

GoldenGate for Oracle to MySQL

Objective

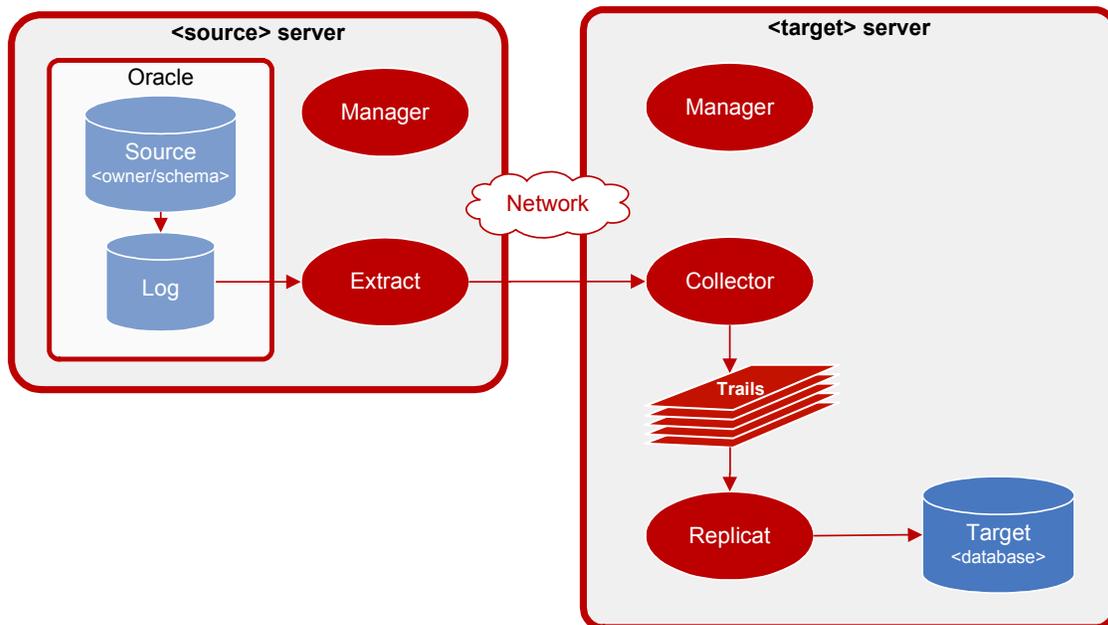
Upon completion of this lesson, you will be able to keep two heterogeneous databases synchronized, in this case Oracle to MySQL.

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

Oracle to MySQL configuration

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



Overview of Tasks

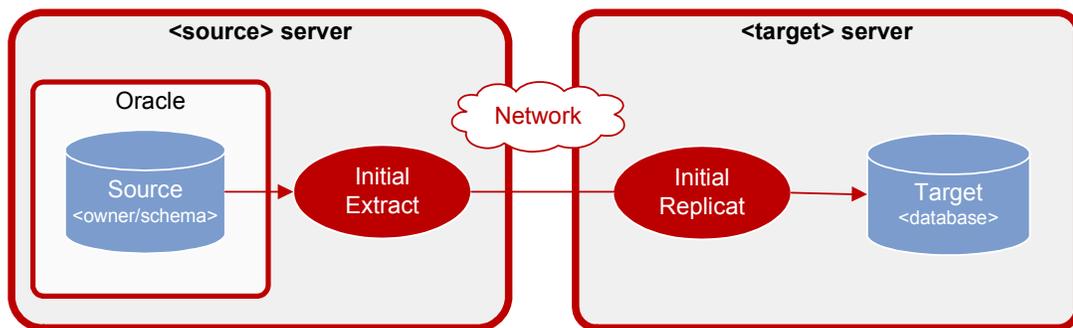
Prepare the Environment

In order to execute this lesson, the GoldenGate application 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 lesson 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 Oracle capture, the capture process is configured to capture change data directly from the Oracle online redo logs or archive 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 lab include the following.

- GoldenGate installed in both the source <install location> and the target <install location>.
- The MySQL <database> defined 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 Oracle tables
- Add supplemental logging

Prepare your Oracle 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. Create the source tables and load the initial data.

Using SQL*Plus, create and populate the TCUSTOMER and TCUSTORD tables by running the **demo_ora_create.sql** and **demo_ora_insert.sql** files found in the install directory.

Execute the following commands on the <source> system.

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



Verify the results:

```
SQL> select * from tcustmer;  
SQL> select * from tcustord;  
SQL> exit
```

3. Add supplemental logging

Using GGSCI, log in to the database on the <source> and turn on supplemental logging for the TCUSTMER and TCUSTORD tables.

```
Shell> ggsci  
GGSCI> DBLOGIN USERID <login>, PASSWORD <password>  
GGSCI> ADD TRANDATA <owner/schema>.TCUSTMER  
GGSCI> ADD TRANDATA <owner/schema>.TCUSTORD
```

Verify that supplemental logging has been turned on for these tables.

```
GGSCI> INFO TRANDATA <owner/schema>.TCUST*
```

Prepare the MySQL target environment

1. Configure the Manager process

Execute the following command on the <target> Teradata 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. Create target files

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
```

Verify the results:

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

```
mysql> describe TCUSTMER;
mysql> describe TCUSTORD;
mysql> exit
```




```
RMTTASK REPLICAT, GROUP RINI<unique id>
TABLE <owner/schema>.TCUSTMER;
TABLE <owner/schema>.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>
TARGETDB <database>, USERID <login>, PASSWORD <password>
SOURCEDEFS ./dirdef/source.def
DISCARDFILE ./dirrpt/RINI<unique id>.dsc, PURGE
MAP <owner/schema>/TCUSTMER, TARGET <database>.TCUSTMER;
MAP <owner/schema>/TCUSTORD, TARGET <database>.TCUSTORD;
```

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>
```


Note: When Oracle Automatic Storage Management (ASM) is in use, the TRANLOGOPTIONS ASMUSER and ASMPASSWORD must be set in the Extract parameter file. For more information refer to the *GoldenGate for Windows & UNIX Administrator* and *Reference* manuals.

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 EORA<unique id>,  
MEGABYTES 5
```

Verify the results:

```
GGSCI> INFO RMTTRAIL *
```

4. Start the capture process

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

Verify the results:

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

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 <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 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>
TARGETDB <database>, USERID <login>, PASSWORD <password>
HANDLECOLLISIONS
SOURCEDEFS ./dirdef/source.def
DISCARDFILE ./dirrpt/RMSQ<unique id>.DSC, PURGE
MAP <owner/schema>/TCUSTMER, TARGET <database>.tcustmer;
MAP <owner/schema>/TCUSTORD, TARGET <database>.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 SOURCEDEFS



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> sqlplus <login>/<password>
SQL> @demo_ora_misc
```

Verify change capture and delivery

2. Verify results on the source system

Execute the following commands on the <source> system.

```
SQL> select * from tcustmer;
SQL> select * from tcustord;
SQL> exit
```

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



