number - entitydev.last\_number) > 20 THEN 'ALS ahead - check replication'

ELSE 'ENTITYDEV ahead - investigate'

END as status

FROM

(SELECT last\_number FROM dba\_sequences@als\_link WHERE sequence\_name = 'TINSIDCNT') als,

(SELECT last\_number FROM dba\_sequences WHERE sequence\_name = 'TINSIDCNT') entitydev;

-- Check sequence properties match

SELECT

'Property Check' as check\_type,

s1.increment\_by as als\_increment,

s2.increment\_by as entitydev\_increment,

s1.cache\_size as als\_cache,

s2.cache\_size as entitydev\_cache,

s1.max\_value as als\_max,

s2.max\_value as entitydev\_max,

CASE

WHEN s1.increment\_by = s2.increment\_by

AND s1.cache\_size = s2.cache\_size

AND s1.max\_value = s2.max\_value THEN 'Properties Match'

ELSE 'Properties Differ - Fix Required'

END as status

FROM

(SELECT \* FROM dba\_sequences@als\_link WHERE sequence\_name = 'TINSIDCNT') s1,

(SELECT \* FROM dba\_sequences WHERE sequence\_name = 'TINSIDCNT') s2;

**Quick Implementation Checklist for TINSIDCNT**

* [ ] **Step 1**: Enable supplemental logging on ALS
* [ ] **Step 2**: Create Extract parameter file (EXT\_TINS) on ALS
* [ ] **Step 3**: Create Replicat parameter file (REP\_TINS) on ENTITYDEV
* [ ] **Step 4**: Sync initial TINSIDCNT value (currently ~254223193)
* [ ] **Step 5**: Start Extract on ALS using GGSCI
* [ ] **Step 6**: Start Replicat on ENTITYDEV using GGSCI
* [ ] **Step 7**: Test by calling TINSIDCNT.NEXTVAL on ALS
* [ ] **Step 8**: Verify value appears on ENTITYDEV within cache tolerance
* [ ] **Step 9**: Monitor for errors in GoldenGate reports
* [ ] **Step 10**: Set up regular monitoring of sequence sync