# Mail Online GoldenGate Replication

This document details the steps that are needed in order to get uni-directional replication working between 2 Oracle Databases.

Source Host and Database: and-hsk-rac10inta1 / MOLINTA (molint1/2)  
 Oracle Database 10.2.0.5

Target Host and Database: mol-dbtest1 / molintpc (molintpc1/2)  
 Oracle Database 12.1.0.2

Note: All of the GoldenGate processes are running from node 1 on source and target.

# Software Installation

Source: GoldenGate 11.2.0.1.20 (Patch [18377868](https://support.oracle.com/epmos/faces/PatchDetail?requestId=17473095&_afrLoop=273529212719555&patchId=18377868&_afrWindowMode=0&_adf.ctrl-state=z503wd6xk_457) for compatibility with 10gR2)  
 Media Location: /u01/source/goldengate\_11.2  
 Installation: /u05/orafra/goldengate

Target: GoldenGate 12.2.0.1.1  
 Media Location: /u01/source/goldengate\_12.2  
 Installation: /export/home/andracintbackup/goldengate/12.2/db\_1

# Configuration Files

For both the 10g and 12c servers, the configuration files are stored in the following locations

Parameters: $GGHOME/dirprm  
Extract/Replicat Scripts: $GGHOME/diroby  
Transdata Scripts: $GGHOME/diroby  
Macros: $GGHOME/dirprm/macros  
Testing Scripts: $GGHOME/dirsql  
Sequence Scripts: $GGHOME/dirsql

# Initial Setup

Create or amend environment variables on the hosts  
export GGHOME=/u05/orafra/goldengate (For 10g Host)  
export GGHOME=/export/home/andracintbackup/goldengate/12.2/db\_1 (For 12c Host)  
export LD\_LIBRARY\_PATH=$GGHOME:$LD\_LIBRARY\_PATH:$ORACLE\_HOME/lib

Sub directories need to be created. These will hold the trail, parameter and report files.  
GGSCI> CREATE SUBDIRS

Supplemental Logging must be enabled at the database level where an extract is occurring  
SQL> alter database add supplemental log data;

To support DDL replication the Recycle Bin must be turned off on all 10gR2 databases  
SQL> alter system set recyclebin=off scope=both;

On the 12c Database the following initialisation parameter must be set  
SQL> alter system set enable\_goldengate\_replication=true;

On both databases create a GoldenGate administration user  
SQL> create tablespace goldengate datafile size 1000M autoextend on next 100M;  
SQL> create user ggate identified by ggate123 default tablespace goldengate;

For the user on 12c you can assign the necessary administrative rights with the following command  
SQL> exec dbms\_goldengate\_auth.grant\_admin\_privilege('ggate');

For the user on 10g you will need to run the following procedure instead   
SQL> exec dbms\_streams\_auth.grant\_admin\_privilege('GGATE');

You will also need to manually assign the following grants  
SQL> grant CONNECT to ggate;  
grant CREATE SESSION to ggate;  
grant RESOURCE to ggate;  
grant SELECT ANY TABLE to ggate;  
grant ALTER SESSION to ggate;  
grant CREATE TABLE to ggate;  
grant FLASHBACK ANY TABLE to ggate;  
grant SELECT ANY DICTIONARY to ggate;  
grant execute on dbms\_flashback to ggate;  
grant execute on utl\_file to ggate;

DDL Setup  
Run the following scripts on the Source Host, they can be found in the GoldenGate home directory.  
You will be prompted for the username that was created above.  
$ sqlplus / as sysdba  
SQL> @marker\_setup.sql  
SQL> @ddl\_setup.sql  
SQL> @role\_setup.sql  
SQL> GRANT GGS\_GGSUSER\_ROLE TO ggate;  
SQL> @ddl\_enable  
SQL> @ddl\_pin ggate

## Supplemental Log Groups

The recommended way of creating these is to login to GGSCI and run the “add trandata” command.  
However it appears that this command locks the table before creating the log group.

See below article from Julian Dyke for more information (Table Level Supplemental Logging) …  
<http://www.juliandyke.com/Research/GoldenGate/GoldenGateSupplementalLogging.php>  
Therefore I decided not to run this against the production database.

Instead I used Supplemental Data Logging …

SQL> ALTER TABLE MAILONLINE.ARTICLES ADD SUPPLEMENTAL LOG DATA (ALL) COLUMNS;  
SQL> ALTER TABLE MAILONLINE.CHANNELS ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;

For tables involved in Conflict Detection and Resolution (CDR) log all columns.  
For all other tables only log Primary Keys.

The script for “all columns” can be found in $GGHOME/dirsql/ log\_groups\_all.sql  
The create script for “primary keys” can be found in $ GGHOME/dirsql/ log\_groups\_pk.sql

NOTE: The command for “schematrandata” cannot be used with 10gR2 without a patch being applied.

Create Checkpoint Table  
It is recommended practice to store GoldenGate checkpoints in a table rather than the trail file.  
Only replication processes use checkpoints so this only needs to be done on the target host.

GGSCI> EDIT PARAMS ./GLOBALS

GGSCHEMA GGATE  
CHECKPOINTTABLE GGATE.CHKPTAB

GGSCI> DBLOGIN USERID ggate, PASSWORD ggate123  
GGSCI> ADD CHECKPOINTTABLE GGATE.CHKPTAB

# Manager Setup

GGSCI> edit param mgr

GGSCI> START MANAGER

Both MGR parameter files should be very similar. They should all have PORT, DYNAMICPORTLIST, AUTOSTART, AUTORESTART and PURGEOLDEXTRACTS parameters.

The 12c parameter file must also have the following parameter in order to allow incoming traffic …

ACCESSRULE, PROG \*, IPADDR 10.251.64.35, ALLOW

# Extract Setup

The extract processes are created on the source database. Many different processes will need to be created to keep up with the changes to the source database. However it is important to note that tables with foreign key dependencies must be kept within the same extract.

For the proof of concept there are 4 separate extract processes. MAILONLINE, RDRCOMMENTS, REGISTRATION and one for all of the other schemas.  
For the examples below I will just detail the MAILONLINE schema.

GGSCI> edit param EXTMOL01

The below can also be done via an obey file in $GGHOME/diroby

GGSCI> OBEY /u07/goldengate/diroby/extmol01.oby

GGSCI> DBLOGIN USERID ggate, PASSWORD ggate123  
GGSCI> ADD EXTRACT EXTMOL01, TRANLOG, THREADS 2, BEGIN NOW  
GGSCI> ADD EXTTRAIL ./dirdat/ml, EXTRACT EXTMOL01, MEGABYTES 100  
GGSCI> START EXTRACT EXTMOL01

GGSCI> INFO ALL

If status shows as stopped or abended then either view the “ggserr.log” file in the GoldenGate home directory or type “VIEW REPORT EXTMOL01” from within ggsci.

# Data Pump Setup

The GoldenGate data pump extract will move the local trail files from the source to the target host.

GGSCI> edit param DPMOL01

GGSCI> DBLOGIN USERID ggate, PASSWORD ggate123  
GGSCI> ADD EXTRACT DPMOL01, EXTTRAILSOURCE, ./dirdat/source/ml  
GGSCI> ADD RMTTRAIL ./dirdat/target/ml, EXTRACT DPMOL01  
GGSCI> START EXTRACT DPMOL01

GGSCI> INFO ALL

Once again check ggserr.log or “VIEW REPORT DPMOL01”

## Compression and TCP Buffer Size

An important consideration for the Pump process is whether or not to compress the trail files before they are copied to the remote host. The compress option can result in compression ratios of 4:1 but does cause extra CPU usage.

Increasing the size of the TCP socket buffer size can also help copying trail files.  
On Solaris check the following to see what the configured maximum TCP buffer sizes are …

ndd -get /dev/tcp tcp\_recv\_hiwat  
ndd -get /dev/tcp tcp\_xmit\_hiwat

The default value for Solaris is 49512. Check to see if these values can be increased. There is no point setting the TCPBUFSIZE parameter to be greater than these values.

To change the values ask the System Administrator to do the following …

ndd -set /dev/tcp tcp\_recv\_hiwat 262144  
ndd -set /dev/tcp tcp\_xmit\_hiwat 262144

Also add details to /etc/rc2.d/S99ndd so that the values persist after a reboot.

NOTE: For the 10g GoldenGate setup, the TCP Buffer Size was not increased. No excess lag was noted during the day, so this step is probably not needed.

# Initial Load

Once the above has been completed you can now begin work on the initial load of the data.

If the migration is going to the same version of the database then it is better to use Data Guard and perform a failover. You can then extract the exact SCN that you need using the following SQL

SQL> SELECT STANDBY\_BECAME\_PRIMARY\_SCN FROM V$DATABASE;

There is a separate document that details how the initial load was done. This is called “GoldenGate Database Initial Setup”. It documents how to go from 10g to 12c.

A version of this document has been attached to this document below, although where possible use the latest document at www.dubetech.co.uk …



# Replication Setup

First check that the Data Pump extract process on the source is sending trail files to the target host.  
This can be checked by making sure there are files in the $GGHOME/dirdat directory. Without these files there will be nothing to replicate on the target database.

All below commands are done on the target host

GGSCI> EDIT PARAMS REPMOL01

GGSCI> DBLOGIN USERID ggate, PASSWORD ggate123  
GGSCI> ADD REPLICAT REPMOL01, CHECKPOINTTABLE GGATE.CHKPTAB, EXTTRAIL dirdat/ml

If this is the first time we are starting the replication then we will need to include the “atcsn” parameter. This means when the replicat process reads the trail file, it will start at the SCN given in the “start replicat” command.

CSN stands for Commit Sequence Number. It uniquely identifies a point in time when a transaction is committed. For Oracle the System Change Number (SCN) will be the same as the CSN.  
   
GGSCI> START REPLICAT REPMOL01 atcsn 867420988

Or if you are just restarting the replicat process …

GGSCI> STARTREPLICAT REPMOL01  
GGSCI> STATUS ALL

Once again check ggserr.log or “VIEW REPORT DPMOL01”

On occasions the replicat process does not read the correct trail file. It can be manually pointed at the correct file by using the following command, whilst the replicat process is not running.

GGSCI> ALTER REPLICAT REPMOL01, EXTSEQNO <TRAILFILE\_NO>, EXTRBA <TRAILFILE\_POSITION>

If all is well you should now be able to perform DML and DDL statements on the source database and they will magically appear on the target database.

# Monitoring

To view details of all GoldenGate processes  
GGSCI> INFO ALL

To view information on a particular GoldenGate process  
GGSCI> INFO EXTRACT EXTREG01  
GGSCI> INFO EXTRACT EXTREG01, DETAIL

To view statistics of a particular GoldenGate process  
GGSCI> STATS EXTRACT EXTREG01  
GGSCI> STATS EXTRACT EXTREG01 REPORTRATE MIN  
GGSCI> STATS REPLICAT REPMOL01 TOTALSONLY MAILONLINE.\*  
GGSCI> STATS EXTRACT EXTREG01 TOTALSONLY REGISTRATION .\*, REPORTRATE MIN

To view the latency between the source and target databases  
GGSCI> LAG EXTRACT EXTMOL  
GGSCI> LAG EXTRACT EXT\*

To examine the current Checkpoints on the target database, use the following SQL …

set linesize 100  
col group\_name format a12  
col scn format a16  
select GROUP\_NAME, LOG\_CSN as SCN, SEQNO, RBA, LAST\_UPDATE\_TS  
from GGATE.CHKPTAB;

For the current Mail Online setup it should return 4 rows detailing the replication processes for MAILONLINE, RDRCOMMENTS, REGISTRATION and a final one covering all other schemas.

## Cloud Control

When extract are running you will see that “streams miscellaneous event” becomes one of the top wait events. According to document 1317122.1 this is due to a bug with GoldenGate 11g. This wait event is not seen when using GoldenGate 12c.

As there is no way of fixing this issue, this wait event can be ignored.

# Errors

## Delete Cascade

This occurs when GoldenGate sends across the SQL for the delete cascade part of the operation.  
This causes the delete statement to be executed twice on the target and the 2nd time GoldenGate errors with ORA-01403. The transaction then rolls back and the record remains in the table.

There are three ways to get around this issue.

1. DBOPTIONS DEFERREFCONST  
   This will delay checking and enforcement of integrity constraints until the Replicat transaction commits.
2. MAP MAILONLINE.USER\_TRANSFER\_ITEM, TARGET MAILONLINE.USER\_TRANSFER\_ITEM, REPERROR (1403, IGNORE);  
   This entry will ignore all ORA-01403 errors on the table in question. However this will also ignore any update errors, so only use if you know there are no updates on the table.
3. MAP MAILONLINE.USER\_TRANSFER\_ITEM, TARGET MAILONLINE.USER\_TRANSFER\_ITEM,  
    COMPARECOLS (ON DELETE KEY),  
    RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, DISCARD)),  
    RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));  
   This uses CDR and specifies if the key column for the table doesn’t exist then by default we will discard the error and carry on.

## Materialised View

Problems can occur when GoldenGate attempts to run the SQL from a Materialised View refresh on the target database. The following error message may be seen …

ORA-01732: data manipulation operation not legal on this view

If the refresh is also going to run on the target database then there is no point in replicating the materialised view. Therefore you can add the following entry in the parameter file

MAPEXCLUDE MAILONLINE.MV\_LATEST\_ARTICLE\_AUTHOR;

It is important to place this entry before the MAP schema wildcard entry.

## Primary Key

A warning will be seen in the error log file whenever GoldenGate has to perform a database operation on a table without a primary key. In these cases it will have to log every column, which can mean very large trail files and more memory used.

It is possible to tell GoldenGate what the primary key columns are without creating a primary key on the database table. This can be achieved using the KEYCOLS parameter.

MAP MAILONLINE.ASSET\_PUBLISHER\_LAST\_RUN, TARGET MAILONLINE.ASSET\_PUBLISHER\_LAST\_RUN, KEYCOLS(ITEM\_TYPE);

## Skipping Errors

Where possible it is preferable to enter rules to skip errors rather than manually skipping the error every time. But for one off errors it may be simpler to instruct the replicat process to skip the transaction.

It is important to remember that this will skip the whole transaction. Not necessarily just the problematic SQL statement.

GGSCI> START REPLICAT EXTMOL01 SKIPTRANSACTION

## All Other Errors

For specific “OGG” error messages please look at the following Wiki page …

<http://dubetech.co.uk/doku.php?id=error_messages>