

Oracle GoldenGate - Basic Configuration

The following procedure describes a minimal Oracle Golden Gate configuration. It is intended for research and/or training purposes and is not intended to be a production configuration. Once the basic configuration has been tested, new features can be added incrementally simplifying any troubleshooting.

For a basic configuration I used two Linux VMs (OEL5U6) running single instance Oracle 11.2.0.3 databases. I created both databases using DBCA.

In the following configuration I have used the following hosts and databases:

	Source	Target
Hostname	vm4	vm5
Database Name	NORTH	SOUTH

Prepare Databases

Enable Forced Logging

Forced logging should be enabled on the source database to ensure that all changes are written to the redo logs. DBCA does not enable force logging during database creation.

To check if FORCE_LOGGING is enabled in the source database use:

```
[oracle@vm4]$ sqlplus / as sysdba
SQL> SELECT force_logging FROM v$database;

FORCE_LOGGING
-----
NO
```

If not enabled, then enable FORCE_LOGGING using:

```
[oracle@vm4]$ sqlplus / as sysdba
SQL> ALTER DATABASE FORCE LOGGING;

Database altered.
```

Verify that FORCE_LOGGING has been enabled successfully using:

```
[oracle@vm4]$ sqlplus / as sysdba
SQL> SELECT force_logging FROM v$database;

FORCE_LOGGING
-----
YES
```

The documentation recommends performing a redo log switch to ensure that the change is applied in all subsequent redo.

```
[oracle@vm4]$ sqlplus / as sysdba
SQL> ALTER SYSTEM SWITCH LOGFILE;

System altered.
```

Enable Minimal Supplemental Logging

If minimal supplemental logging is not enabled, the extract process fails to start. DBCA does not enable minimal supplemental during database creation.

Minimal supplemental logging only needs to be configured on the source database. It may be prudent to enable it on the target database as well.

Check if minimal supplemental logging is currently enabled:

```
[oracle@vm4]$ sqlplus / as sysdba
SQL> SELECT supplemental_log_data_min FROM v$database;

SUPPLEMENTAL_LOG_DATA_MIN
-----
NO
```

If not enabled then configure minimal supplemental logging:

```
[oracle@vm4]$ sqlplus / as sysdba
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;

Database altered.
```

Verify that minimal supplemental logging is now enabled:

```
[oracle@vm4]$ sqlplus / as sysdba
SQL> SELECT supplemental_log_data_min FROM v$database;

SUPPLEMENTAL_LOG_DATA_MIN
-----
YES
```

Switch the log file again to ensure that all subsequent redo contains minimal supplemental logging:

```
[oracle@vm4]$ sqlplus / as sysdba
SQL> ALTER SYSTEM SWITCH LOGFILE;

System altered.
```

Prepare Test Environment

Create Test User

In order to test the GoldenGate configuration I created a new schema (US01) containing a new table (T1) as follows. The table is derived from DBA_OBJECTS.

In both databases create a user (schema) called US01. For example:

```
[oracle@vm4]$ sqlplus / as sysdba
SQL> CREATE USER us01 IDENTIFIED BY us01;

User created.
```

In both databases grant the following permissions to the new user. For example:

```
[oracle@vm4]$ sqlplus / as sysdba
SQL> GRANT CONNECT,RESOURCE,DBA TO us01;

Grant succeeded.
```

As these are test databases, security is not an issue and therefore DBA privilege has been granted to the new user.

Create Test Table

In the source database (NORTH) create the T1 table using a subset of rows from DBA_OBJECTS

```
[oracle@vm4]$ sqlplus us01/us01
SQL> CREATE TABLE t1 AS
SELECT object_id,owner,object_name,object_type
FROM dba_objects
WHERE object_id <= 10000;

Table created.
```

In the source database (NORTH), add a primary key constraint and index to table T1:

```
[oracle@vm4]$ sqlplus us01/us01
SQL> ALTER TABLE t1 ADD CONSTRAINT t1_i1 PRIMARY KEY (object_id);

Table altered.
```

Start Listener Processes

Before attempting to configure the network, ensure that the listener processes are running on both servers.

```
[oracle@vm4]$ lsnrctl start
[oracle@vm5]$ lsnrctl start
```

Configure Network

For this basic configuration, an entry is required in the *TNSNAMES.ORA* file on each server describing the database on the other server.

On the source database (vm4) I added the following entry for the SOUTH database to *\$ORACLE_HOME/network/admin/tnsnames.ora*

```
SOUTH =
  (DESCRIPTION=
    (ADDRESS= (PROTOCOL=TCP) (HOST=vm5) (PORT=1521) )
    (CONNECT_DATA=
      (SERVER=DEDICATED)
      (SERVICE_NAME=SOUTH)
    )
  )
```

Verify the connection using SQL*Plus. For example:

```
[oracle@vm4]$ sqlplus us01/us01@SOUTH
SQL> SELECT name FROM v$database;

NAME
-----
SOUTH
```

On the target database (vm5), I added the following entry for the NORTH database to *\$ORACLE_HOME/network/admin/tnsnames.ora*

```
NORTH =
  (DESCRIPTION=
    (ADDRESS= (PROTOCOL=TCP) (HOST=vm4) (PORT=1521) )
    (CONNECT_DATA=
      (SERVER=DEDICATED)
      (SERVICE_NAME=NORTH)
    )
  )
```

Verify the connection using SQL*Plus. For example:

```
[oracle@vm5]$ sqlplus us01/us01@NORTH
SQL> SELECT name FROM v$database;

NAME
-----
NORTH
```

Create Database Links

On each server create a database link to the other database. This is only required to set up the test data. It is not required for GoldenGate.

On the source server (vm4) as the US01 user, create the following database link:

```
[oracle@vm4]$ sqlplus us01/us01
SQL> CREATE DATABASE LINK south

CONNECT TO us01 IDENTIFIED BY us01 USING 'SOUTH';
```

On the target server (vm5) as the US01 user, create the following database link:

```
[oracle@vm5]$ sqlplus us01/us01
SQL> CREATE DATABASE LINK north

CONNECT TO us01 IDENTIFIED BY us01 USING 'NORTH';
```

Copy Test Data to Target Database

Create an initial copy of the test data on the target database

In the target database (SOUTH) create a copy of the T1 table using the SQL*Plus command:

```
[oracle@vm5]$ sqlplus us01/us01
SQL> CREATE TABLE t1 AS SELECT * FROM t1@NORTH;
```

In the target database (SOUTH), add a primary key constraint and index to table T1:

```
[oracle@vm5]$ sqlplus us01/us01
SQL> ALTER TABLE t1 ADD CONSTRAINT t1_i1 PRIMARY KEY (object_id);
```

In the source database (NORTH), verify the number of rows in the original table:

```
[oracle@vm4]$ sqlplus us01/us01
SQL> SELECT COUNT(*) FROM t1;

COUNT(*)
-----
9830
```

In the target database, verify that there is the same number of rows in the new table:

```
[oracle@vm5]$ sqlplus us01/us01
SQL> SELECT COUNT(*) FROM t1;

COUNT(*)
-----
9830
```

Note that the actual number of rows in table T1 should be identical for databases NORTH and SOUTH. However, it may vary in other databases

Configure GoldenGate

Download and Install GoldenGate software

The GoldenGate software can be downloaded from downloads.oracle.com or alternatively from the Oracle E-Delivery website. The software should be installed on both servers.

The 11.2.1.0.1 file for Linux x86-64 is `ogg112101_fbo_ggs_Linux_x64_oral1g_64bit.zip`

The zip file is around 90MB in size.

On each server, create a new directory for the GoldenGate software:

```
[oracle@vm4]$ mkdir /home/oracle/goldengate
```

Copy the download file to the new directory and unzip using:

```
[oracle@vm4]$ unzip ogg112101_fbo_ggs_Linux_x64_oral1g_64bit.zip
Archive: ogg112101_fbo_ggs_Linux_x64_oral1g_64bit.zip
inflating: fbo_ggs_Linux_x64_oral1g_64bit.tar
inflating: OGG_WinUnix_Rel_Notes_11.2.1.0.1.pdf
inflating: Oracle GoldenGate 11.2.1.0.1 README.txt
inflating: Oracle GoldenGate 11.2.1.0.1 README.doc
```

The tar archive is around 230MB in size.

Extract the files from the tar archive using:

```
[oracle@vm4]$ tar xfv fbo_ggs_Linux_x64_oral1g_64bit.tar
```

Configure Environment Variables

The `/home/oracle/.bash_profile` should already contain the following environment variables.

```
export ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
export PATH=$ORACLE_HOME/bin:$PATH
```

On the source server add the following entries to `/home/oracle/.bash_profile`

```
export ORACLE_SID=NORTH
export PATH=/home/oracle/goldengate:$PATH
export LD_LIBRARY_PATH=/home/oracle/goldengate:$ORACLE_HOME/lib
```

Ensure the environment variables are set using:

```
[oracle@vm4]$ source /home/oracle/.bash_profile
```

On the target server add the following entries to `/home/oracle/.profile`

```
export ORACLE_SID=SOUTH
export PATH=/home/oracle/goldengate:$PATH
export LD_LIBRARY_PATH=/home/oracle/goldengate:$ORACLE_HOME/lib
```

Ensure the environment variables are set using:

```
[oracle@vm5]$ source /home/oracle/.bash_profile
```

Create GoldenGate Subdirectories

On each server create subdirectories for GoldenGate using the CREATE SUBDIRS command. For example:

```
[oracle@vm4]$ ggsci
GGSCI (vm4) 1> CREATE SUBDIRS
Creating subdirectories under current directory /home/oracle/goldengate

Parameter files /home/oracle/goldengate/dirprm: already exists
Report files /home/oracle/goldengate/dirrpt: created
Checkpoint files /home/oracle/goldengate/dirchk: created
Process status files /home/oracle/goldengate/dirpcs: created
SQL script files /home/oracle/goldengate/dirsq: created
Database definitions files /home/oracle/goldengate/dirdef: created
Extract data files /home/oracle/goldengate/dirdat: created
Temporary files /home/oracle/goldengate/dirtmp: created
Stdout files /home/oracle/goldengate/dirout: created
```

Create the GoldenGate Schema Owner

A new user should be created to own the GoldenGate database objects.

On each server create the GoldenGate schema owner. For example:

```
[oracle@vm4]$ sqlplus / as sysdba
SQL> CREATE USER gg01 IDENTIFIED BY gg01;

User created.
```

On each server grant DBA role to the GoldenGate schema owner

```
[oracle@vm4]$ sqlplus / as sysdba
SQL> GRANT CONNECT, RESOURCE, DBA TO gg01;

Grant succeeded.
```

On each server set the GGSHEMA in the global parameter file.

```
[oracle@vm4]$ ggsci
GGSCI (vm4) 1> EDIT PARAMS ./GLOBALS
```

In this example the parameters file is /home/oracle/goldengate/GLOBALS

Add the following entry:

```
GGSCHEMA gg01
```

Save and close the file

Create GoldenGate Tablespace

On the source server create a new tablespace for the GoldenGate objects. Ensure that AUTOEXTEND is enabled.

```
[oracle@vm4]$ sqlplus / as sysdba
SQL> CREATE TABLESPACE goldengate
DATAFILE '/u01/app/oradata/NORTH/goldengate01.dbf'
SIZE 100M
AUTOEXTEND ON;

Tablespace created.
```

Set the new tablespace as the default for the GoldenGate user:

```
[oracle@vm4]$ sqlplus / as sysdba
SQL> ALTER USER gg01 DEFAULT TABLESPACE goldengate;

User altered.
```

On the target server create a new tablespace for the GoldenGate objects. Again ensure that AUTOEXTEND is enabled.

```
[oracle@vm5]$ sqlplus / as sysdba
SQL> CREATE TABLESPACE goldengate
DATAFILE '/u01/app/oradata/SOUTH/goldengate01.dbf'
SIZE 100M
AUTOEXTEND ON;

Tablespace created.
```

Set the new tablespace as the default for the GoldenGate user:

```
[oracle@vm5]$ sqlplus / as sysdba
SQL> ALTER USER gg01 DEFAULT TABLESPACE goldengate;

User altered.
```

Run Role Setup script

On the source server run the Role setup script. Specify the GoldenGate schema name when prompted.

```
[oracle@vm4]$ sqlplus / as sysdba
SQL> @role_setup

GG5 Role setup script

This script will drop and recreate the role GGS_GGSUSER_ROLE
To use a different role name, quit this script and then edit the params.sql script
to change the gg_role parameter to the preferred name. (Do not run the script.)
You will be prompted for the name of a schema for the GoldenGate database objects.
NOTE: The schema must be created prior to running this script.
NOTE: Stop all DDL replication before starting this installation.
Enter GoldenGate schema name:GG01

Wrote file role_setup_set.txt

PL/SQL procedure successfully completed.

Role setup script complete

Grant this role to each user assigned to the Extract, GGSCI, and Manager processes,
by using the following SQL command:
GRANT GGS_GGSUSER_ROLE TO
where is the user assigned to the GoldenGate processes.
```

On the source server grant GGS_GGSUSER_ROLE to the GoldenGate user:

```
[oracle@vm4]$ sqlplus / as sysdba
SQL> GRANT GGS_GGSUSER_ROLE TO gg01;

Grant succeeded.
```

Configure Manager Parameters

On both servers configure the MGR parameters:

```
[oracle@vm4]$ ggsci
GGSCI (vm4) 1> EDIT PARAMS MGR
```

The above command created the file */home/oracle/goldengate/dirprm/mgr.prm*

Add the following parameters to the parameter file:

```
PORT 7809
DYNAMICPORTLIST 7810-7820
```

Save and close the parameter file

Configure Extract Parameters

In this example the extract process will be called "ex1"

On the source server, create the parameter file for Extract ex1:

```
[oracle@vm4]$ ggsci
GGSCI (vm4) 1> EDIT PARAMS ex1
```

The above command created the file */home/oracle/goldengate/dirprm/ex1.prm*

Add the following parameters to the new file:

```
EXTRACT ex1
USERID gg01, PASSWORD gg01
EXTTRAIL /home/oracle/goldengate/dirdat/ex
TABLE us01.*;
```

Configure Data Pump Parameters

In this example the Data Pump process will be called dp1

On the source server create the parameter file for Data Pump process dp1:

```
[oracle@vm4]$ ggsci
GGSCI (vm4) 1> EDIT PARAMS dp1
```

The above command created the file */home/oracle/goldengate/dirprm/dp1.prm*

Add the following parameters to the new file:

```
EXTRACT dp1
USERID gg01, PASSWORD gg01
RMTHOST vm5, MGRPORT 7809
RMTTRAIL /home/oracle/goldengate/dirdat/rt
TABLE us01.*;
```

Save and close the parameter file

Create Check Point Table

The check point table should be created in the target database.

On the target server login as the GG01 user and add the check point table:

```
[oracle@vm5]$ ggsci
GGSCI (vm5) 1> DBLOGIN USERID gg01, PASSWORD gg01
```

```
Successfully logged into database.
GGSCI (vm5) 2> ADD CHECKPOINTTABLE gg01.checkpointtable
Successfully created checkpoint table gg01.checkpointtable.
```

The name of the check point table must be added to the GLOBALS file on the target server.

On the target server edit the GLOBALS file

```
[oracle@vm5]$ ggsci
GGSCI (vm5) 1> EDIT PARAMS ./GLOBALS
```

Add the CHECKPOINTTABLE parameter to the existing file. For example:

```
GGSCHEMA gg01
CHECKPOINTTABLE gg01.checkpointtable
```

Save and close the GLOBALS parameter file.

Configure Replication Parameters

On the target server create the parameter file for replication process rep1:

```
[oracle@vm5]$ ggsci
GGSCI (vm5) 1> EDIT PARAMS rep1
```

The above command created the file `/home/oracle/goldengate/dirprm/rep1.prm`

Add the following parameters to the new file:

```
REPLICAT rep1
USERID gg01, PASSWORD gg01
ASSUMETARGETDEFS
DISCARDFILE /home/oracle/goldengate/discards, PURGE
MAP us01.*, TARGET us01.*;
```

Note that the DISCARDFILE parameter includes the PURGE keyword. If PURGE is not specified then the replication process will fail the second time it is started. Alternatively use the APPEND keyword to append output to the existing file.

Configure Supplemental Logging for Replicated Tables

On the source server configure supplemental logging for all tables that will be replicated. In this example there is only one table (T1)

Supplemental logging can be configured by any user that has privileges to modify the underlying database table.

```
[oracle@vm4]$ ggsci
GGSCI (vm4) 1> DBLOGIN USERID us01, PASSWORD us01
Successfully logged into database.

GGSCI (vm4) 2> ADD TRANDATA t1
Logging of supplemental redo data enabled for table US01.T1.
```

Add the Extract Process

On the source server add the Extract process (ex1)

```
[oracle@vm4]$ ggsci
GGSCI (vm4) 1> ADD EXTRACT ex1, TRANLOG, BEGIN NOW

EXTRACT added.
```

Add the Extract Trail

On the source server add the Extract trail (*/home/oracle/goldengate/dirdat/ex*)

```
[oracle@vm4]$ ggsci
GGSCI (vm4) 1> ADD EXTTRAIL /home/oracle/goldengate/dirdat/ex, EXTRACT ex1

EXTTRAIL added.
```

Add the Data Pump Process

On the source server add the Data Pump process (dp1)

```
[oracle@vm4]$ ggsci
GGSCI (vm4) 1> ADD EXTRACT dp1 EXTTRAILSOURCE /home/oracle/goldengate/dirdat/ex

EXTRACT added.
```

Add the Data Pump Trail

On the source server add the Data Pump trail (*/home/oracle/gg/dirdat/rt*). This trail is created on the target server. However, the name is required in order to set up the Data Pump process on the source server.

```
[oracle@vm4]$ ggsci
GGSCI (vm4) 1> ADD RMTTRAIL /home/oracle/goldengate/dirdat/rt, EXTRACT dp1

RMTTRAIL added.
```

Add the Replication Process

On the target server add the Replication process (rep1)

```
[oracle@vm5]$ ggsci
GGSCI (vm5) 1> ADD REPLICAT rep1, EXTTRAIL /home/oracle/goldengate/dirdat/rt

REPLICAT added.
```

Start GoldenGate

Start Manager

On the source server, start the GoldenGate manager:

```
[oracle@vm4]$ ggsci
GGSCI (vm4) 1> START MANAGER

Manager started.
```

On the target server, start the GoldenGate manager:

```
[oracle@vm5]$ ggsci
GGSCI (vm5) 1> START MANAGER

Manager started.
```

Start Extract Process

On the source server start the Extract (ex1)

```
[oracle@vm4]$ ggsci
GGSCI (vm4) 1> START EXTRACT ex1

Sending START request to MANAGER ...

EXTRACT EX1 starting

Verify that the Extract has started successfully using INFO EXTRACT:

[oracle@vm4]$ ggsci
GGSCI (vm4) 10> INFO EXTRACT ex1
EXTRACT EX1 Last Started 2013-02-27 12:57 Status RUNNING
Checkpoint Lag 00:00:24 (updated 00:00:05 ago)
Log Read Checkpoint Oracle Redo Logs
2013-02-27 12:57:01 Seqno 6, RBA 30736
SCN 0.0 (0)
```

The status should be RUNNING.

Start Data Pump Process

On the source server, start the Data Pump (dp1):

```
[oracle@vm4]$ ggsci
GGSCI (vm4.com) 3> START EXTRACT dp1

Sending START request to MANAGER ...

EXTRACT DP1 starting
```

Verify that the Data Pump has started successfully using INFO EXTRACT:

```
[oracle@vm4]$ ggsci
GGSCI (vm4.com) 2> INFO EXTRACT dp1
EXTRACT DP1 Last Started 2013-02-27 11:57 Status RUNNING
Checkpoint Lag 00:00:00 (updated 00:00:00 ago)
Log Read Checkpoint File /home/oracle/gg/dirdat/ex000000
First Record RBA 0
```

The status should be RUNNING.

Start Replication Process

On the target server, start the Replicat process (rep1):

```
[oracle@vm5]$ ggsci
GGSCI (vm5) 1> START REPLICAT rep1
Sending START request to MANAGER ...
REPLICAT REP1 starting
```

Verify that the Replicat process has started successfully using INFO EXTRACT:

```
[oracle@vm5]$ ggsci
GGSCI (vm5) 2> INFO REPLICAT rep1
REPLICAT REP1 Last Started 2013-02-27 11:58 Status RUNNING
Checkpoint Lag 00:00:00 (updated 00:00:04 ago)
Log Read Checkpoint File /home/oracle/gg/dirdat/rt000000
First Record RBA 0
```

The status should be RUNNING.

Test Replication

On the source server, check the number of rows in table T1:

```
[oracle@vm4]$ sqlplus us01/us01
SQL> SELECT COUNT(*) FROM t1;

COUNT(*)
-----
9830
```

On the target server, check the number of rows in table T1:

```
[oracle@vm5]$ sqlplus us01/us01
SQL> SELECT COUNT(*) FROM t1;

COUNT(*)
-----
9830
```

On the source server, add some rows to table T1. For example:

```
[oracle@vm4]$ sqlplus us01/us01
SQL> INSERT INTO t1 (object_id,owner,object_name,object_type)
SELECT object_id,owner,object_name,object_type
FROM dba_objects
WHERE object_id BETWEEN 10001 AND 11000;

1000 rows created

SQL> COMMIT;

Commit complete
```

On the source server check the number of rows in table T1:

```
[oracle@vm4]$ sqlplus us01/us01
SQL> SELECT COUNT(*) FROM t1;

COUNT(*)
-----
10830
```

On the target server check the number of rows in table T1:

```
[oracle@vm5]$ sqlplus us01/us01
SQL> SELECT COUNT(*) FROM t1;

COUNT(*)
-----
10830
```

The number of rows should be identical in both tables

This completes the basic configuration.