GoldenGate for MySQL to Oracle

Objective

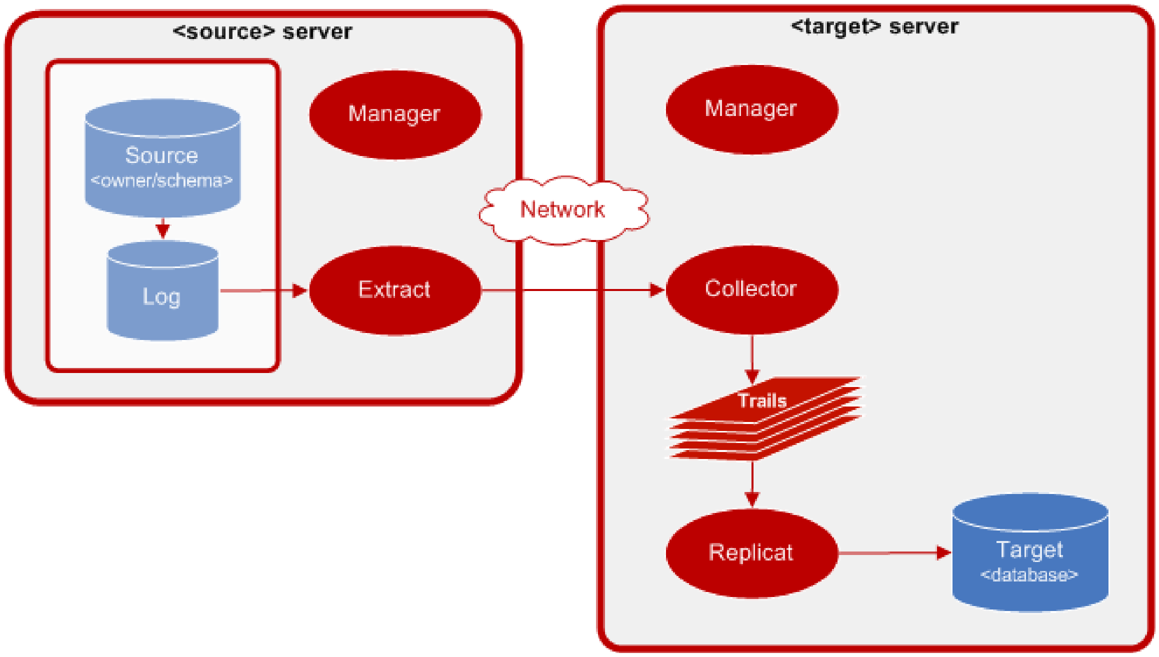
Upon completion of this lesson, you will be able to keep two databases synchronized, in this case MySQL to Oracle.

During this lesson, you will learn how to:

- Prepare your environment to configure the GoldenGate processes
- Configure and execute the initial data load
- Configure and start the change capture of database operations
- Configure and start the change delivery of database operations

MySQL to MySQL configuration

The following diagram illustrates GoldenGate a configuration with MySQL source data being replicated to an Oracle target database.



Overview of Tasks

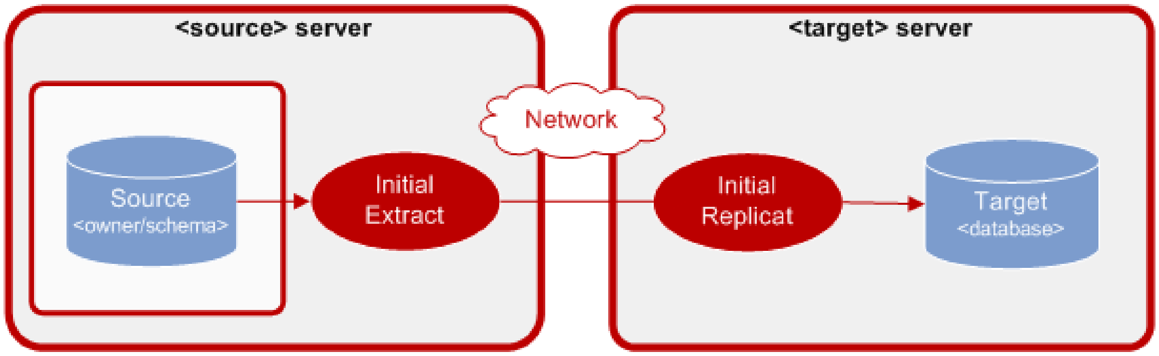
Prepare the Environment

Oracle GoldenGate must be installed on both the source and target systems. The installation includes a sample database and scripts to generate initial data as well as subsequent update operations. The source and target tables are created and loaded with initial data. The

GoldenGate Manager processes are also started so that other processes may be configured and started. And finally, source definitions are generated and transferred to the target system.

Initial Data Load

To initially load data across heterogeneous databases, GoldenGate provides the ability to perform initial data synchronization while your application remains active. This tutorial demonstrates using Extract to pull data from the source files and send it directly to Replicat on the target system.



Configure Change Capture

For log-based MySQL capture, the capture process is configured to capture change data directly from the MySQL logs and store the changes in queues known as GoldenGate remote trails.

Configure Change Delivery

Once the tables have been initially loaded with data, the Replicat is configured to deliver the captured change data into the target database.

Prerequisites

The prerequisites for this tutorial include the following.

- Oracle GoldenGate 11*g* for MySQL installed in the source **<install location>.**

  Note: Make sure you install Oracle GoldenGate for MySQL with an OS account that is in the same OS group as the MySQL Server OS account
- Oracle GoldenGate 11*g* for Oracle installed in the target **<install location>.**
- The MySQL **<database>** created in the source environment.
- The Oracle **<database>** created in the target environment.

Exercise 1.
Prepare the Environment

Objective

The goals of this exercise are to:

- Configure and start the Manager processes
- Create and load practice data to the source and target tables

Prepare your MySQL source environment

1. Configure the Manager process on the source

On the **<source>** system, create the Manager parameter file and specify the port it should use.

- Create the Manager parameter file.

```
Shell> cd <install location>
Shell> ggsci
GGSCI> EDIT PARAMS MGR
```

- Use the editor to assign a port.

```
--GoldenGate Manager parameter file
PORT <port>
```

- Start the Manager.

```
GGSCI> START MGR
```

- Verify that the Manager has started.

```
GGSCI> INFO MGR
```

2. Set MySQL server configuration parameters

On the **<source>** system, before starting the MySQL service, set the **MYSQL_HOME** environment variable to point to the installation location (for example **/usr/bin**) of MySQL, and set the **LD_LIBRARY_PATH** environment variable to add the installation location of Oracle GoldenGate for MySQL.

```
Shell>export MYSQL_HOME=<MySQL bin location>
Shell>export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<installation location
of Oracle GoldenGate for MySQL>
```

Edit the MySQL server configuration file. If the file does not yet exist, create the file **/etc/my.cnf**.

```
Shell>su - root
Shell>vi /etc/my.cnf
```

Insert the following into the file:

```
[mysqld]
log-bin=/var/lib/mysql/<hostname>-bin
max_binlog_size=4096
binlog_format=row
socket=/tmp/mysql.sock

[client]
socket=/tmp/mysql.sock
```

Start the MySQL service.

```
Shell>su - root
Shell>service start mysql
```

Note: Oracle Golden Gate 11*g* for MySQL requires the MySQL socket file to reside in **/tmp**. Some MySQL RPM installations default to placing **mysql.sock** in **/var/lib/mysql**. The settings in **my.cnf** override the default socket file location.

Verify that the MySQL ODBC connector is installed. As the **roo**t Linux user run the following command:

```
Shell>rpm -qa *odbc*
mysql-connector-odbc-3.51.26r1127-1.el5
```

If it doesn't already exist, create the **/usr/local/etc/odbc.ini** file. Add the following into the file:

```
[ODBC Data Sources]
<source db> = MyODBC 3.51 Driver DSN

[<source db>]
Driver = /usr/lib/libmyodbc3.so
Description = Connector/ODBC 3.51 Driver DSN
Server = localhost
Port = 3306
User = <source db login>
Password = <source db password>
Database = <source db name>
Option = 3
Socket = /tmp/mysql.sock
```

3. Create the source tables and load the initial data.

Create and populate the TCUSTMER and TCUSTORD tables by running the **demo_mysql_create.sql** and **demo_mysql_insert.sql** files found in the install directory.

Execute the following commands on the **<source>** system.

Note: To avoid confusion with the **<** directive, the variables have been placed in brackets for the mysql commands that follow.

```
Shell> cd {install location}
Shell> mysql {database} -u{login} -p{password} <
demo_mysql_create.sql
Shell> mysql {database} -u{login} -p{password} <
demo_mysql_insert.sql
```

Verify the results:

```
Shell> mysql {database} -u{login} -p
Enter Password: {password}

mysql> describe TCUSTMER;
mysql> describe TCUSTORD;
mysql> select * from TCUSTMER;
mysql> select * from TCUSTORD;
mysql> exit
```

Create source definitions

4. Configure source definition generator

Execute the following commands on the **<source>** system.

- Bring up the DEFGEN parameter file in the editor.

  ```
  GGSCI> EDIT PARAM DEFGEN
  ```

- Type in the parameters shown below.

```
DEFSFILE ./dirdef/source.def, PURGE
SOURCEDB <dsn>, USERID <login>, PASSWORD <password>
TABLE <owner/schema>.TCUSTMER;
TABLE <owner/schema>.TCUSTORD;
```

- Exit from GGSCI.

  ```
  GGSCI> EXIT
  ```

5. Execute the source definition generator

Execute the source definition generator from the Windows command prompt.

```
Shell> defgen paramfile .\dirprm\defgen.prm
```

6. Transfer the source definition file to the target system

Execute the following commands on the **<source>** system to transfer the definition file to the **<target>** system.

The **cd** command should be set to the **<target>** location and the **lcd** to the **<source>** location.

```
Shell> ftp <target>
Name (<target>:<login>): <login>
Password: <password>
ftp> ascii
ftp> cd <install loc>/dirdef
ftp> lcd <install loc>/dirdef
ftp> put source.def
ftp> bye
```

Note: To avoid overlaying existing definition files, always **put** the file as **<db type>.def** instead of source.def (e.g. MSSQL.def).

Prepare your target Oracle environment

Configure the Manager

1. Configure the Manager process on the target system

Execute the following commands on the **<target>** system.

- Start the command interface

```
Shell> cd <install location>
Shell> ggsci
```

- Specify the port that the Manager should use.

```
GGSCI> EDIT PARAMS MGR
```

```
-- GoldenGate Manager Parameter file
PORT <port>
```

- Start Manager

```
GGSCI> START MANAGER
```

Verify the results:

```
GGSCI> INFO MANAGER
```

Create the tables

2. Create target Oracle tables

Execute the following commands on the **`<target>`** system to run the script that creates the tables.

```
Shell> cd <install location>
Shell> sqlplus <login>/<password>
SQL> @demo_ora_create
```

Verify the results:

```
SQL> desc tcustmer
SQL> desc tcustord
SQL> exit
```

Exercise 2.
Initial Data Load Using Direct Load Method

● ● ● ● ● ● ● ● ● ● ● ● ● ● ●

Objective

The goals of this exercise are to:

- Configure and execute the initial data load capture
- Configure and execute the initial data delivery

Configure initial load

1. Add the initial load capture batch task group

Execute the following commands on the **<source>** system to add an Extract process called EINI<unique id>[1].

- Execute the following command in GGSCI to create the batch task.

```
Shell> cd <install location>
Shell> ggsci
GGSCI> ADD EXTRACT EINI<unique id>, SOURCEISTABLE
```

- Verify the results:

```
GGSCI> INFO EXTRACT *, TASKS
```

2. Configure the initial load capture parameter file

Execute the following command in GGSCI to open the editor on the **<source>** system.

```
GGSCI> EDIT PARAMS EINI<unique id>
```

Add the following lines to the parameter file.

```
--
-- GoldenGate Initial Data Capture
-- for TCUSTMER and TCUSTORD
--
EXTRACT EINI<unique id>
```

[11] The process names used in tutorials, for example EINIBD, are made up of 1) one character for the GoldenGate process (E for Extract, R for Replicat); 2) three or four to describe the process type (INI for initial data load, ORA for capture from or delivery to an Oracle database, etc.) and 3) two characters to create a unique identifier (usually your initials).

```
SOURCEDB <source db>, USERID <login>, PASSWORD <password>
RMTHOST <target>, MGRPORT <port>
RMTTASK REPLICAT, GROUP RINI<unique id>
TABLE <source db>.TCUSTMER;
TABLE <source db>.TCUSTORD;
```

Configure initial load delivery

3. Add the initial load delivery batch task group

Execute the following commands on the **<target>** system.

```
GGSCI> ADD REPLICAT RINI<unique id>, SPECIALRUN
```

Verify the results:

```
GGSCI> INFO REPLICAT *, TASKS
```

4. Configure the initial load delivery parameter file

Execute the following commands on the **<target>** system.

```
GGSCI> EDIT PARAMS RINI<unique id>
```

```
--
-- Change Delivery parameter file for
-- TCUSTMER and TCUSTORD changes
--
REPLICAT RINI<unique id>
USERID <login>, PASSWORD <password>
SOURCEDEFS ./dirdef/source.def
DISCARDFILE ./dirrpt/RINI<unique id>.dsc, PURGE
MAP "<source db>.TCUSTMER", TARGET <SCHEMA>.TCUSTMER;
MAP "<source db>.TCUSTORD", TARGET <SCHEMA>.TCUSTORD;
```

Note: Remember to use <db type>.def if you renamed the source.def when you transferred it to the <target>.

Note: Enter the MySQL source db and table names in the correct case and place double quotes around them for delivering changes from a case sensitive system such as MySQL.

5. Execute the initial load process

Execute the following commands on the **<source>** system.

```
GGSCI> START EXTRACT EINI<unique id>
```

Verify the results:

Execute the following commands on the **<target>** system.

```
GGSCI> VIEW REPORT RINI<unique id>
```

Exercise 3.

Configure Change Capture

Objective

The goals of this exercise are to:

- Configure and add the Extract process that will capture changes.
- Add the trail that will store the changes.
- Start the Extract process.

Configure change capture

1. Add the Extract group

Execute the following command on the **<source>** system to add an Extract group named EORA<unique id>.

```
GGSCI> ADD EXTRACT EMSQ<unique id>, TRANLOG, BEGIN NOW
```

Verify the results:

```
GGSCI> INFO EXTRACT EMSQ<unique id>
```

2. Create the Extract parameter file

Execute the following commands on the **<source>** system.

```
GGSCI> EDIT PARAM EMSQ<unique id>
```

```
--
-- Change Capture parameter file to capture
-- TCUSTMER and TCUSTORD Changes
--
EXTRACT EMSQ<unique id>
DBOPTIONS HOST <target host>, CONNECTIONPORT <MySQL server port>
SOURCEDB <source db>, USERID <login>, PASSWORD <password>
RMTHOST <target>, MGRPORT <port>
RMTTRAIL ./dirdat/<trail id>
TRANLOGOPTIONS ALTLOGDEST <location of MySQL log files as defined
in the log-bin parameter in my.cnf>/<hostname>-bin.index
TABLE <source db>.TCUSTMER;
TABLE <source db>.TCUSTORD;
```

Note: Record the two characters selected for your `<trail id>`: ______. You will need this in the next step and when you set up the Replicat.

3. Define the GoldenGate trail

Execute the following command on the **`<source>`** to add the trail that will store the changes on the target.

```
GGSCI> ADD RMTTRAIL ./dirdat/<trail id>, EXTRACT EMSQ<unique id>,
MEGABYTES 5
```

Verify the results:

```
GGSCI> INFO RMTTRAIL *
```

4. Start the capture process

```
GGSCI> START EXTRACT EMSQ<unique id>
```

Verify the results:

```
GGSCI> INFO EXTRACT EMSQ<unique id>, DETAIL
GGSCI> VIEW REPORT EMSQ<unique id>
```

Discussion points

1. Identifying a remote system

What parameter is used to identify the remote target system?

____

2. Sizing the GoldenGate trail

Where do you set how large a GoldenGate trail file may get before it rolls to the next file? What option do you use?

____

____

Exercise 4.
Configure Change Delivery

Objective

The goals of this exercise are to:

- Set up the checkpoint table on the target system.
- Create a named group that includes the Replicat process and the checkpoint tables.
- Configure the Replicat group by adding parameters.
- Start the Replicat group.

Set up the checkpoint table

1. Create a GLOBALS file on the target system

Execute the following commands on the **<target>** system.

○ Create and edit the **GLOBALS** parameter file to add the checkpoint table.

```
Shell> cd <install location>
Shell> ggsci
GGSCI> EDIT PARAMS ./GLOBALS
```

In the text editor, type:

```
CHECKPOINTTABLE <owner/schema>.GGS_CHECKPOINT
```

○ Record the checkpoint table owner and name, then save and close the file.

Table owner ____________________ name ____________________

Note: You could name the table anything you want, but for training purposes we are using **GGS_CHECKPOINT**.

○ Verify that the **GLOBALS** file was created in the root GoldenGate directory, and remove any file extension that was added.

2. Activate the GLOBALS parameters

For the **GLOBALS** configuration to take effect, you must exit the session in which the changes were made. Execute the following command to exit **GGSCI**.

```
GGSCI> EXIT
```

3. Add a Replicat checkpoint table

On the **<target>** system, execute the following commands in **GGSCI**:

```
Shell> cd <install location>
Shell> ggsci
GGSCI> DBLOGIN USERID <login>, PASSWORD <password>
GGSCI> ADD CHECKPOINTTABLE
```

Configure Change Delivery

4. Add the Replicat group

Execute the following command on the **<target>** system to add a delivery group named RORA<unique id>.

```
GGSCI> ADD REPLICAT RORA<unique id>, EXTTRAIL ./dirdat/<trail id>
```

Note: Refer to your Extract set up for the correct two-character <trail id>.

5. Create Replicat parameter file

Execute the following commands on the **<target>** system to bring up the parameter file in the editor.

```
GGSCI> EDIT PARAM RORA<unique id>
```

Type in the following parameters

```
--
-- Change Delivery parameter file to apply
-- TCUSTMER and TCUSTORD Changes
--
REPLICAT RORA<unique id>
USERID <login>, PASSWORD <password>
HANDLECOLLISIONS
SOURCEDEFS ./dirdef/source.def
DISCARDFILE ./dirrpt/RORA<unique id>.DSC, PURGE
MAP "<source db>.TCUSTMER", TARGET <SCHEMA>.TCUSTMER;
MAP "<source db>.TCUSTORD", TARGET <SCHEMA>.TCUSTORD;
```

Note: Remember to use <db type>.def if you renamed the source.def when you transferred it to the <target>.

Note: Enter the MySQL source db and table names in the correct case and place double quotes around them for delivering changes from a case sensitive system such as MySQL.

6. Start the Replicat process

```
GGSCI> START REPLICAT RORA<unique id>
```

Verify the results:

```
GGSCI> INFO REPLICAT RORA<unique id>
```

Discussion points

1. When to use HANDLECOLLISIONS

When would you use HANDLECOLLISIONS? What does it do?

____

____

2. What information is supplied by SOURCEDEFS?

____

____

3. What is the purpose of the DISCARDFILE?

____

____

Exercise 5.
Generate Activity and Verify Results

Objective

The goals of this exercise are to:

- Execute miscellaneous update, insert, and delete operations on the source system.
- Verify the delivery of the changes to the target
- Turn off the error handling used for initial load.

Generate database operations

1. Execute miscellaneous update, insert, and delete operations

Execute the following commands on the **<source>** system.

```
Shell> cd {install location}
Shell> mysql {database} -u{login} -p{password} < demo_mysql_misc.sql
```

Verify change capture and delivery

2. Verify results on the source system

Execute the following commands on the **<source>** system.

```
Shell> mysql {database} -u{login} -p
Enter Password: {password}

mysql> select * from TCUSTMER;
mysql> select * from TCUSTORD;
mysql> exit

Shell> ggsci
GGSCI> SEND EXTRACT EMSQ<unique id>, REPORT
GGSCI> VIEW REPORT EMSQ<unique id>
```

3. Verify your results on the target system

Execute the following commands on the **<target>** system to verify the target data.

```
Shell> cd <install location>
Shell> mysql <database> -u<login> -p
```

```
Enter Password: <password>

mysql> select * from TCUSTMER;
mysql> select * from TCUSTORD;
mysql> exit

Shell> ggsci
GGSCI> SEND REPLICAT RORA<unique id>, REPORT
GGSCI> VIEW REPORT RORA<unique id>
```

Turn off error handling

4. Turn off initial load error handling for the running delivery process

```
GGSCI> SEND REPLICAT RORA<unique id>, NOHANDLECOLLISIONS
```

5. Remove initial load error handling from the parameter file

```
GGSCI> EDIT PARAMS RORA<unique id>
```

Remove the HANDLECOLLISIONS parameter.