



Technical Notes IBM Oracle International Competency Center (ICC)

March 25, 2013

email address: ibmoracle@us.ibm.com

Hints and Tips on migrating Oracle RAC[™] Database for different platforms & versions using Oracle GoldenGate[™]

This paper is based on our experiences migrating an Oracle RAC Database 10.2.0.5 running Red Hat Enterprise Linux 5.6 from a HP[™] Integrity BL860c Server Blade to an Oracle RAC Database 11.2.0.3 running on an IBM POWER[™] 750 System with AIX 7.1 using Oracle GoldenGate 11.2.1.

The source environment had only 1 database with a size of 2 TB. Originally we used some known methods of migration such as export & import data and transport tablespace but we decided to use Oracle Golden Gate which we believe could give us better migration time and this would allow that our client's business had a minimal downtime time during migration.

This document is published "as-is". There is no guarantee from IBM that the procedures described below will work on every installation. This Technical Note merely describes the experiences of the author at a specific installation.

Setting up the Environment

Before we start our implementation, our target database must be created. It is recommended that this database has the same globalization and configuration characteristics as the source database. I recommend creating a template through the utility DBCA in the source database and then creating the target database using this template. After that, we can review it and change the current parameters because the target database is now on version 11.2.0.3.

In my implementation the schema of all the core applications is named EDYFICAR. We did an export and import of the entire schema EDYFICAR without contents, example:

Source Database:

```
s12rac01> expdp userid=system/oracle directory=DIR_EXP schemas=EDYFICAR  
content=METADATA_ONLY dumpfile=exp_bd.dmp logfile=exp_bd.log
```

Target Database:

```
sp12rac01> expdp userid=system/oracle directory=DIR_IMP full=y dumpfile=exp_bd.dmp  
logfile=imp_bd.log
```



The contents will be populated with Oracle GoldenGate.

I. Installing Oracle GoldenGate in both environments (Source / Target).

a) Download the software.

Oracle GoldenGate is a tool that allows us to do migrations with zero-downtime, replicating information between heterogenic environments.

Information about the product can be found at the following URL:

<http://www.oracle.com/technetwork/middleware/goldengate/documentation/index.html>

The product can be downloaded via the following URL:

<http://www.oracle.com/technetwork/middleware/goldengate/downloads/index.html>

For my scenario I downloaded the following installers:

- Oracle GoldenGate V11.2.1.0.2 for Oracle 10g on Linux Itanium
- Oracle GoldenGate V11.2.1.0.1 for Oracle 11g 64bit for AIX 5.3/6.1

Once we have downloaded the installers, we will proceed to copy the installer in each server and decompress it, example: tar xvf ggs_AIX_ppc_ora11g_64bit.tar

The hostnames of my source servers are: s12rac01 and s12rac02 and the others in my target servers are: sp12rac01 and sp12rac02.

Note 1: The user who execute the software of GoldenGate must have to set up correctly the variable environments ORACLE_BASE, ORACLE_HOME, ORACLE_SID, NLS_LANG and LD_LIBRARY_PATH (Linux) /LIBPATH(AIX).

According to My Oracle Support (MOS) Note: 969551.1 (How is NLS_LANG Set When You Run GoldenGate on the Superset Character Sets Database?), it is important that we define the NLS_LANG to the database configuration, then, it will assume US7ASCII.

Note 2: In the case of Oracle RAC as source database, if the archived logs aren't stored in a store shared, Oracle GoldenGate must be installed in each server **that** forms **a** cluster. In my scenario our source database saved is archived logs in a diskgroup of ASM (store shared), so we must only install Oracle GoldenGate in a single node of all cluster.

b) Create GoldenGate's directory

In the unzip folder we must execute the following command:

```
s12rac01> ./ggsci  
GGSCI> create subdirs
```



II. Configuration of Oracle GoldenGate - Source Database.

- a) We must create a user database to connect it with Oracle GoldenGate (this step must be done also in the target database). In our implementation the user database is gg.

```
SQL> create user gg identified by oracle;  
SQL> grant DBA to gg;
```

Note: If we don't want to give the DBA role to our new user then we must execute the following procedure:

```
cd <folder_golden_gate>  
sqlplus / as sysdba  
SQL> @marker_setup.sql  
SQL> @ddl_setup.sql  
SQL> @role_setup.sql  
SQL> grant ggs_ggsuser_role to gg;
```

- b) Enable supplemental logging in the database.

```
SQL> alter database add supplemental log data;
```

We can verify it executing the following query and it must give us the value of YES.

```
SQL> select supplemental_log_data_min from v$database;
```

- c) Each table that we would like to replicate must have the supplemental setted up.

```
s12rac01> ./ggsci  
GGSCI> dblogin userid gg password oracle  
GGSCI> add trandata edyficar.MOVIMIENTOS  
GGSCI> add trandata edyficar.SALDOS_DIARIOS  
GGSCI> add trandata edyficar.MOVIMIENTOS_CONTABLES  
GGSCI> add trandata edyficar.CO_BALANCE_RESULTADOS  
GGSCI> add trandata edyficar.BITACORA  
GGSCI> add trandata edyficar.HISTORIA_VISTA  
GGSCI> add trandata edyficar.HISTORY  
GGSCI> add trandata edyficar.REP_HOJARESUMEN  
GGSCI> add trandata edyficar.ASIENTOS  
GGSCI> add trandata edyficar.REP_PAGARE  
GGSCI> add trandata edyficar.RCC_RES_GEN  
...
```

We can also generate a script and then we can execute it, example:

Creating the script:

```
SQL> set head off
SQL> set pagesize 0
SQL> spool script.obey
SQL> select 'add trandata edyficar.'||table_name from dba_tables where owner='EDYFICAR';
SQL> spool off
```

Executing the script:

```
GGSCI> dblogin userid gg password oracle
GGSCI> obey script.obey
```

If we wish to know which are the tables where the supplemental are not activated, we must execute the following script:

```
SQL> select table_name
       from dba_tables
       where owner='EDYFICAR'
       minus
       select table_name
       from dba_log_groups
       where owner = 'EDYFICAR';
```

It is important to understand some Oracle GoldenGate's background processes and concepts that we will run in our replicat solution.

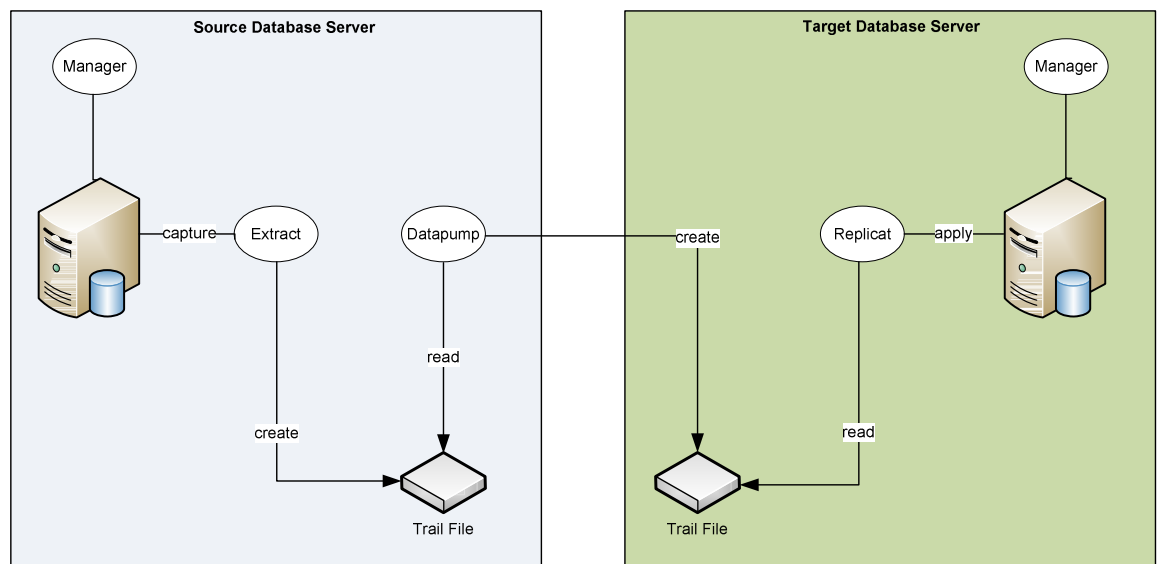


Figure 1

Each installation of Oracle GoldenGate must be running only one Manager process which is responsible of controlling and coordinating the other background process of Oracle GoldenGate.



Extract is the process of extracting data that is inserted, updated or deleted from the source database. The extract process creates files named trail files where it saves the captured information of our source database.

Datapump is an extract process that reads the trail file in the source environment and creates a remote trail file in the target server. The remote file is an exact copy of the trail file. It is an optional process due to the fact that extract process also can do the same action but it is not recommendable.

Replicat process is responsible of reading the trail file created by datapump process and applies the changes in the target database.

We can get an amount of extract, datapump and replicat that we needed. If our database has a considerable size it is not recommended to create only one extract and one replicat.

With this minimum information we will configure the Manager, Extract and Datapump process in the source environment.

d) Configuration Manager.

This step must be done also in the target database

```
GGSCI> edit params mgr
      port 7840
      purgeoldextracts dirdat/*

GGSCI> start manager
```

Here we configure a Manager who will work in the port 7840 TCP/IP. The manager will purge the trail files that were applied in the target database. Optionally we can have this syntax:

```
purgeoldextracts dirdat/*, usecheckpoints, minkeepdays 2
```

Here we specified that will remain 2 days of trail files applied in disk.

Note 1: It is important to know that the log of Oracle GoldenGate is the file ggserr.log and it is found in the stage of Oracle GoldenGate.

Note 2: The configuration's files of the background process (manager, extract and replicat) are found in the following directory: <software_golden_gate>/dirprm.

e) Configuration Extractor.

The extractor must be able to read every archived and redo log generated by every instance in the Oracle RAC Database.

In my scenario I have created 6 extracts where each one extracts a group by a set of tables. I will detail only 1 extract.

```
GGSCI> edit param ext1c
      extract ext1c
      userid gg, password oracle
      tranlogoptions asmuser sys@ASM, asmpassword <password_sys_asm>
      extrail dirdat/tc
      table EDYFICAR.MOVIMIENTOS_CONTABLES;
      table EDYFICAR.CO_BALANCE_RESULTADOS;
      table EDYFICAR.BITACORA;
      table EDYFICAR.HISTORIA_VISTA;
      table EDYFICAR.HISTORY;

GGSCI> add extract ext1c, tranlog, threads 2, begin now
      add extrail dirdat/tc, extract ext1c, MEGABYTES 1024

GGSCI> start extract ext1c
```

In the configuration, we specify that there is a connection to the instance ASM and the information extracted from the archived and redo logs are recorded in text files named trail files in its size of 1 GB as a maximum in the directory <software_golden_gate>/dirdat/. It is not recommended to work with a big trail file size.

The trail files will have the name of "tc" with a number of 6 digits example: tc000001, tc000002, etc.

I recommend specifying only 2 characters for the name of the trail files since Oracle Golden Gate will complete the name with 6 digits as the example showed before.

If the source environment is an Oracle RAC Database we must specify the number of threads in the cluster, it is specified in the following line: tranlog, threads #.

Note: The listener of our source database must have a static register of our instance +ASM.



Datapump Configuration.

In my scenario I have created also 6 datapump extractors. I will detail only 1 datapump extract.

```
GGSCI> edit params dp1c
      extract dp1c
      passthru
      rmthost sp12rac01, mgrport 7840
      rmtrail dirdat/tc
      table EDYFICAR.MOVIMIENTOS_CONTABLES;
      table EDYFICAR.CO_BALANCE_RESULTADOS;

GGSCI> add extract dp1c, extrailsources dirdat/tc
      add rmtrail dirdat/tc, extract dp1c, MEGABYTES 1024

GGSCI> start extract dp1c
```

Here we are creating a Datapump process; it will communicate it with the Manager in the new server (sp12rac01) by the port 7840 TCP/IP.

The datapump process will read the trail files stored in <software_golden_gate>/dirdat/tc<xxxxxx> and it will create remotely trail files in the new server in sizes of 1 GB in the format tc plus 6 digits.

This new remote trail file only will have information about the changes of the tables MOVIMIENTOS_CONTABLES and CO_BALANCE_RESULTADOS according to the configuration of this datapump extract process showed.

The option passthru indicates that Oracle GoldenGate won't check if source and target tables defined previously in the datapump extract process are the same even at level of data types, structures, etc. It is recommended that tables were created with the same structure because it gives them better performance.

At this point we have obtained the entire configuration captured from the source database and it has been sent to target database.

We can validate our configuration with the following scripts:

```
GGSCI> info all (We can see the states of all the background process configured)

GGSCI> show report (We can see if there are errors reported by Oracle Golden Gate).
```

Note: If we wish to implement a bi-directional replication, we must execute the following steps:

```
GGSCI> edit param ext1c
      extrail <file_name>
      ...
      tranlogoptions excludeuser gg;
      table <table_name>
```



...

The option `tranlogoptions excludeuser` allows that transactions made by the user `gg` can't be captured in the source database, it will avoid that a transaction do an infinite cycle in a configuration bi-directional.

Source and target environments must have 1 extract and 1 replicat each one.

More information in My Oracle Support (MOS) Note: 1404384.1 (How to Avoid Transaction Looping in an Active-Active(Bi-Directional) Environment).



III. Configuration Oracle Golden Gate - Target Database.

- a) Replicat will be created in the target database. In my implementation I have created 6 replicats. In this document I will detail the configuration of 1 replicat:

```
GGSCI> edit param rep2c
      replicat rep2c
      userid gg, password oracle
      assumetargetdefs
      discardfile dirrpt/rep1c.txt, purge
      map EDYFICAR.MOVIMIENTOS_CONTABLES, target
      EDYFICAR.MOVIMIENTOS_CONTABLES;
      map EDYFICAR.CO_BALANCE_RESULTADOS, target
      EDYFICAR.CO_BALANCE_RESULTADOS;

GGSCI> dblogin userid gg, password oracle
GGSCI> add checkpointtable gg.CKPT3
GGSCI> add replicat rep1c, extrail dirdat/tc, checkpointtable gg.CKPT3
```

This replicat process reads the trail files named tc<xxxxxx> created by the datapump replicat process and record the changes applied in a table named CKPT3. This process will record a log that will be remove each time that replicat process is restarted.

It is important that we don't start the replicat process still. We must begin the load process, where all the information in source database of an exact time will be copied in target database (this phase is named initial load); then target database will synchronize with respect to source database.

If we need to remove the extract process, we must execute the following commands:

```
GGSCI> stop replicat rep2c
GGSCI> delete replicat rep2c!
```

It is important that we disable the triggers and constraints with delete cascade in target database because it can duplicate the information. I can disable it with the following script:

```
SQL> set head off
      set linesize 500
      set pagesize 0
      set feed off
      spool script.sql
      select 'alter trigger edyficar.'||trigger_name||' disable;' from dba_triggers where
      owner='EDYFICAR';
      spool off

SQL> spool script.sql
      select 'alter table edyficar.'||table_name||' disable constraint "'||CONSTRAINT_NAME||"';'
      from dba_constraints
```



```
where delete_rule = 'CASCADE' and owner = 'EDYFICAR' and status='ENABLED';
```

It is also recommendable to disable primary keys, foreign keys and constraints because the process of synchronization can take a lot of time, for example:

```
SQL> alter table <table_name> disable constraint <constraint_name>;
```

```
SQL> spool script.sql  
      select 'alter table edyficar.'||table_name||' disable constraint "'||CONSTRAINT_NAME||"';'  
      from dba_constraints where owner='EDYFICAR' and status='ENABLED';  
      spool off
```

These triggers and constraints will be enabled after the migration process.



IV. Initial Load (Source / Target).

There are some methods in order to do the Initial Load process. In my scenario I choose the method export / import datapump because it is a faster method.

a) Source Database:

At this point, our extract and datapump process are running capturing information from our source database.

We will execute an export datapump consistent in a SCN point, example:

```
SQL> column current_scn format 99999999999999999999
SQL> select current_scn from v$database;
```

This query gives the current SCN in a point in the time and will use this value for executing the export datapump.

```
s12rac01> expdp userid= system/oracle directory=DIR_EXP
tables=EDYFICAR.MOVIMIENTOS_CONTABLES,
EDYFICAR.CO_BALANCE_RESULTADOS,EDYFICAR.BITACORA,
EDYFICAR.HISTORIA_VISTA,EDYFICAR.HISTORY
dumpfile=edyficar1.dmp,edyficar2.dmp parallel=2
flashback_scn=<#_current_scn> exclude=INDEX,STATISTIC
```

We must include tables that will participate in the replication.

b) Target Database:

We will import the dmp generated by the export datapump.

```
sp12rac01> impdp userid=system/oracle directory=DIR_IMP full=y
dumpfile=edyficar1.dmp,edyficar2.dmp parallel=2
```

c) Synchronization:

At this point the initial load is finished, so we must begin the replicat process in target database.

```
GGSCI> start replicat rep1c, aftercsn <#_current_scn>
```

Note: Our target database has information in real time received from source database. Then we must stop the Oracle GoldenGate process when we want to enable the primary services in the new database, simply with the command:

```
GGSCI> stop *
GGSCI> stop manager
```



Don't forget to enable the triggers, constraints, primary key and foreign keys again, those which were disabled during the synchronization phase.

V. Replication Operations DDL (Source / Target).

In my implementation of the migration project we didn't decide to replicate DDL operations because every structure modification in production was suspended until migration was done so we didn't need to replicate anything at this level. The following lines show the configuration suggested getting this done. :

If the Oracle Database version is older than version 11gR1, we must disable the recycle bin.

```
SQL> alter system set recyclebin='off';
```

We must execute the following steps:

```
SQL> grant execute on utl_file to gg;
```

Modify the file GLOBALS stored in the stage of Oracle GoldenGate and add the following line:
GGSCHEMA gg

```
sqlplus / as sysdba
SQL> @marker_setup.sql
SQL> @ddl_setup.sql
SQL> @role_setup.sql
SQL> grant ggs_ggsuser_role to gg;
SQL> @ddl_enable.sql
```

These scripts also are found in the stage of Oracle GoldenGate.

*.- gg is the username of database that was created in the step II.a.

The extractor process must have the following line of configuration (ddl include all):

```
GGSCI> edit param extlc
extract extla
userid gg, password oracle
tranlogoptions asmuser sys@ASM, asmpassword oracle
extrail dirdat/tc
ddl include all
table EDYFICAR.*;
tableexclude EDYFICAR.MOVIMIENTOS;
```

The replicat process must have the following line of configuration:

```
GGSCI> edit param extlc
...
ddl include all
```



ddlerror default ignore retryop

Conclusions

The migration of this database was successful because we were able to migrate 2TB of information from a Linux platform to AIX in the interval of an acceptable time for our client. The actions and worked done at post-migration were minimal; allowing us to do migrations with zero-downtime. Also, Oracle GoldenGate provides a console where we can monitor our replication process and it let us know if a process is down or not synchronized, which in our case was very useful.

About the author

Francisco Riccio Chávez is an IBM Senior IT Specialist recognized at Oracle ACE since 2011. His current Oracle Certification levels are: OCP 11i Applications Database Administrator, OCP Oracle Database 9i/10g/11g, OCA OAS 10g, OCA Oracle PLSQL Developer, Oracle RAC 11g and Grid Infrastructure Administrator Expert, Oracle Database 10g RAC Certified Administrator Expert, Oracle Database 11g Performance Tuning Certified Expert, Managing Oracle Database 10g on Linux, Oracle Database SQL Expert, Oracle Real Application Clusters 11g Essentials and Oracle Database 11g Essentials for Implementers. He also teaches Oracle's certification at Peruvian universities. He works at IBM Global Technology Services SSO Peru Multi country SSA LA with the role SSO Data Management Service Line Owner for SSA.

Please send an e-mail to the IBM Oracle International Competency Center at ibmoracle@us.ibm.com or to the author at friccio@pe.ibm.com if there is any question about this topic.