**Oracle GoldenGate: Many Source to One Target Replication**

This document outlines the steps to configure Oracle GoldenGate to replicate data from three separate source databases on Linux to a single target database on Windows, all replicating the scott.emp table.

**1. Prerequisites and Assumptions**

Before you begin, ensure the following:

* **GoldenGate Software Installed:** Oracle GoldenGate 21c (or a compatible version) is installed on all three Linux source servers and the Windows target server at the specified GG\_HOME paths.
* **Database Connectivity:** Network connectivity exists between all source servers and the target server, allowing TNS connections.
* **Oracle Databases Running:** All source (KRUPA, suresh, mahesh) and target (proddb) Oracle databases are up and running.
* **scott Schema:** The scott schema with the emp table exists and is identical in structure on all four databases.
* **Disk Space:** Sufficient disk space is available for GoldenGate trails and checkpoints.
* **Firewall Rules:** Any firewalls are configured to allow communication on the GoldenGate Manager port (default 7809).

**2. Source Database Preparation (Linux - for each KRUPA, suresh, mahesh)**

You will repeat the following steps for each of your three source databases: KRUPA, suresh, and mahesh.

**2.1 Database Configuration**

1. **Enable Archivelog Mode:** GoldenGate requires the database to be in ARCHIVELOG mode to capture changes.
2. -- Connect as SYSDBA
3. SQL> SELECT LOG\_MODE FROM V$DATABASE;
4. -- If not ARCHIVELOG, proceed with:
5. SQL> SHUTDOWN IMMEDIATE;
6. SQL> STARTUP MOUNT;
7. SQL> ALTER DATABASE ARCHIVELOG;
8. SQL> ALTER DATABASE OPEN;
9. **Enable Supplemental Logging:** This ensures that GoldenGate captures all necessary information for replication, including primary key and unique key columns.
10. -- Connect as SYSDBA
11. SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA; -- Database-level minimal supplemental logging
12. SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY, UNIQUE INDEX) COLUMNS; -- Recommended for DML operations

**2.2 GoldenGate User Creation and Grants**

Create the GoldenGate user ogg\_linux in each source database and grant necessary privileges.

-- Connect as SYSDBA

SQL> CREATE USER ogg\_linux IDENTIFIED BY YourStrongPassword DEFAULT TABLESPACE USERS TEMPORARY TABLESPACE TEMP;

SQL> GRANT CONNECT, RESOURCE TO ogg\_linux;

SQL> GRANT SELECT ANY DICTIONARY TO ogg\_linux;

SQL> GRANT UNLIMITED TABLESPACE TO ogg\_linux;

SQL> GRANT ALTER ANY TABLE TO ogg\_linux;

SQL> GRANT FLASHBACK ANY TABLE TO ogg\_linux;

SQL> GRANT SELECT ANY TABLE TO ogg\_linux;

SQL> GRANT CREATE SESSION TO ogg\_linux;

SQL> GRANT ALTER SYSTEM TO ogg\_linux;

SQL> GRANT CREATE TABLE TO ogg\_linux;

SQL> GRANT DROP ANY TABLE TO ogg\_linux;

SQL> GRANT CREATE SEQUENCE TO ogg\_linux;

SQL> GRANT ALTER ANY SEQUENCE TO ogg\_linux;

SQL> GRANT EXECUTE ON DBMS\_LOCK TO ogg\_linux;

SQL> GRANT SELECT on V\_$ARCHIVED\_LOG to ogg\_linux; -- For integrated extract

SQL> GRANT SELECT on V\_$LOGMNR\_CONTENTS to ogg\_linux; -- For integrated extract

SQL> GRANT EXECUTE ON DBMS\_FLASHBACK TO ogg\_linux;

SQL> GRANT EXECUTE ON UTL\_FILE TO ogg\_linux; -- If using UTL\_FILE

SQL> GRANT ALTER SESSION TO ogg\_linux;

SQL> GRANT MERGE ANY VIEW TO ogg\_linux;

SQL> GRANT GLOBAL QUERY REWRITE TO ogg\_linux;

SQL> GRANT SELECT ANY TRANSACTION TO ogg\_linux;

SQL> GRANT SELECT ANY SEQUENCE TO ogg\_linux;

-- For integrated extract, also grant:

SQL> GRANT CAPTURE\_ADMIN TO ogg\_linux;

**2.3 GoldenGate Manager Process (Source - for each)**

Navigate to the GoldenGate home on Linux: /u01/app/oracle/product/gg Start the GGSCI command prompt:

cd /u01/app/oracle/product/gg

ggsci

In GGSCI:

GGSCI> EDIT PARAMS MGR

Add the following lines to mgr.prm:

PORT 7809

DYNAMICPORTLIST 7810-7820

AUTORESTART EXTRACT \*, RETRIES 5, WAITMINUTES 3, RESETMINUTES 60

PURGEOLDEXTRACTS /u01/app/oracle/product/gg/dirdat/\*, USECHECKPOINTS, MINKEEPALL 2h, MAXDAYS 5

Save and exit.

GGSCI> START MGR

GGSCI> INFO MGR

**2.4 GoldenGate Extract Process (Source - for each)**

This will be an Integrated Extract on each source.

**Source: KRUPA**

GGSCI> ADD EXTRACT EXTKRUPA, INTEGRATED TRANLOG, BEGIN NOW

GGSCI> EDIT PARAMS EXTKRUPA

Add the following lines to extkrupa.prm:

EXTRACT EXTKRUPA

SETENV (ORACLE\_SID=KRUPA)

USERID ogg\_linux, PASSWORD YourStrongPassword

DBOPTIONS DISABLETRG

TRANLOGOPTIONS DBLOGREADER

EXTTRAIL /u01/app/oracle/product/gg/dirdat/tk

TABLE SCOTT.EMP;

Save and exit.

**Source: SURESH**

GGSCI> ADD EXTRACT EXTSURESH, INTEGRATED TRANLOG, BEGIN NOW

GGSCI> EDIT PARAMS EXTSURESH

Add the following lines to extsuresh.prm:

EXTRACT EXTSURESH

SETENV (ORACLE\_SID=suresh)

USERID ogg\_linux, PASSWORD YourStrongPassword

DBOPTIONS DISABLETRG

TRANLOGOPTIONS DBLOGREADER

EXTTRAIL /u01/app/oracle/product/gg/dirdat/ts

TABLE SCOTT.EMP;

Save and exit.

**Source: MAHESH**

GGSCI> ADD EXTRACT EXTMAHESH, INTEGRATED TRANLOG, BEGIN NOW

GGSCI> EDIT PARAMS EXTMAHESH

Add the following lines to extmahesh.prm:

EXTRACT EXTMAHESH

SETENV (ORACLE\_SID=mahesh)

USERID ogg\_linux, PASSWORD YourStrongPassword

DBOPTIONS DISABLETRG

TRANLOGOPTIONS DBLOGREADER

EXTTRAIL /u01/app/oracle/product/gg/dirdat/tm

TABLE SCOTT.EMP;

Save and exit.

**2.5 GoldenGate Data Pump (Source - for each)**

Each source will have a Data Pump sending changes to the single target.

**Data Pump from KRUPA to PRODDB**

GGSCI> ADD EXTRACT PUMPKRUPA\_TO\_PRODDB, EXTTRAILSOURCE /u01/app/oracle/product/gg/dirdat/tk

GGSCI> EDIT PARAMS PUMPKRUPA\_TO\_PRODDB

Add the following lines to pumpkrupa\_to\_proddb.prm:

EXTRACT PUMPKRUPA\_TO\_PRODDB

PASSTHRU

RMTHOST <IP\_OF\_PRODDB\_SERVER>, MGRPORT 7809

RMTTRAIL C:\App\oracle\product\gg\dirdat\rp\_k

TABLE SCOTT.EMP;

Save and exit.

**Data Pump from SURESH to PRODDB**

GGSCI> ADD EXTRACT PUMPSURESH\_TO\_PRODDB, EXTTRAILSOURCE /u01/app/oracle/product/gg/dirdat/ts

GGSCI> EDIT PARAMS PUMPSURESH\_TO\_PRODDB

Add the following lines to pumpsuresh\_to\_proddb.prm:

EXTRACT PUMPSURESH\_TO\_PRODDB

PASSTHRU

RMTHOST <IP\_OF\_PRODDB\_SERVER>, MGRPORT 7809

RMTTRAIL C:\App\oracle\product\gg\dirdat\rp\_s

TABLE SCOTT.EMP;

Save and exit.

**Data Pump from MAHESH to PRODDB**

GGSCI> ADD EXTRACT PUMPMAHESH\_TO\_PRODDB, EXTTRAILSOURCE /u01/app/oracle/product/gg/dirdat/tm

GGSCI> EDIT PARAMS PUMPMAHESH\_TO\_PRODDB

Add the following lines to pumpmahesh\_to\_proddb.prm:

EXTRACT PUMPMAHESH\_TO\_PRODDB

PASSTHRU

RMTHOST <IP\_OF\_PRODDB\_SERVER>, MGRPORT 7809

RMTTRAIL C:\App\oracle\product\gg\dirdat\rp\_m

TABLE SCOTT.EMP;

Save and exit.

**2.6 Register Extracts and Add Trails (Source - for each)**

**Source: KRUPA**

GGSCI> REGISTER EXTRACT EXTKRUPA DATABASE

GGSCI> ADD EXTTRAIL /u01/app/oracle/product/gg/dirdat/tk, EXTRACT EXTKRUPA, MEGABYTES 50

GGSCI> ADD RMTTRAIL C:\App\oracle\product\gg\dirdat\rp\_k, EXTRACT PUMPKRUPA\_TO\_PRODDB, MEGABYTES 50

**Source: SURESH**

GGSCI> REGISTER EXTRACT EXTSURESH DATABASE

GGSCI> ADD EXTTRAIL /u01/app/oracle/product/gg/dirdat/ts, EXTRACT EXTSURESH, MEGABYTES 50

GGSCI> ADD RMTTRAIL C:\App\oracle\product\gg\dirdat\rp\_s, EXTRACT PUMPSURESH\_TO\_PRODDB, MEGABYTES 50

**Source: MAHESH**

GGSCI> REGISTER EXTRACT EXTMAHESH DATABASE

GGSCI> ADD EXTTRAIL /u01/app/oracle/product/gg/dirdat/tm, EXTRACT EXTMAHESH, MEGABYTES 50

GGSCI> ADD RMTTRAIL C:\App\oracle\product\gg\dirdat\rp\_m, EXTRACT PUMPMAHESH\_TO\_PRODDB, MEGABYTES 50

**3. Target Database Preparation (Windows - proddb)**

**3.1 Database Configuration**

1. **Enable Supplemental Logging (Optional, but good practice for target):** While primarily for source, if you ever plan to use this database as a source or have complex replication scenarios, it's good practice.
2. -- Connect as SYSDBA
3. SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
4. SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY, UNIQUE INDEX) COLUMNS;

**3.2 GoldenGate User Creation and Grants**

Create the GoldenGate user ogg\_windows in the target database and grant necessary privileges.

-- Connect as SYSDBA

SQL> CREATE USER ogg\_windows IDENTIFIED BY YourStrongPassword DEFAULT TABLESPACE USERS TEMPORARY TABLESPACE TEMP;

SQL> GRANT CONNECT, RESOURCE TO ogg\_windows;

SQL> GRANT UNLIMITED TABLESPACE TO ogg\_windows;

SQL> GRANT ALTER ANY TABLE TO ogg\_windows;

SQL> GRANT FLASHBACK ANY TABLE TO ogg\_windows;

SQL> GRANT SELECT ANY TABLE TO ogg\_windows;

SQL> GRANT INSERT ANY TABLE TO ogg\_windows;

SQL> GRANT UPDATE ANY TABLE TO ogg\_windows;

SQL> GRANT DELETE ANY TABLE TO ogg\_windows;

SQL> GRANT CREATE SESSION TO ogg\_windows;

SQL> GRANT ALTER SYSTEM TO ogg\_windows;

SQL> GRANT CREATE TABLE TO ogg\_windows;

SQL> GRANT DROP ANY TABLE TO ogg\_windows;

SQL> GRANT CREATE SEQUENCE TO ogg\_windows;

SQL> GRANT ALTER ANY SEQUENCE TO ogg\_windows;

SQL> GRANT EXECUTE ON DBMS\_LOCK TO ogg\_windows;

SQL> GRANT MERGE ANY VIEW TO ogg\_windows;

SQL> GRANT SELECT ANY TRANSACTION TO ogg\_windows;

-- For integrated replicat, also grant:

SQL> GRANT IMP\_FULL\_DATABASE TO ogg\_windows; -- For initial load using GoldenGate

SQL> GRANT GG\_ADMIN TO ogg\_windows; -- If using GG 21c and later

**3.3 GoldenGate Manager Process (Target)**

Navigate to the GoldenGate home on Windows: C:\App\oracle\product\gg Start the GGSCI command prompt:

cd C:\App\oracle\product\gg

ggsci

In GGSCI:

GGSCI> EDIT PARAMS MGR

Add the following lines to mgr.prm:

PORT 7809

DYNAMICPORTLIST 7810-7820

AUTORESTART REPLICAT \*, RETRIES 5, WAITMINUTES 3, RESETMINUTES 60

PURGEOLDEXTRACTS C:\App\oracle\product\gg\dirdat\\*, USECHECKPOINTS, MINKEEPALL 2h, MAXDAYS 5

Save and exit.

GGSCI> START MGR

GGSCI> INFO MGR

**3.4 GoldenGate Replicat Processes (Target)**

You will need a separate Replicat process for each source database to consume its respective remote trail.

**Replicat for KRUPA to PRODDB**

GGSCI> ADD REPLICAT REP\_KRUPA\_TO\_PRODDB, INTEGRATED, EXTTRAIL C:\App\oracle\product\gg\dirdat\rp\_k

GGSCI> EDIT PARAMS REP\_KRUPA\_TO\_PRODDB

Add the following lines to rep\_krupa\_to\_proddb.prm:

REPLICAT REP\_KRUPA\_TO\_PRODDB

SETENV (ORACLE\_SID=proddb)

USERID ogg\_windows, PASSWORD YourStrongPassword

ASSUMETARGETDEFS

MAP SCOTT.EMP, TARGET SCOTT.EMP;

Save and exit.

**Replicat for SURESH to PRODDB**

GGSCI> ADD REPLICAT REP\_SURESH\_TO\_PRODDB, INTEGRATED, EXTTRAIL C:\App\oracle\product\gg\dirdat\rp\_s

GGSCI> EDIT PARAMS REP\_SURESH\_TO\_PRODDB

Add the following lines to rep\_suresh\_to\_proddb.prm:

REPLICAT REP\_SURESH\_TO\_PRODDB

SETENV (ORACLE\_SID=proddb)

USERID ogg\_windows, PASSWORD YourStrongPassword

ASSUMETARGETDEFS

MAP SCOTT.EMP, TARGET SCOTT.EMP;

Save and exit.

**Replicat for MAHESH to PRODDB**

GGSCI> ADD REPLICAT REP\_MAHESH\_TO\_PRODDB, INTEGRATED, EXTTRAIL C:\App\oracle\product\gg\dirdat\rp\_m

GGSCI> EDIT PARAMS REP\_MAHESH\_TO\_PRODDB

Add the following lines to rep\_mahesh\_to\_proddb.prm:

REPLICAT REP\_MAHESH\_TO\_PRODDB

SETENV (ORACLE\_SID=proddb)

USERID ogg\_windows, PASSWORD YourStrongPassword

ASSUMETARGETDEFS

MAP SCOTT.EMP, TARGET SCOTT.EMP;

Save and exit.

**4. Initial Load Strategy**

Before starting the change synchronization, you need to load the existing data from scott.emp on each of your source databases (KRUPA, suresh, mahesh) to scott.emp on proddb.

A common and robust approach is to export scott.emp from *each* source and import it into the single target. **Crucially, you must perform the initial load carefully to avoid conflicts if the same primary key exists in multiple sources.** If the data is truly disjoint (e.g., each source has unique primary keys), then a direct import is fine. If not, you might need data transformation or a merged initial load. For simplicity, we assume disjoint data or that TABLE\_EXISTS\_ACTION=APPEND is acceptable and handles duplicates (which might result in constraint errors if primary keys collide).

Here, we'll outline a simple expdp/impdp method for each source.

1. **Export scott.emp from each Source (Linux Command Prompt):**
   * **On KRUPA server:**
   * expdp ogg\_linux/YourStrongPassword@KRUPA DUMPFILE=emp\_krupa.dmp LOGFILE=emp\_krupa\_exp.log TABLES=scott.emp
   * **On SURESH server:**
   * expdp ogg\_linux/YourStrongPassword@suresh DUMPFILE=emp\_suresh.dmp LOGFILE=emp\_suresh\_exp.log TABLES=scott.emp
   * **On MAHESH server:**
   * expdp ogg\_linux/YourStrongPassword@mahesh DUMPFILE=emp\_mahesh.dmp LOGFILE=emp\_mahesh\_exp.log TABLES=scott.emp
2. These will create dump files in your DATA\_PUMP\_DIR on each source. Copy these three dump files (emp\_krupa.dmp, emp\_suresh.dmp, emp\_mahesh.dmp) to the Windows target server into a directory accessible by Oracle (e.g., C:\temp\data\_pump\_dir).
3. **Import scott.emp to Target (proddb - Windows Command Prompt):**
   * First, consider truncating the target table if it already contains data from previous tests or if you need a clean sync:
   * -- Connect as SCOTT or OGG\_WINDOWS with appropriate grants
   * SQL> TRUNCATE TABLE scott.emp;
   * **Import from KRUPA dump:**
   * impdp ogg\_windows/YourStrongPassword@proddb DUMPFILE=emp\_krupa.dmp LOGFILE=emp\_krupa\_imp.log TABLES=scott.emp TABLE\_EXISTS\_ACTION=APPEND
   * **Import from SURESH dump:**
   * impdp ogg\_windows/YourStrongPassword@proddb DUMPFILE=emp\_suresh.dmp LOGFILE=emp\_suresh\_imp.log TABLES=scott.emp TABLE\_EXISTS\_ACTION=APPEND
   * **Import from MAHESH dump:**
   * impdp ogg\_windows/YourStrongPassword@proddb DUMPFILE=emp\_mahesh.dmp LOGFILE=emp\_mahesh\_imp.log TABLES=scott.emp TABLE\_EXISTS\_ACTION=APPEND
4. Use TABLE\_EXISTS\_ACTION=APPEND to add data from each dump file. If primary key collisions are expected, this will result in errors for duplicate rows. A more advanced initial load might involve a custom GoldenGate initial load (using TABLE and RULENAME in parameter files, or initial load via file) or data merging logic.

**5. Starting GoldenGate Processes**

Once the initial load is complete and verified, you can start the GoldenGate processes.

**5.1 On each Source (Linux)**

**For KRUPA:**

GGSCI> START EXTKRUPA

GGSCI> START PUMPKRUPA\_TO\_PRODDB

GGSCI> INFO ALL

**For SURESH:**

GGSCI> START EXTSURESH

GGSCI> START PUMPSURESH\_TO\_PRODDB

GGSCI> INFO ALL

**For MAHESH:**

GGSCI> START EXTMAHESH

GGSCI> START PUMPMAHESH\_TO\_PRODDB

GGSCI> INFO ALL

**5.2 On Target (Windows)**

GGSCI> START REP\_KRUPA\_TO\_PRODDB

GGSCI> START REP\_SURESH\_TO\_PRODDB

GGSCI> START REP\_MAHESH\_TO\_PRODDB

GGSCI> INFO ALL

**6. Verification and Monitoring**

* **Check Status:** Use INFO ALL or STATUS <process\_name> in GGSCI to check the status of your processes on both source and target.
* **View Reports:** Use VIEW REPORT <process\_name> to check process logs for errors or important messages.
* **Lag Reports:** SEND EXTRACT <extract\_name>, REPORT and SEND REPLICAT <replicat\_name>, REPORT for detailed lag information.
* **Insert/Update/Delete on Sources:** Perform some DML operations on scott.emp on KRUPA, suresh, and mahesh databases and verify that the changes appear on the proddb database.

**Important Notes:**

* **Passwords:** Replace YourStrongPassword with actual strong passwords.
* **IP Addresses:** Replace <IP\_OF\_PRODDB\_SERVER> with the actual IP address or hostname of your Windows target server.
* **Error Handling:** Monitor GoldenGate report files and database alert logs for any errors.
* **Conflict Resolution:** For many-to-one replication, especially if the same rows (based on primary key) can be modified on multiple sources, you **must** implement conflict detection and resolution (CDR) rules in your Replicat parameter files. This guide does not include CDR, but it's critical for true multi-source environments. Examples include HANDLECOLLISIONS, DEDUP\_KEY\_COLUMNS, RESOLVECONFLICT parameters.
* **Security:** In a production environment, GoldenGate best practices recommend hardening security (e.g., network encryption, secure parameter storage).
* **Integrated Processes:** This configuration uses Integrated Extract and Integrated Replicat, which are recommended for modern Oracle databases.
* **Character Sets:** Ensure character sets are compatible between source and target databases to avoid data corruption.
* **Underscores in SETENV:** For ORACLE\_SID in SETENV, use underscores.
* **Case Sensitivity:** Be mindful of case sensitivity for paths and database names on Linux and Windows.
* **Directory Permissions:** Ensure the ogg\_linux OS user has read/write permissions to /u01/app/oracle/product/gg/dirdat on the source servers, and ogg\_windows OS user on the target.