GoldenGate for Oracle to Teradata

Objective

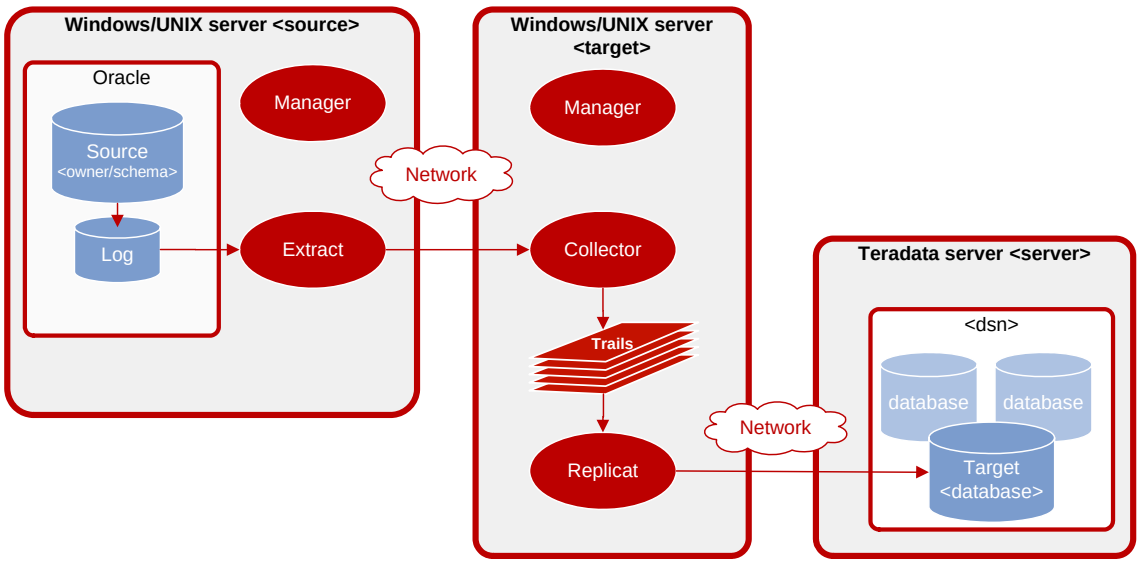
Upon completion of this lesson, you will be able to keep an Oracle database synchronized with a Teradata database using GoldenGate.

During this lesson, you will learn how to:

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

Oracle to Teradata configuration

The following diagram illustrates GoldenGate installed on either Windows or UNIX for both the Oracle source data and and the Teradata target data.



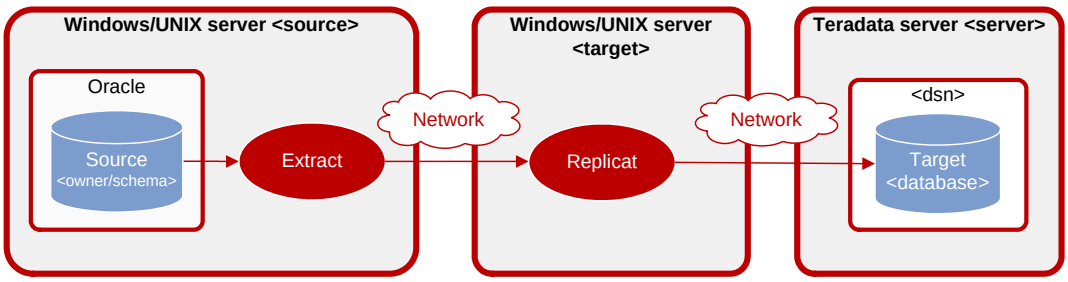
Overview of Tasks

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 Initial Data Load

Initial database synchronization is almost always required. To initially load data across heterogeneous databases, you have limited choices. GoldenGate provides the ability to perform initial data synchronization while your application remains active. This lesson demonstrates using the direct load method to capture data directly from the source tables and send the data directly to the delivery process 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 delivery process is configured to synchronize 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 sample tables and load data into source tables
- Add supplemental logging
- Generate source definitions file

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>
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*
```

4. Configure source definition generator

Execute the following commands on the **<source>** system to create the DEFGEN parameter file and add the listed parameters.

```
Shell> cd <install location>
Shell> ggsci
GGSCI> edit param defgen
```

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

```
GGSCI> exit
```

5. Execute the source definition generator

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

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

6. Transfer the source definition file to the target system

Execute the following commands to transfer the souce.def file to the **<target>** system.

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

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

Prepare the Teradata target environment

6. 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
```

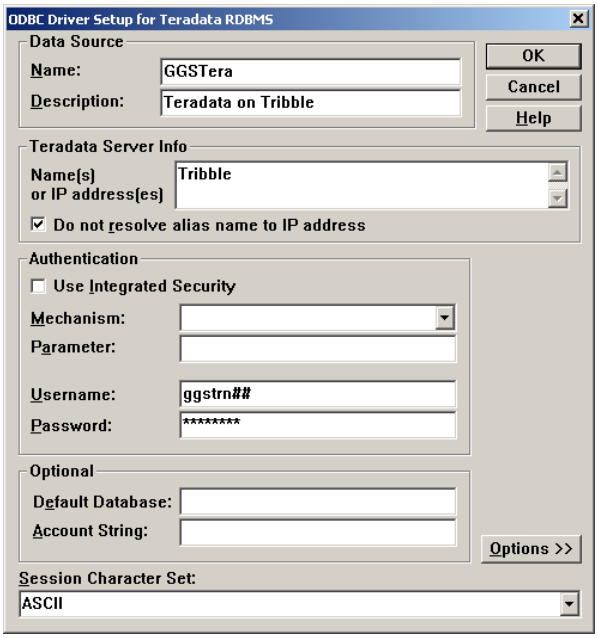
- Verify the results:

```
GGSCI> INFO MANAGER
```

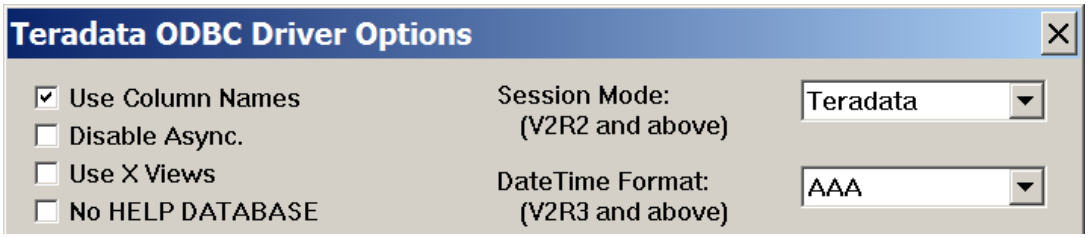
7. Create ODBC data source name

An ODBC system data source stores information about how to connect to the Teradata **<server>**. Execute the following commands on the **<target>** Teradata system.

Create your system DSN for Teradata with a name of **<dsn>** to connect to your target **<server>** name. These examples show a DSN named GGSTera and a server named **Tribble**, but these should be changed to match your names.



Click on the **Options** button. Select **Teradata** for the **Session Mode** and **AAA** for the **DateTime Format**.



8. Create target files

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

Edit the script, demo_tera_create.sql and change the **<database>** to the identifier for your target database.

```
create table <database>.tcustmer;
create table <database>.tcustord;
```

Using the Teradata command interface, BTEQ, logon and execute the script.

```
BTEQ> .logon <server>/<login>
         Password: <password>

BTEQ> .run file = ./demo_tera_create.sql;
```

Exercise 2.
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 and verify the initial load of data.

Configure initial load capture

1. Add the initial load capture batch task group

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

```
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 commands on the **<source>** system.

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

```
--
-- GoldenGate Initial Data Capture
-- for TCUSTMER and TCUSTORD
--
EXTRACT EINI<unique id>
USERID <login>, 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).

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 RINI<unque id>*, 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>
TARGETDB <dsn>, USERID <userid>, PASSWORD <password>
DISCARDFILE ./dirrpt/RINI<unique id>.DSC, PURGE
SOURCEDEFS ./dirdef/source.def
MAP <owner/schema>.TCUSTMER, TARGET <database>.TCUSTMER;
MAP <owner/schema>.TCUSTORD, TARGET <database>.TCUSTORD;
```

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

Execute the initial load

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 **<source>** system.

Wait 10 seconds ….

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

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

Verify the results:

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

2. Create the Extract parameter file

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

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

```
--
-- Change Capture parameter file to capture
-- TCUSTMER and TCUSTORD Changes
--
EXTRACT EORA<unique id>
USERID <login>, PASSWORD <password>
RMTHOST <target>, MGRPORT <port>
RMTTRAIL ./dirdat/<trail id>
TABLE <owner/schema>.TCUSTMER;
TABLE <owner/schema>.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.

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 *Oracle GoldenGate Reference* manual.

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?

_____

_____

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. Edit the GLOBALS file on the target system

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

❍ Edit the **GLOBALS** parameter file to add the checkpoint table.

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

In the text editor, add the following line:

```
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**.

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 <dsn>, 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 RTER<unique id>.

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

Type in the following parameters

```
--
-- Change Delivery parameter file to apply
-- TCUSTMER and TCUSTORD Changes
--
REPLICAT RTER<unique id>
TARGETDB <dsn>, USERID <login>, PASSWORD <password>
HANDLECOLLISIONS
SOURCEDEFS ./dirdef/source.def
DISCARDFILE ./dirrpt/RTER<unique id>.DSC, PURGE
MAP <owner/schema>.tcustmer, TARGET <database>.tcustmer;
MAP <owner/schema>.tcustord, TARGET <database>.tcustord;
```

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

6. Start the Replicat process

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

Verify the results:

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

Discussion points

1. When to use HANDLECOLLISIONS

For which stage of GoldenGate processing – change capture, initial data load, or change delivery – 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> 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 to logon to BTEQ on the **<target>** system and verify the target data.

```
Shell> bteq
BTEQ> .logon <server>/<login>
```

This will prompt you for the <password>. After that is entered you may query the database.

```
BTEQ> select * from <database>.tcustmer;
BTEQ> select * from <database>.tcustord;

Shell> cd <install location>
Shell> ggsci
GGSCI> SEND REPLICAT RTER<unique id>, REPORT
GGSCI> VIEW REPORT RTER<unique id>
```

Turn off error handling

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

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

5. Remove initial load error handling from the parameter file

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

Remove the HANDLECOLLISIONS parameter.