# GoldenGate for MySQL to MySQL

## Objective

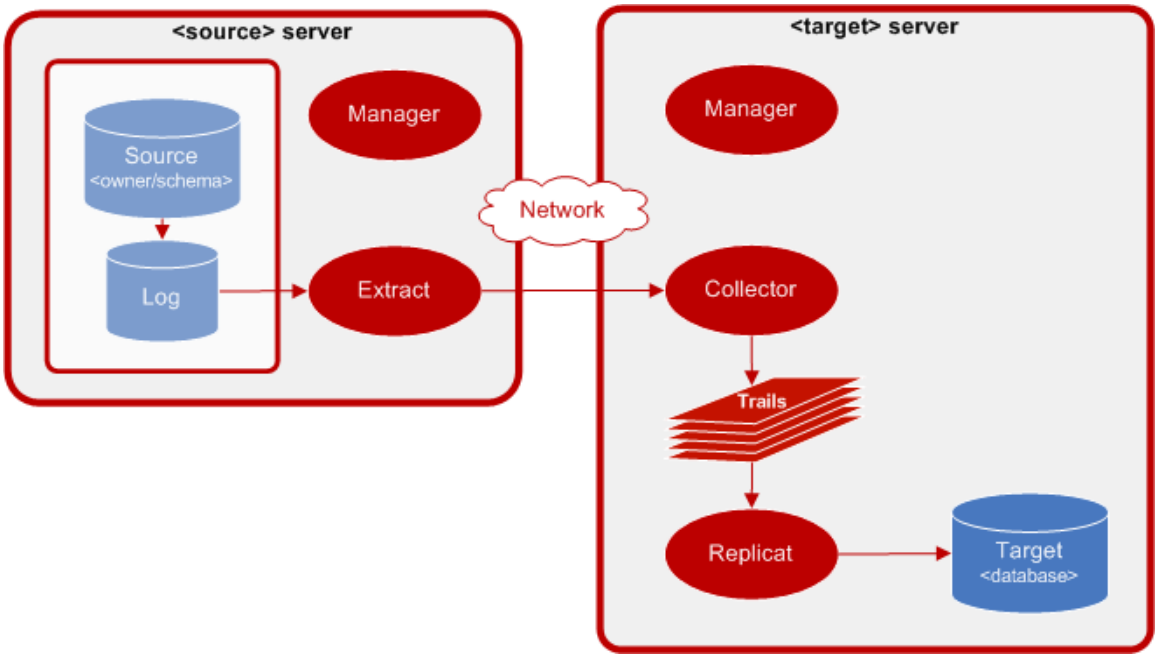
Upon completion of this lesson, you will be able to keep two databases synchronized, in this case MySQL to MySQL.

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 a MySQL 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.

### 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 both the source **<install location>** and the target **<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

- The MySQL **<database>** created in the source and 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
```

## Prepare the MySQL target environment

### 1. Configure the Manager process

Execute the following command 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
```

## 2. Set MySQL server configuration parameters

On the **<target>** 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
```

## 3. Create and populate target tables

Execute the following commands on the **<target>** 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
```

Verify that the MySQL ODBC connector is installed. As the **root** 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 under the **[ODBC Data Sources]** section:

```
<target db> = MyODBC 3.51 Driver DSN

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

## Exercise 2.
# 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 3.
# 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 <target database>.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 **ggschkpt**.

- 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 SOURCEDB <database>, USERID root, 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 RMSQ<unique id>.

```
GGSCI> ADD REPLICAT RMSQ<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 RMSQ<unique id>
```

Type in the following parameters

```
--
-- Change Delivery parameter file to apply
-- TCUSTMER and TCUSTORD Changes
--
REPLICAT RMSQ<unique id>
DBOPTIONS HOST <target host>, CONNECTIONPORT <MySQL server port>
TARGETDB <target db>, USERID <root login>, PASSWORD <password>
HANDLECOLLISIONS
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/RMSQ<unique id>.DSC, PURGE
MAP <source db>.TCUSTMER, TARGET <target db>.TCUSTMER;
MAP <source db>.TCUSTORD, TARGET <target db>.TCUSTORD;
```

### 6. Start the Replicat process

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

Verify the results:

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

## Discussion points

Search in the *Windows/UNIX Reference Guide* for the information on the following questions.

### 1. When to use HANDLECOLLISIONS

When would you use HANDLECOLLISIONS? What does it do?

_____

_____

### 2. What information is supplied by ASSUMETARGETDEFS

_____

_____

## Exercise 4.
# 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 RMSQ<unique id>, REPORT
GGSCI> VIEW REPORT RMSQ<unique id>
```

## Turn off error handling

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

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

### 5. Remove initial load error handling from the parameter file

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

Remove the HANDLECOLLISIONS parameter.