Oracle GoldenGate for Oracle to Oracle

Objective

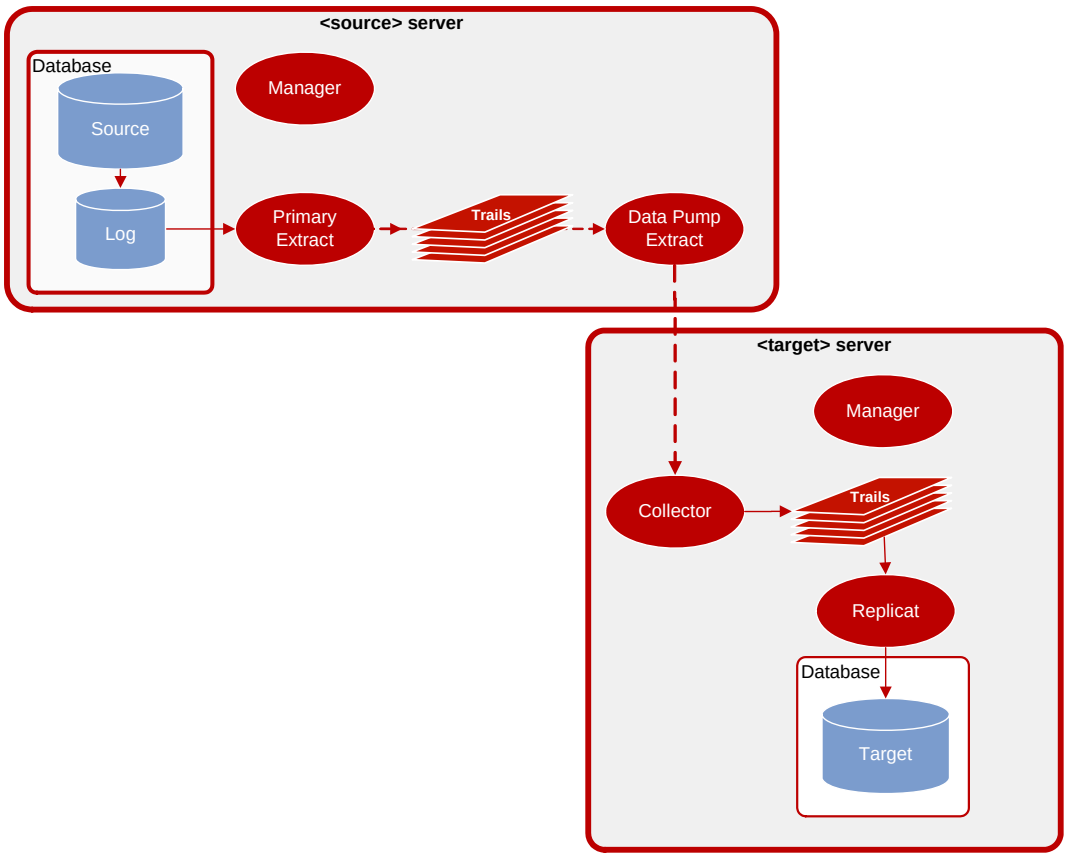
Upon completion of this lesson, you will be able to keep two Oracle databases synchronized.

During this lesson, you will:

- Prepare the database and the Oracle GoldenGate environment.
- Configure and execute the initial data load process
- Configure and start the change capture of database operations.
- Configure and start the change delivery process.

Oracle to Oracle configuration

The following diagram illustrates Oracle GoldenGate installed on two systems connected by TCP/IP – one containing the source data and the other the replicated data.



Overview of tasks

Prepare the Environment

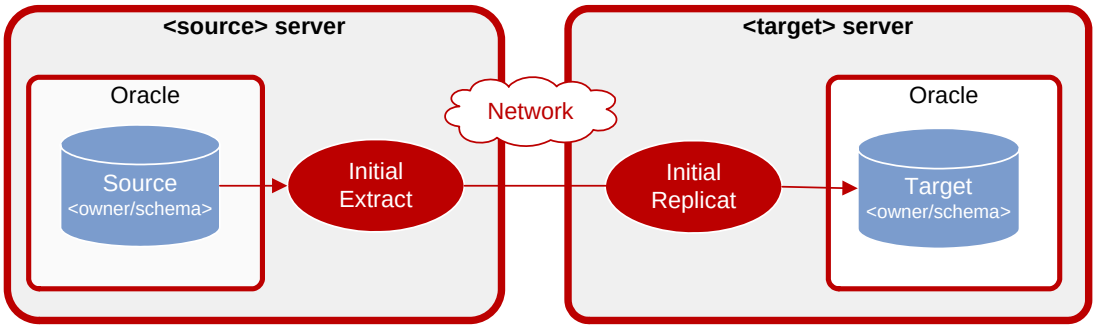
In order to execute this lesson, the GoldenGate application must be installed on both the source and target systems. The installation includes a sample database and scripts to generate initial data as well as subsequent update operations. The source and target tables are created and loaded with initial data. The GoldenGate Manager processes are also started so that other processes may be configured and started.

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 Initial Data Load

To initially load data, there are techniques such as Backup/Restore or Export/Import, both of which have pros and cons. Or you may use the GoldenGate application to perform your initial data load while the application remains active. This lesson demonstrates using Extract to pull data from the source tables and sending the data directly to the Replicat on the target system.



Configure Change Delivery

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

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 TCUSTMER 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 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.
Configure Change Capture using a Data Pump

Objective

The goals of this exercise are to:

- Configure and add the Extract process that will capture changes.
- Add the local trail that will store these changes.
- Configure and add a data pump Extract to read the local trail and create a remote trail on the target.
- Add the remote trail.
- Start the two Extract processes.

Configure the primary Extract

1. Create the Extract parameter file

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

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

```
--
-- Change Capture parameter file to capture
-- TCUSTMER and TCUSTORD changes
--
EXTRACT EORA<unique id>
USERID <login>, PASSWORD <password>
EXTTRAIL ./dirdat/<local trail id>
TABLE <owner/schema>.TCUSTMER;
TABLE <owner/schema>.TCUSTORD;
```

Notes:

1. Record the two characters selected for your <local trail id>: _____. You will need this information to set up the data pump Extract.

2. 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 *Oracle GoldenGate Reference* manual.

2. Add the Extract group

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

```
GGSCI> ADD EXTRACT EORA<unique id>, TRANLOG, BEGIN NOW, THREADS
<instances>
```

Note: In a non-RAC environment, the THREADS parameter can be omitted or the **THREADS** <instances> can be set to **1**.

Verify results:

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

3. Define the GoldenGate local trail

Execute the following commands on the **<source>** system to add the local trail declared in the EORA<unique id> Extract parameters.

```
GGSCI> ADD EXTTRAIL./dirdat/<local trail id>, EXTRACT EORA<unique
id>, MEGABYTES 5
```

Configure the data pump

4. Create the data pump parameter file.

Execute the following commands on the **<source>** system to define a data pump Extract named EPMP<unique id> to pull data from the local GoldenGate trail and route these changes to GoldenGate on the target.

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

```
--
-- Data Pump parameter file to read the local
-- trail of TCUSTMER and TCUSTORD changes
--
EXTRACT EPMP<unique id>
PASSTHRU
RMTHOST <target>, MGRPORT <port>
RMTTRAIL ./dirdat/<remote trail id>
TABLE <owner/schema>.TCUSTMER;
TABLE <owner/schema>.TCUSTORD;
```

Note: Record the two characters selected for your <remote trail id>: _____. You will need this information later when you define the remote trail that is read by Replicat.

5. Add the data pump Extract group

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

```
GGSCI> ADD EXTRACT EPMP<unique id>, EXTTRAILSOURCE ./dirdat/<local
trail id>
```

Verify results:

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

6. Define the GoldenGate remote trail

Execute the following commands on the **<source>** system to add the remote trail declared in the EPMP<unique id>Extract parameters. This will be located on the target system.

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

Start the Extract processes

7. Start the primary Extract process

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

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

Verify the results:

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

8. Start the data pump Extract process

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

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

Verify the results:

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

Discussion points

1. Using PASSTHRU

What is the function of the **PASSTHRU** parameter?

_____

_____

2. Identifying the local extract trail

A data pump moves data between an Extract and a Replicat. How does it know to read the local extract trail instead of a transaction log?

_____

_____

3. Identifying a remote system

What parameters are used to identify the remote target system?

_____

_____

4. The advantage of a data pump

What is the advantage of using a data pump when updating a remote system?

_____

_____

Exercise 3.
Initial Data Load using Direct Load Method

Objective

The goals of this exercise are to:

- Configure a task to load the initial data from a source table
- Configure the delivery of the data to the target
- Execute the initial load of data.

Initial data capture

1. Add the initial load capture batch task group

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

```
GGSCI> ADD EXTRACT EINI<unique id>, SOURCEISTABLE
```

Verify the result:

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

2. Configure the initial load capture parameter file

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

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

```
--
-- GoldenGate Initial Data Capture
-- for TCUSTMER and TCUSTORD
--
EXTRACT EINI<unique id>
USERID <userid>, PASSWORD “<password>”
RMTHOST <target>, MGRPORT <port>
RMTTASK REPLICAT, GROUP RINI<unique id>
TABLE <owner/schema>.TCUSTMER;
TABLE <owner/schema>.TCUSTORD;
```

[1] The process names used in lab exercises, 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)..

Initial data delivery

3. Add the initial load delivery batch task

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>
```

```
--
-- GoldenGate Initial Load Delivery
--
REPLICAT RINI<unique id>
ASSUMETARGETDEFS
USERID <login>, PASSWORD <password>
DISCARDFILE ./dirrpt/RINI<unique id>.dsc, PURGE
MAP <owner/schema>.*, TARGET <owner/schema>.*;
```

5. Execute the initial load process

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

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

Verify the results on the **<source>** system:

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

Verify the results on the **<target>** system:

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

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>.ggschkpt
```

❍ 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 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
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/RORA<unique id>.DSC, PURGE
MAP <owner/schema>.tcustmer, TARGET <owner/schema>.tcustmer;
MAP <owner/schema>.tcustord, TARGET <owner/schema>.tcustord;
```

6. Start the Replicat process

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

Verify the results:

```
GGSCI> INFO REPLICAT RORA<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. **When should you use ASSUMETARGETDEFS?**

   _____

   _____

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> 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> sqlplus <userid>/<password>
SQL> select * from tcustmer;
SQL> select * from tcustord;
SQL> 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.