

Write by: Munoz, Gabriel
Email: gabriel.munoz@acando.no

How to install and configure Oracle Golden Gate – Initial Setup

A while ago I was presented with the challenge of synchronizing several databases in order to move data from a source database to a stage database and finally to a target database. Although there are several ways to achieve this, it was decided to use Oracle Golden Gate to migrate the data, in real time, to a stage server and then to a target server.

There was also a need of creating some interfaces in order to mask the database structure on source server and fulfill the customer's demands. There the Oracle Data Integrator comes in, but we will discuss the ODI on another blog.

A- Topology

I have installed Oracle VirtualBox 5.0.2 on my computer. I have created two virtual servers which contain Oracle Enterprise Linux 6.7 x86-64 and Oracle Database 12c, version: 12.1.0.2. The Oracle Golden Gate I installed is version: 12.1.2.1.

- Source server: dbsrc
- Stage server: stgdb
- Target server: dbtgt

The stage and target databases reside on the same server for simplicity. In this blog I will focus on Oracle Golden Gate initial setup on both source database dbsrc and target database stgdb. The database dbtgt will be used as an example in my future blog on Oracle Data Integrator.

Source database dbsrc and target database stgdb

I downloaded the Oracle Golden Gate file:

121210_fbo_ggs_Linux_x64_shiphome.zip from <http://otn.oracle.com>. There are several download files to choose from. One has to choose the one corresponding to the relevant technology. In this case Oracle Golden Gate for Oracle Linux x86-64.

B- Installing Oracle Golden Gate

I logged on source and target servers as oracle user that owns the software of Oracle Database and Oracle Golden Gate.

I moved the file *121210_fbo_ggs_Linux_x64_shiphome.zip* to source and target server and unzipped the file.

The folder created on source and target servers is then: *fbo_ggs_Linux_x64_shiphome*

SourceServer

Then change directory to *fbo_ggs_Linux_x64_shiphome/Disk1*.

At this point one has to check that the following environment variables are set on source server:

Write by: Munoz, Gabriel

Email: gabriel.munoz@acando.no

```
- ORACLE_HOME = /db0/oracle/product/12.1.0.2/db_1  
- ORACLE_SID = dbsrc
```

and on target server:

```
- ORACLE_HOME = /db0/oracle/product/12.1.0.2/db_1  
- ORACLE_SID = stgdb
```

The images below show the installation of Oracle Golden Gate on source server and the same installation should be performed on target server.

Installation Oracle Golden Gate on source server

Press Next:

Installation - step 2 of 5

Specify the home of Oracle Golden Gate. Choose an Oracle_Home different from the one of the Oracle database and press Next:

Installation Step 3 of 5

Then press Install.

Installation Step 5 of 5

Press Close.

At this point the OGG software is installed on source and target servers.

C- Configuring Oracle Golden Gate for initial setup

I will show here the initial configuration on source and target servers for initial extract and replicat. There are several steps to be performed on the source and target databases.

1- Update LD_LIBRARY_PATH in the profile file on both source and target servers.

For example my file .bash_profile on source server looks like this:

```
# .bash_profile  
  
# Get the aliases and functions  
if [ -f ~/.bashrc ]; then  
    ~/.bashrc  
fi  
  
# User specific environment and startup programs  
  
PATH=$PATH:$HOME/bin
```

Write by: Munoz, Gabriel

Email: gabriel.munoz@acando.no

```
export PATH

# Oracle Settings
export TMP=/tmp
export TMPDIR=$TMP

export ORACLE_HOSTNAME=srcdb.localdomain
export ORACLE_UNQNAME=dbsrc
export ORACLE_BASE=/db0/oracle
export ORACLE_HOME=$ORACLE_BASE/product/12.1.0.2/db_1
export ORACLE_GG_HOME=$ORACLE_BASE/product/12.1.2.1/ogg_1
export ORACLE_SID=dbsrc

export PATH=/usr/sbin:$PATH
export PATH=$ORACLE_HOME/bin:$PATH

export LD_LIBRARY_PATH=$ORACLE_GG_HOME:$ORACLE_HOME/lib:/lib:/usr/lib
export CLASSPATH=$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib

if [ $USER = "oracle" ]; then
if [ $SHELL = "/bin/ksh" ]; then
ulimit -p 16384
ulimit -n 65536
else
ulimit -u 16384 -n 65536
fi
fi
```

And the .bash_profile on target server should look similar, but the variables ORACLE_UNQNAME and ORACLE_SID have to be set to stgdb to reflect the target database.

2- Enabling Supplemental logging on source database dbsrc in order to log all transactions and loads. This step is not required for target database stgdb.

Log on source database as sys. I use Oracle SQLDeveloper which gives a nice graphical interface against the database:

```
SELECT supplemental_log_data_min, force_logging FROM v$database;
```

If the result is NO then run:

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
ALTER DATABASE FORCE LOGGING;
```

Then run again:

```
SELECT supplemental_log_data_min, force_logging FROM v$database;
```

Write by: Munoz, Gabriel

Email: gabriel.munoz@acando.no

The result should be YES now.

Then switch the log files:

```
ALTER SYSTEM SWITCH LOGFILE;
```

3- Enable archivelog. The Oracle source database must have archivelog enabled. This step is not required for target database.

Log on source database dbsrc as sys and run:

```
SQL> archive log list;
SQL> shutdown immediate;
SQL> startup mount;
SQL> alter database archivelog;
SQL> alter database open;
SQL> archive log list;
```

4.- Create Oracle Golden Gate user on source and target databases:

Log on source and target databases as sys:

a- Create tablespace for OGG user:

```
CREATE TABLESPACE ogg_data LOGGING
DATAFILE '/db0/oracle/oradata/dbsrc/ogg_data_01.dbf'
SIZE 1024M REUSE AUTOEXTEND ON NEXT 512M MAXSIZE UNLIMITED
EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO;
Here I have changed the location of datafile on target server to:
DATAFILE '/db0/oracle/oradata/stgdb/ogg_data_01.dbf'
```

b- Create Golden Gate user:

```
CREATE USER ogg_user IDENTIFIED BY oracle
DEFAULT TABLESPACE ogg_data
TEMPORARY TABLESPACE temp;
```

c- Grant relevant privileges to user ogg_user:

```
GRANT CONNECT, RESOURCE TO ogg_user;
GRANT SELECT ANY DICTIONARY, SELECT ANY TABLE TO ogg_user;
GRANT CREATE TABLE TO ogg_user;
GRANT FLASHBACK ANY TABLE TO ogg_user;
GRANT EXECUTE ON dbms_flashback TO ogg_user;
GRANT EXECUTE ON utl_file TO ogg_user;
GRANT CREATE ANY TABLE TO ogg_user;
GRANT INSERT ANY TABLE TO ogg_user;
GRANT UPDATE ANY TABLE TO ogg_user;
```

Write by: Munoz, Gabriel

Email: gabriel.munoz@acando.no

```
GRANT DELETE ANY TABLE TO ogg_user;  
GRANT DROP ANY TABLE TO ogg_user;  
GRANT ALTER ANY TABLE TO ogg_user;  
GRANT ALTER SYSTEM TO ogg_user;  
GRANT LOCK ANY TABLE TO ogg_user;  
GRANT SELECT ANY TRANSACTION to ogg_user;  
ALTER USER ogg_user QUOTA UNLIMITED ON ogg_data;
```

d- Authenticate the user ogg_user with Golden Gate package:

```
exec DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE('ogg_user');
```

d- Enable Golden Gate on both databases:

```
ALTER SYSTEM SET enable_goldengate_replication=TRUE;
```

5a.- Edit manager parameter on source server:

Go to Golden Gate home:

```
$ cd $ORACLE_GG_HOME
```

Then run:

```
$. /ggsci
```

Golden gate home

Then type:

```
ggsci > edit param mgr
```

Then the parameter manager file on source server should look like this:

```
PORT 7809  
PurgeOldExtracts /db0/oracle/product/12.1.2.1/ogg_1/dirdat/* UseCheckPoints, MinKeepHours 48
```

5b.- Edit manager parameter on target server:

```
ggsci > info mgr  
ggsci > edit param mgr
```

Then the parameter manager file on target server should look like this:

```
PORT 7809
```

Save the changes and then return to ggsci command interface.

Then run the following commands:

Write by: Munoz, Gabriel

Email: gabriel.munoz@acando.no

```
ggsci > stop mgr
ggsci > start mgr
ggsci > info mgr detail
```

Edit manager parameter on target server

6.- Storing credentials on wallet. This step is performed on both source and target servers. Run the following commands on Golden Gate interface ggsci. The connect string dbsrc below is the tnsnames of the source database.

```
ggsci > create wallet
ggsci > Add CredentialStore
ggsci > alter CredentialStore Add User ogg_user@dbsrc Password oracle Alias ogg_user
ggsci > info CredentialStore
```

Similarly the commands on Golden Gate interface ggsci on target server are:

```
ggsci > create wallet
ggsci > Add CredentialStore
ggsci > alter CredentialStore Add User ogg_user@stgdb Password oracle Alias ogg_user
ggsci > info CredentialStore
```

Storing credentials on wallet

Then we test the credentials. Run the following command on ggsci interface on both source and target servers:

```
ggsci > DBLogin UserIDAlias ogg_user
```

Successfully logged into database.

At this point the initial Oracle Golden Gate setup is completed and ready on source and target database servers.

On the next blogs I will cover examples of Oracle Golden Gate initial load and Oracle Data Integrator.

Sources:

*Photo: "GoldenGateBridge-001" by Rich Niewiroski Jr. - <http://www.projectrich.com/gallery>. Licensed under CC BY 2.5 via Wikimedia Commons