# Configuring Change Synchronization ( Part I )

## Practice Overview

In this practice you will implement the whole change synchronization cycle. You will configure a primary Extract process, a Data Pump process and a Replicat process that will keep the databases in sync after the initial load is finished. Needless to say, this is the core functionality of Oracle GoldenGate.

The elements of configuring change synchronization are basically simple: **initial load** which runs for once, **change synchronization** which is an ongoing process that keeps the source and the target databases in sync.

However, the timing of starting each process could be challenging because you want to make sure that the data that has been loaded by the initial load is not applied again by the change synchronization. Usually, there are three ways to handle this challenge:

- **Option 1**: this procedure is safe and simple but requires application downtime.
  - Stop the changes on the source database (stop the applications)
  - Start the initial load
  - After the initial load finishes, start the Extract and the Replicat
  - Allow changes on the source database to resume

- **Option 2**: this procedure is safe but a little bit more complicated. It does not require application downtime though.
  - Start the Extract
  - Start the initial load
  - When the initial load finishes, take a note of the CSN at which the Replicat finishes applying the initial load
  - Start the Replicat at the CSN point that is after the CSN noted in the previous step

- **Option 3**: this procedure has some risk of data loss or conflict. It does not require downtime.
  - Start the Extract
  - Start the initial load
  - Wait till the initial load finishes
  - Start the Replicat after configuring it to ignore the conflicts
  - Wait till the conflicts disappear
  - Re-start the Replicat after disabling conflict-ignore option

## Practice Environment

This practice assumes that you have implemented the previous practice. Therefore, the Direct-load initial load Extract and Replicat tasks 1are already implemented on the VirtualBox appliances.

**You will implement Option 1 in Part 1 of this practice and Option 2 in part 2.**

To make implementing the course practices easier, I recommend opening two Putty sessions to each system. One for ggsci and OS commands, and the other for SQL*Plus statements.

## Implementing Change Synchronization Option 1

In the following procedure, you will configure change synchronization by making the source dataset read-only, executing the initial load, running the Extract and Replicat, and setting the source dataset to read-write status again.

### A. Prepare the target dataset

**1.** Make sure the databases are up, running and accessible through the listener

```
sqlplus system/oracle@db1
sqlplus system/oracle@db2
```

**2.** Truncate the tables in the HRTRG schema in the target database.

```
TRUNCATE TABLE HRTRG.JOB_HISTORY;
TRUNCATE TABLE HRTRG.EMPLOYEES;
TRUNCATE TABLE HRTRG.JOBS;
TRUNCATE TABLE HRTRG.DEPARTMENTS;
TRUNCATE TABLE HRTRG.LOCATIONS;
TRUNCATE TABLE HRTRG.REGIONS;
```

**3.** Make sure the constraints and the triggers in the HRTRG schema are disabled and the indexes were dropped. You have done so in the previous practice.

```
SELECT TABLE_NAME||' - '||CONSTRAINT_NAME || '-' || STATUS
FROM DBA_CONSTRAINTS
WHERE STATUS <> 'DISABLED' AND OWNER= 'HRTRG'
  AND TABLE_NAME IN ('JOB_HISTORY','EMPLOYEES','JOBS',
'DEPARTMENTS','LOCATIONS','REGIONS')
ORDER BY TABLE_NAME, CONSTRAINT_NAME;

SELECT TABLE_NAME||' - '|| TRIGGER_NAME || '-' || STATUS
FROM DBA_TRIGGERS
WHERE STATUS <> 'DISABLED' AND OWNER= 'HRTRG'
  AND TABLE_NAME IN ('JOB_HISTORY','EMPLOYEES','JOBS',
'DEPARTMENTS','LOCATIONS','REGIONS')
ORDER BY TABLE_NAME, TRIGGER_NAME;

SELECT INDEX_NAME FROM DBA_INDEXES
 WHERE OWNER = 'HRTRG' AND TABLE_NAME IN (
'JOB_HISTORY','EMPLOYEES','JOBS','DEPARTMENTS','LOCATIONS','REGIONS');
```

## B. Stop the change operations on the source dataset

**4.** Make the source tables read only.

You can use any method to stop the changes on the source database. You may start up the source database in READONLY mode, stop the listener, or simply stop the applications that get access to the database providing no other entity is accessing the source data.

```
Alter Table HR.JOB_HISTORY READ ONLY;
Alter Table HR.EMPLOYEES READ ONLY;
Alter Table HR.JOBS READ ONLY;
Alter Table HR.DEPARTMENTS READ ONLY;
Alter Table HR.LOCATIONS READ ONLY;
Alter Table HR.REGIONS READ ONLY;
```

## C. Setup the Extract and the trails

In the following steps, you will set up the primary Extract process and its trail files.

**5.** Create the Extract parameter file

Make sure the USERID parameter comes before the parameter `EXTTRAIL`, as shown in the code below.

You typically add more parameters in an Extract parameter file that the parameters shown in the code below. You will learn about more Extract parameters as you progress in the course.

“`edit param`” command is equivalent to “`edit params`” command.

```
ggsci> edit param esrv1
```

```
Extract esrv1
USERID ogg, PASSWORD oracle
ExtTrail ./dirdat/es
Table HR.JOB_HISTORY;
Table HR.EMPLOYEES;
Table HR.JOBS;
Table HR.DEPARTMENTS;
Table HR.LOCATIONS;
Table HR.REGIONS;
```

**6.** Add the Extract process.

```
ggsci> Add Extract esrv1, TranLog, Begin Now
```

**7.** Add a local extract trail that links the trail to the `esrv1` Extract.

```
ggsci> Add ExtTrail ./dirdat/es, Extract esrv1
```

## D. Setup the Data Pump process

In the following steps, you will set up the Data Pump Extract process

**8.** Create the Data Pump Extract parameter file

```
ggsci> edit param psrv1
```

```
Extract psrv1
RmtHost ggsrv2, MgrPort 7810
RmtTrail ./dirdat/rt
Passthru
Table HR.*;
```

**9.** Add the Data Pump Extract group

```
Add Extract psrv1, ExtTrailSource ./dirdat/es
Add RmtTrail ./dirdat/rt, Extract psrv1
```

## E. Setup the Checkpoint table and the Replicat process

In the following steps, you will set up the checkpoint table and the Replicat process in the target system.

**10.** On the target system, add the checkpoint table.

```
ggsci> DBLogin UserID ogg@db2, Password oracle
ggsci> Add CheckpointTable OGG.GG_CHECKPOINT
ggsci> Info CheckpointTable OGG.GG_CHECKPOINT
ggsci> List Tables OGG.GG*
```

**11.** Create the Replicat parameter file and add the parameters to it as shown below.

```
ggsci> edit param rsrv2
```

```
Replicat rsrv2
DiscardFile ./dirrpt/rsrv2.dsc, Purge
USERID ogg, PASSWORD oracle
Map HR.*, Target HRTRG.*;
```

**12.** Add the Replicat process group and link it to the trails generated by the Data Pump process.

```
ggsci> Add Replicat rsrv2, ExtTrail ./dirdat/rt, CHECKPOINTTABLE OGG.GG_CHECKPOINT
```

## F. Run the initial load

Instantiate the target dataset with the method of your choice. In this practice, you will use the Direct Load method.

**13.** In the source and target systems, startup the Manager process, if it is down.

```
cd $GG_HOME
ggsci
ggsci> info mgr
ggsci> start mgr
ggsci> info mgr
```

**14.** In the source database, verify that the initial load Extract task parameter file (`edlinit`) is configured.

```
ggsci> view param edlinit
```

**15.** In the target database, verify that the initial load Replicat task parameter file (`rdlinit`) is configured.

```
ggsci> view param rdlinit
```

**16.** In the source system, start the initial load Extract task.

```
ggsci> start extract edlinit
```

**17.** Keep monitoring the status of the task till it finishes (its status becomes STOPPED when it finishes).

```
info extract edlinit
```

**18.** On the target system, view the initial load Replicat task report. Make sure no error was reported.

```
ggsci> view report rdlinit
```

**19.** Verify that the initial load was successful.

Do not go to the next step, until the initial load successfully finished. If you have to re-perform the initial load, truncate the source tables and restart the initial load Extract.

```
sqlplus system/oracle@db1
SELECT COUNT(*) FROM HR.EMPLOYEES;

conn system/oracle@db2
SELECT COUNT(*) FROM HRTRG.EMPLOYEES;
```

**20.** In `db2`, create the indexes of the target dataset.

The script below was created in the previous practice.

```
sqlplus / as sysdba
@/home/oracle/scripts/hrtrg_cindexs.sql
```

**21.** Enable the HRTRG constraints that you had disabled before you started the initial load.

```
SET SERVEROUTPUT ON
exec DBMS_OUTPUT.ENABLE(1000)
DECLARE
 V_SCHEMA VARCHAR2(10) := 'HRTRG';
 V_SQL VARCHAR2(32000);
 CURSOR c_DisableConstraints IS
 SELECT 'alter table '|| OWNER|| '.' ||TABLE_NAME||' enable constraint
'||CONSTRAINT_NAME sqlstatement, TABLE_NAME, CONSTRAINT_NAME
 FROM DBA_CONSTRAINTS WHERE OWNER= V_SCHEMA
 AND TABLE_NAME IN (
'JOB_HISTORY','EMPLOYEES','JOBS','DEPARTMENTS','LOCATIONS','REGIONS')
 AND STATUS='DISABLED'
 ORDER BY CONSTRAINT_TYPE;
BEGIN
 FOR r IN c_DisableConstraints LOOP
 BEGIN
  V_SQL := r.sqlstatement;
  EXECUTE IMMEDIATE (V_SQL);
  --DBMS_OUTPUT.PUT_LINE(R.TABLE_NAME ||'-' || R.CONSTRAINT_NAME);
 EXCEPTION
  WHEN OTHERS THEN
   DBMS_OUTPUT.PUT_LINE('FAILED: ' || V_SQL);
 END ;
 END LOOP;
END;
/
```

## G. Start up the change synchronization and verify its operation

**22.** Start the Extract process, and then verify the results.

If any of the processes failed to start, check out the `ggserr.log` file

A quicker way to start all the processes at once is the command “`start *`”.

```
ggsci> start extract esrv1
ggsci> info extract esrv1
ggsci> start extract psrv1
ggsci> info ER *
ggsci> info all
```

**23.** Look in the trail directory and verify that local trails are created.

```
ggsci> sh ls dirdat
```

**24.** In the target system, start the Replicat process, and then check its status.

```
ggsci> start replicat rsrv2
ggsci> info replicat rsrv2
ggsci> sh ls ./dirdat
```

**25.** Make the source tables read write. This imitates making the source database available for read/write operations.

```
sqlplus system/oracle@db1
Alter Table HR.JOB_HISTORY READ WRITE;
Alter Table HR.EMPLOYEES READ WRITE;
Alter Table HR.JOBS READ WRITE;
Alter Table HR.DEPARTMENTS READ WRITE;
Alter Table HR.LOCATIONS READ WRITE;
Alter Table HR.REGIONS READ WRITE;
```

**26.** Perform basic verification on the configuration.

You will learn in a separate lecture more details on how to monitor the GoldenGate operation.

```
-- on db1
sqlplus system/oracle@db1

SELECT FIRST_NAME FROM HR.EMPLOYEES WHERE EMPLOYEE_ID=100;
UPDATE HR.EMPLOYEES SET FIRST_NAME= 'John' WHERE EMPLOYEE_ID=100;
COMMIT;

-- on db2
sqlplus system/oracle@db2
SELECT FIRST_NAME FROM HRTRG.EMPLOYEES WHERE EMPLOYEE_ID=100;
```

## Summary

In this practice you have created Extract, Data Pump, and Replicat processes to implement the online change synchronization from the source database to the target database. You have stopped the changes on the source database during the initial load.

In part 2 of this practice lecture, you will implement the change synchronization without having to disable the data changes on the source database.