

Oracle® GoldenGate

Fundamentals of Oracle to Oracle

Version 10.4

October 2009

ORACLE®

Copyright © 1995, 2009 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

● ● ● ● ● ● ● ● ● ● ● ● ● ● ●

Lab Preparation	4
Installation of GoldenGate for Windows.....	6
Installation of GoldenGate for UNIX.....	9
Introduction to the Command Interface	11
GoldenGate for Oracle to Oracle.....	12
Prepare the Environment.....	14
Initial Data Load using Direct Load Method	17
Configure Change Capture	19
Configure Change Delivery	21
Generate Activity and Verify Results.....	24
Workshops.....	26
GoldenGate Documentation for Windows/UNIX.....	27
Using Commands on Windows/UNIX	32
Using Parameters.....	36
Using the Logdump Utility for Windows/UNIX	40
Using Encryption	45
Configure a Data Pump for Windows/UNIX.....	49
Initial Data Load using File to Replicat	56
Stored Procedures	59
Student Notes	69

Lab Preparation

Identify the following information that you will use to select versions for the installation and to substitute actual values for <items> during the lab exercises.

Your Oracle source environment

Operating system

Type (UNIX, Linux, Windows) _____

<userid> _____ <password> _____

Database

Database <login> _____ <password> _____

<owner/schema> _____ Number of RAC <instances> _____

GoldenGate

<source> system identifier _____

GoldenGate <install location> _____

Manager <port> _____

Student's <unique id> if applicable _____

<mgr service> name if on Windows _____

GoldenGate references

Checkpoint files	dirchk
GoldenGate trails	dirdat
Data definition files	dirdef
Parameter files	dirprm
Process status files	dirpcs
Report files	dirrpt
SQL script files	dirsql
Temporary files	dirtmp

Your Oracle target environment

Operating system

Type (Linux, UNIX, Windows) _____

<userid> _____ <password> _____

Database

Database <owner/schema> _____

Database <login> _____ <password> _____

GoldenGate

<target> system identifier (Host name or IP address) _____

<install location> for GoldenGate _____

Manager <port> _____

Student's <unique id> if applicable _____

<mgr service> name if on Windows _____

GoldenGate references

Checkpoint files	dirchk
GoldenGate trails	dirdat
Data definition files	dirdef
Parameter files	dirprm
Process status files	dirpcs
Report files	dirrpt
SQL script files	dirsql
Temporary files	dirtmp

Exercise 1.

Installation of GoldenGate for Windows

• • • • •

Objective

Upon completion of this lesson, you will have installed GoldenGate on a Windows platform.

The goals of this exercise are to:

- Find installation components and instructions.
- Find and download GoldenGate documentation.
- Install GoldenGate on Windows
- Create a GLOBALS parameter file to name the Manager process
- Add the Manager as a Windows service

Prepare for the installation

1. Locate installation components

Installation components are found on the download pages of the GoldenGate support website, <http://support.goldengate.com>. You will need a user id and password to access this site. Your instructor can provide your id if you don't already have one.

2. Locate installation instructions

Installation instructions, including instructions for upgrades, are available on the documentation pages the GoldenGate support website. Please review the instructions for Windows before proceeding.

3. Download user manuals

User manuals, including Administrator and Reference Guides, are also available on the documentation pages of the GoldenGate support website. Select the appropriate version based on your operating system and GoldenGate release and download to your workstation.

• • • • •

Install GoldenGate

4. Select the appropriate installation combination

Go to <http://support.goldengate.com/>, select Software Downloads, and then select the appropriate combination of operating system version and database version and click Search Downloads. Save the appropriate version of software to your workstation.

5. Put the files in place

Unzip the contents into the GoldenGate <install location> folder.

Note! Please make sure the full name of the folder where GoldenGate is to be installed does not contain any spaces. Your instructor should have supplied the source <install location > and/or target <install location> to be used.

6. Create folders

Create the required folders in the installation directory.

```
Shell> cd <install location>
Shell> ggsci
GGSCI> CREATE SUBDIRS
GGSCI> EXIT
```

7. Oracle-specific installation steps

The following steps are only required when installing GoldenGate for Oracle.

- Turn on supplemental logging at the database level.

This command requires the ALTER DATABASE permissions.

```
Shell> sqlplus <login>/<password>
```

- Switch to the next redo log file.

```
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
SQL> ALTER SYSTEM SWITCH LOGFILE;
SQL> EXIT
```

Install the GoldenGate Manager process

8. Create a GLOBALS parameter file

Execute the following commands from the <install location>.

```
Shell> ggsci
GGSCI> EDIT PARAMS ./GLOBALS
```

In the text editor, type the following:

```
MGRSERVNAME <mgr server>
```

During the lab preparation your instructor supplied a <mgr service> name to be used for this parameter. Using a GLOBALS file in each GoldenGate instance allows you to run multiple Managers as services on Windows. When the service is installed, the Manager name is referenced in GLOBALS, and this name will appear in the Windows Services control panel.

Note: Check to ensure that the GLOBALS file has been added in the GoldenGate installation directory and that it does not have an extension.

Execute the following command to exit GGSCI.

```
GGSCI> EXIT
```

9. Install the Manager service

Execute the following command to install GoldenGate Manager as a Windows service and add GoldenGate events to the Windows Event Viewer.

```
Shell> INSTALL ADDSERVICE ADDEVENTS
```

Note: Adding the Manager as a service is an optional step used when there are multiple environments on the same system or when you want to control the name of the manager for any reason.

Exercise 2.

Installation of GoldenGate for UNIX

• • • • •

Objective

Upon completion of this lesson, you will have installed GoldenGate on a UNIX platform.

During this lesson, you will learn how to:

- Locate installation components and instructions.
- Find and download GoldenGate documentation.
- Install GoldenGate for UNIX.

Prepare for the installation

1. Locate installation components

Installation components are found on the download pages of the GoldenGate support website, <http://support.goldengate.com>. You will need a user id and password to access this site. Your instructor can provide your ids if you don't already have one.

2. Locate installation Instructions

Installation instructions, including instructions for upgrades, are available on the documentation pages on the GoldenGate support website. Please review the instructions for UNIX before proceeding.

3. Download user manuals

User manuals, including Administrator and Reference Guides, are also available on the documentation pages of the GoldenGate support website. Select the appropriate version based on your operating system and GoldenGate release and download to your workstation.

Install GoldenGate

4. Select the appropriate installation combination

Go to <http://support.goldengate.com/>, select Software Downloads, and then select the appropriate combination of operating system version and database version and

• • • • •

click Search Downloads. Save the appropriate version of software to your workstation.

5. Put the files in place

FTP the installation zip in binary mode from the PC to the installation directory on the machine where GoldenGate will be running.

6. Extract the files

Extract the compressed archive using the following command. All GoldenGate files will be placed in the current subdirectory.

```
Shell> gzip -d {filename}.tar.gz
Shell> tar -xvof {filename}.tar
```

7. Create folders

Create the required sub-directories in the installation directory.

```
Shell> cd <install location>
Shell> ggsci
GGSCI> CREATE SUBDIRS
GGSCI> EXIT
```

8. Oracle-specific installation steps

The following steps are only required when GoldenGate will be capturing data from the server. Therefore it is only needed when you are installing GoldenGate for Oracle as your <source> environment.

- Turn on supplemental logging at the database level.

This command requires the ALTER DATABASE permissions.

```
Shell> sqlplus <login>/<password>
```

```
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
```

- Switch to the next redo log file.

```
SQL> ALTER SYSTEM SWITCH LOGFILE;
SQL> EXIT
```

Exercise 3.

Introduction to the Command Interface

• • • • •

Objective

Upon completion of this lesson, you will have used the GoldenGate command interface (GGSCI) and learned how to retrieve online help for all commands.

Using GGSCI

1. Launch the command interface

The command interface can be launched by running it from a command prompt.

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

2. View HELP summary for all commands

```
GGSCI> HELP
GGSCI> HELP ALL
```

3. View HELP summary for a COMMAND/ENTITY

```
GGSCI> HELP ADD EXTRACT
GGSCI> HELP ADD EXTTRAIL
```

4. View your command history

```
GGSCI> HISTORY
```

5. View a brief informational summary of all processes

```
GGSCI> INFO ALL
```

6. View the GoldenGate application log

Messages from GoldenGate processes are stored in the application log. This log is named ggserr.log and is located in the installation directory. Use the VIEW GGSEVT command to view the log.

```
GGSCI> VIEW GGSEVT
```

• • • • •

GoldenGate for Oracle to Oracle

Objective

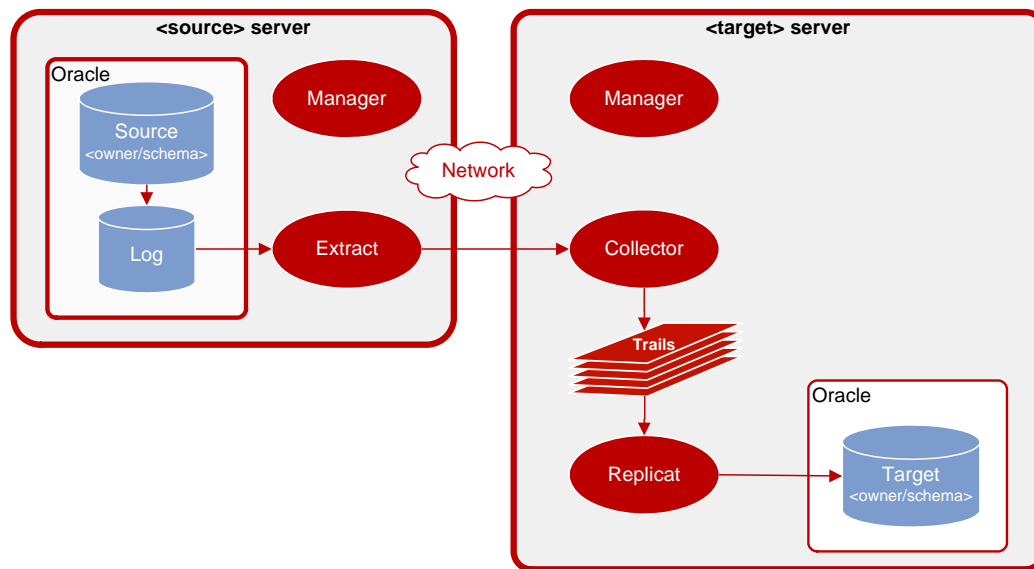
Upon completion of this lesson, you will be able to keep two Oracle databases synchronized.

During this lesson, you will:

- Prepare the database and the 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 GoldenGate installed on two systems connected by TCP/IP – one containing the source data and the other the replicated data.



Overview of tasks

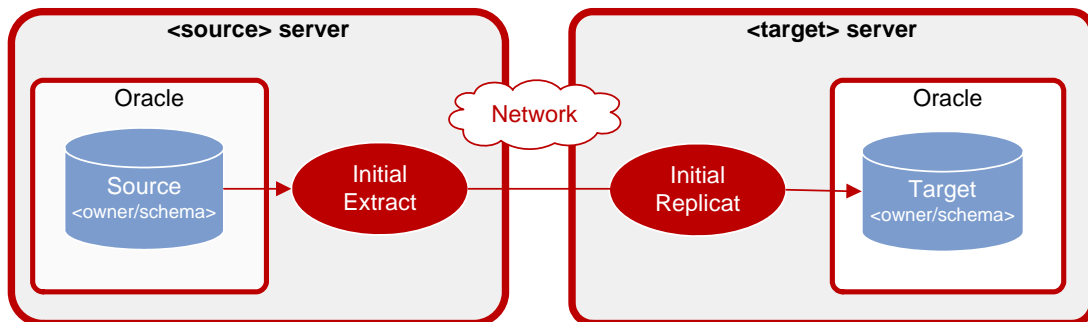
Prepare the Environment

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 Initial Data Load

To initially load data, there are techniques such as Backup/Restore or Export/Import, both of which have pros and cons. Alternatively 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 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 deliver the captured change data into the target database.


```
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 5.**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>
RMTHOST <target>, MGRPORT <port>
RTMTASK 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 6.

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 *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?

Exercise 7.

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 8.**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.

Workshops

This section includes a selection of exercises to provide more in-depth training in certain areas such as:

- Using GoldenGate documentation
- More practice using GGSCI commands
- More practice using parameters
- Using the Logdump utility
- Using encryption
- Using a data pump
- Alternative methods for the initial data load
- Using SQLEXEC
- Replicating DDL

Reference Guide organization

Open the *GoldenGate for Windows and UNIX Reference Guide* and click on the Bookmarks tab on the left of the window to display the bookmarks.

3. Typographic conventions

Click on the plus sign (+) to the left of About the GoldenGate Guides. Notice that the Typographic conventions used in this manual section provides information on how to interpret the syntax statements that can be found throughout the manuals. Use this information to answer the following question.

Given the following syntax statements:

```
DISCARDFILE <file name> [, APPEND | PURGE | MEGABYTES <n>]
SEND REPLICAT <group name> {FORCESTOP | GETLAG | STATUS | REPORT}
```

Why are square brackets, [], used in the first instance and curly brackets, {}, in the second?

4. GGSCI commands

Click on the plus sign (+) to the left of GGSCI Commands. Notice that the commands are grouped based on the GoldenGate process or data that is affected. Expand some of the subcategories, such as Extract commands or Trail commands to view the commands that can be used.

Which command summary groups include an ADD command?

5. GoldenGate parameters summaries

Click on the plus sign (+) to the left of GoldenGate Parameters and then on the subcategories, such as Parameter summaries and Replicat parameters summary. Note that the parameters are grouped based on the GoldenGate process to which they apply.

6. GoldenGate parameters alphabetic reference

Click on the plus sign (+) to the left of GoldenGate Parameters and then on Alphabetic Reference. Note that this section lists all of the parameters that are valid on the Windows/UNIX platform.

Click on one of the parameters and find the different types of information that is supplied for each one. Use this information to answer the questions below.

- ☐ The processes the parameter is valid for, such as Extract or Replicat.
- ☐ A description of the use of the parameter.
- ☐ The default value that will be used if the parameter is not entered.
- ☐ A syntax statement of how to enter the parameter.
- ☐ A table explaining any options that can be used with the parameter.
- ☐ Examples of the use of the parameter.

Questions

Which processes allow the SPECIALRUN parameter?

What is the default value of the CHECKMINUTES parameter?

What options can be used with the AUTORESTART parameter? Which are required?

Searching the manuals

7. Word search

Open the *GoldenGate for Windows and UNIX Reference Guide*. Close the Bookmarks tab if it is open. Click on the binoculars Search icon on the Adobe PDF file tool bar or click CNTL-F to open the search dialog.

Enter REPORT (in uppercase) as the word or phrase you would like to search for. Mark the Case-Sensitive and Whole words only checkboxes and clear the others. Press the Search button and note the type of response you get.

8. Partial word search

Again using the *GoldenGate for Windows and UNIX Reference Guide*, enter REPORT (in uppercase) as the word or phrase you would like to search for. Clear all of the checkboxes except Case-Sensitive. Press the Search button and note the type of response you get.

Solutions

1. Finding information using bookmarks

Expand and collapse the bookmarks to locate information on the following topics:

- How to use a filter to select records.
Expand Mapping and Manipulating Data, then Selecting records, then Selecting records with a FILTER clause.
- Troubleshooting trail files that are not being purged.
Expand Troubleshooting GoldenGate, then Troubleshooting file maintenance problems, then Trail files not being purged.

2. Finding information with the index

Go to the index for the manual. Why are some of the entries in uppercase?

The uppercase entries are GoldenGate parameters or commands.

3. Typographic conventions

Given the following syntax statements:

```
DISCARDFILE <file name> [, APPEND | PURGE | MEGABYTES <n>]  
SEND REPLICAT <group name> {FORCESTOP | GETLAG | STATUS | REPORT}
```

Why are square brackets, [], used in the first instance and curly brackets, {}, in the second?

Square brackets indicate multiple optional arguments, so one or more of the options [, APPEND | PURGE | MEGABYTES <n>] may be selected but none are required. Curley brackets indicate mutually-exclusive arguments, so only one of the options {FORCESTOP | GETLAG | STATUS | REPORT} can be selected.

4. GGSCI commands

Which command summary groups include an ADD command?

Extract commands, Trail Commands, Replicat commands, Checkpoint table commands, Trandata commands, and Oracle trace table commands .

5. GoldenGate parameters summaries

6. GoldenGate parameters alphabetic reference

Which processes allow the SPECIALRUN parameter?

Extract and Replicat

What is the default value of the CHECKMINUTES parameter?

10 minutes

What arguments can be used with the AUTORESTART parameter? Which are required?

<process type> and <group name> are required arguments. Optional arguments include the number of RETRIES, WAITMINUTES, and RESETMINUTES.

Searching the manuals

7. Word search

Enter REPORT (in uppercase) as the word or phrase you would like to search for. Mark only the Case-Sensitive and Whole words only checkboxes. Press the Search button and note the type of response you get.

All occurrences of the parameter REPORT are returned in the search list.

8. Partial word search

Again using the *GoldenGate for Windows and UNIX Reference Guide*, enter REPORT (in uppercase) as the word or phrase you would like to search for. Clear all of the checkboxes except Case-Sensitive. Press the Search button and note the type of response you get.

All parameters that include REPORT — such as REPORT, REPORTCOUNT, and REPORTROLLOVER — are in the search list.

Exercise 10.

Using Commands on Windows/UNIX

• • • • •

Objective

Upon completion of this workshop, you will have better understanding of the commands within GoldenGate command interface.

Note: This lab encourages you to interact with command interface. It expects you to use reference materials, online Help, and any previous knowledge you may have to determine the commands to be used. You can record your commands and then verify them using the Solutions section that follows the exercise.

Using the command interface

On Window or UNIX, perform the following actions:

1. Getting started

- ☐ Start the command interface.

- ☐ Login to your database.

2. Finding out about the database

- ☐ List out all tables for your schema.

- ☐ List out all tables for another schema.

3. Making changes in the environment

- ☐ Change the size of trail to 100 MB in size.

- ☐ Tell the Extract to roll to the next trail without stopping the group.

• • • • •

4. Getting information

- Use the STATS command for both Extract and Replicat.

-
- Display the STATS counters for only totals for all of the tables.

-
- Play with the History and FC commands.

-
- Clean up old runtime history, but save the last 3 history lines.

-
- Generate a new report file for all groups.
-

5. Exploring other options

- View the online help and experiment with other commands.

SOLUTIONS: Using Commands

1. Getting started

- Start the command interface.

```
ggsci
```

- Login to your database.

For Oracle:

```
DBLOGIN USERID <login> PASSWORD <password>
```

For SQL Server:

```
DBLOGIN SOURCEDB <dsn> USERID <login> PASSWORD <password>
```

For DB2, Sybase, Teradata:

```
DBLOGIN SOURCEDB <database> USERID <login> PASSWORD <password>
```

For Ingres:

```
DBLOGIN SOURCEDB <dsn>
```

2. Finding out about the database

- List out all tables for your schema.

```
LIST TABLES *
```

- List out all tables for another schema.

```
LIST TABLES <database>.*
```

Or

```
LIST TABLES <owner/schema>.*
```

3. Making changes in the environment

- Change the size of trail to 100 MB in size.

```
STOP EXTRACT <group>  
ALTER RMTTRAIL ./dirdat/<trail id>, MEGABYTES 100, EXTRACT  
<group>  
START EXTRACT <group>
```

- Tell the Extract to roll to the next trail without stopping the group.

```
SEND EXTRACT <group>, ROLLOVER
```

4. Getting information

- Use the STATS command for both Extract and Replicat.

```
STATS <group>
```

- Display the STATS counters for only totals for all of the tables.

```
STATS <group>, TABLE *, TOTALONLY *
```

- Play with the History and FC commands.

```
HISTORY  
FC STATS  
FC INFO MANAGER
```

- Clean up old runtime history, but save the last 3 history lines.

```
CLEANUP EXTRACT *, SAVE 3  
CLEANUP REPLICAT *, SAVE 3
```

- Generate a new report file for all groups.

```
SEND ER *, REPORT
```

5. Exploring other options

- View the online help and experiment with other commands.

```
HELP  
HELP ADD EXTRACT
```

Exercise 11.

Using Parameters

• • • • •

Objective

Upon completion of this lesson, you will have configured the Manager and Extract processes using more advanced parameters.

During this lesson, you will:

- Modify the source system Manager parameters.
- Modify the target system Manager parameters.
- Modify the Extract parameters and restart the Extract process.

Note: This exercise expects you to use reference materials and any previous knowledge you may have to determine the parameters and options to be used. You can record your parameters and then verify them using the Solutions section that follows the exercise.

Modify source Manager parameters

Perform the following exercises on the <source> system.

1. Modify the Manager parameter file

Modify the Manager parameter file for the following:

- Use ports in the range specified by the instructor when starting GoldenGate processes.

- Have an Extract group automatically start when the manager process started.

- Have all Extract groups automatically restart after abending. Manager should retry the operation 3 times, waiting 1 minute between each attempt.

- Report the current lag for each Extract/Replicat group every hour to the application log file.

• • • • •

-
- Write an informational message to the application log file if the lag for any Extract/Replicat group exceeds 30 minutes.
-
- Write a critical message to the application log file if the lag for any Extract/Replicat group exceeds 45 minutes.
-

2. Activate the changes

- Stop and restart the manager to activate the changes.
- Verify that the manager has started.

Modify the target Manager parameters

Perform the following exercises on the <target> system.

3. Modify the Manager parameter file

Modify the Manager parameter file for the following:

- Use ports in the range of 7820 through 7840 when starting GoldenGate processes.
-
- Have all Replicat groups that have a group name of R* automatically start when the manager process is started.
-
- Have all Replicat groups automatically restart after abending. Manager should retry the operation 3 times, waiting 1 minute between each attempt.
-
- Delete the remote trail files once the Replicat has finished processing them and the trail has been kept for at least 2 hours.
-
- Report the current lag for each Extract/Replicat group every hour to the application log file.
-

- Write an informational message to the application log file if the lag for any Extract/Replicat group exceeds 30 minutes.
-

- Write a critical message to the application log file if the lag for any Extract/Replicat group exceeds 45 minutes.
-

4. Activate the changes

- Stop and restart the manager to activate the changes.
- Verify that the manager has started.

Modify the Extract parameters

Perform the following exercises on the <source> system.

5. Modify the Extract parameter file for the following:

- Write a message to the report file after processing every 1000 records.
-
- Only extract data from the TCUSTOMER table if the state column is not “PA”.
-

6. Activate the changes

- Stop and restart the Extract to have the changes take effect.
- Verify that the Extract has started.

SOLUTIONS: Using Parameters

Modify source system Manager parameters

```
GGSCI> EDIT PARAM MGR
```

```
PORT <port>
DYNAMICPORTLIST 7820-7840
AUTOSTART EXTRACT <group>
AUTORESTART EXTRACT *, WAITMINUTES 1, RETRIES 3
LAGREPORTHOURS 1
LAGINFOMINUTES 30
LAGCRITICALMINUTES 45
```

Modify the target system Manager parameters

```
GGSCI> EDIT PARAM MGR
```

```
PORT <port>
DYNAMICPORTLIST 7820-7840
PURGEOLDEXTRACTS <install loc>/dirdat/<trail id>*, USECHECKPOINTS, &
MINKEEPHOURS 2
AUTOSTART REPLICAT R*
AUTORESTART REPLICAT *, WAITMINUTES 1, RETRIES 3
LAGREPORTHOURS 1
LAGINFOMINUTES 30
LAGCRITICALMINUTES 45
```

Modify the Extract parameters

```
GGSCI> EDIT PARAM <extract group>
```

Your existing parameter file should have the following parameters added to it.

```
REPORTCOUNT EVERY 1000 RECORDS
TABLE <owner/schema>.TCUSTMER, WHERE (STATE<>"PA");
TABLE <owner/schema>.TCUSTORD;
```

Exercise 12.

Using the Logdump Utility for Windows/UNIX

.....

Objective

Upon completion of this workshop, you will be able to use the Logdump utility to look at records in a GoldenGate trail file.

In this exercise you will:

- Identify the file header record.
- Identify the trail record and its header area.
- Use the header and detail attributes when displaying a record.
- Display records only associated with the TCUSTORD table.
- Position to the middle of a record and scan for the next record header.
- Find a count of how many records are in a log file.
- Use the Logdump help feature.

Note: This exercise expects you to use reference materials and any previous knowledge to determine the commands to be used. You can record your commands and then verify them using the Solutions section that follows the exercise.

Using Logdump Help

1. Using the Logdump Help feature

- Issue the help command to display a list of all commands available in Logdump.

Identifying file headers

2. Identify the file headers

- Start Logdump and open the <install_path>/dirdat/<trail id>000000 trail file.

.....

- View the first record.

Note the kind of record it is.

- Turn FILEHEADER ON and view the record again.

Note the difference in the display from the output in the last step.

Configuring Logdump's display

3. Using the header and detail attributes when displaying a record

- View the next record.

Note the kind of record it is.

- Position back to RBA 0 in the file, set the header and detail attributes to on, and view the same record again.

Note the difference in the display from the output in the last step.

3. Display only records associated with the TCUSTORD table

- Position back to RBA 0 in the file.

-
- Set a FILTER to only include records for the TCUSTORD table.

Note: Names must be fully qualified and are case sensitive.

- Use the NEXT command a few times to ensure only records for the TCUSTORD table are displayed.

Getting more information

4. Position to the middle of a record and scan for the next record header

- To simulate trying to find the next record when its RBA is unknown, choose an arbitrary number (i.e. 250) and position to that RBA in the trail file. Then scan for the next record header to display that record.

5. Find a count of how many records are in the trail file

- Position back to RBA 0 in the file.

-
- Display a count of the total number of records in the file and a per table count of records in the file.
-

SOLUTIONS: Logdump

Using Logdump Help

1. Use the Logdump Help feature

- Issue the help command to display a list of all commands available in Logdump.

```
LOGDUMP> help
```

Configuring Logdump's display

2. Identify the file headers

- Start Logdump and open the <install_path>/dirdat/<trail id>000000 trail file.

```
Shell> cd <install location>
Shell> logdump
LOGDUMP> open dirdat/rt000000
```

- View the first record.

```
LOGDUMP> n
```

Note the kind of record it is.

This is the trail file header record.

- Turn FILEHEADER ON and view the record again.

```
LOGDUMP> fileheader on
LOGDUMP> pos 0
LOGDUMP> n
```

Note the difference in the display from the output in the last step.

Detail on the trail, the machine being used, the Extract that produced the trail and other information is displayed with FILEHEADER ON.

3. Using the header and detail attributes when displaying a record

- View the first next record.

```
LOGDUMP> n
```

Note the kind of record it is.

This is a standard trail record.

- Position back to RBA 0 in the file, set the header and detail attributes to on, and view the same record again.

```
LOGDUMP> pos 0
LOGDUMP> ghdr on
LOGDUMP> detail on
LOGDUMP> n
LOGDUMP> n
```

Note the difference in the display from the output in the last step.

The trail record header area and detail on the columns has been added to the display.

4. Display records only associated with the TCUSTORD table

- Position back to RBA 0 in the file.

```
LOGDUMP> pos 0
```

- Set a FILTER to only include records for the TCUSTORD table.

```
LOGDUMP> filter include filename <owner/schema>.TCUSTORD
```

Note: Names must be fully qualified and are case sensitive.

- Use the NEXT command a few times to ensure only records for the TCUSTORD table are displayed.

```
LOGDUMP> n
```

Getting more information

5. Position to the middle of a record and scan for the next record header

- To simulate trying to find the next record when its RBA is unknown, choose an arbitrary number (i.e. 250) and position to that RBA in the trail file. Then scan for the next record header to display that record.

```
LOGDUMP> pos 250
LOGDUMP> sfh
```

6. Find a count of how many records are in the trail file

- Position back to RBA 0 in the file.

```
LOGDUMP> pos 0
```

- Display a count of the total number of records in the file and a per table count of records in the file.

```
LOGDUMP> count detail
```

Exercise 13. Using Encryption

● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●

Objective

Upon completion of this workshop, you will have better understanding using encryption for passwords as well as encrypting the data sent over TCP/IP.

During this lesson, you will learn how to:

- Use KEYGEN to generate a user define key
- Use GGSCI to encrypt a database login password
- Modify the parameter file to use the encrypted passwords

Note: This exercise expects you to use reference materials and any previous knowledge you may have to determine the commands to be used. You can record your commands and then verify them using the Solutions section that follows the exercise.

Using KEYGEN to generate a user defined key

1. Generate keys using the KEYGEN utility

From a command prompt, use the KEYGEN utility to generate 4 keys that are 128-bit.

2. Create the ENCKEYS file

Copy those keys into a text file named ENCKEYS stored in the directory where GoldenGate is installed and assign the following names to the keys

FIRSTKEY
YOURKEY
MYKEY
MONKEY

Note: ENCKEYS must be in both the source and target <install location>.

.....

Using GGSCI to generate encrypted login passwords

3. Use the default key to encrypt password

From the GGSCI command prompt, encrypt a password using the default key.

Record the response:

4. Use ENCKEYS to encrypt password

From the GGSCI command prompt, encrypt a password using the user key named MYKEY.

Record the response:

Using Extract parameters for password and data encryption

5. Encrypting trail data in the parameter file

Change some parameters to encrypt the trail data over TCP/IP using the MONKEY key.

6. Encrypting passwords in the parameter file.

Change some parameters to use database password encryption using the encrypted password generated using the MYKEY key.

SOLUTIONS: Encryption

1. Generate keys using the KEYGEN utility

From a command prompt, use the KEYGEN utility to generate 4 keys that are 128-bit.

```
shell> keygen 128 4
```

Sample Results:

```
0xBEADBE598C183E6D23A4E365C9005802
0xC214DF1E3A56526C745EBA34B2382138
0x1BD1046477B04D241AA70438200D4A50
0x758D2A29B30A495CC0EF4E3B8EE17268
```

2. Create the ENCKEYS file

Copy the generated keys into a text file named ENCKEYS stored in the source directory where GoldenGate is installed and assign the indicated names to the keys.

An example of how your ENCKEYS file should look after you copy in your keys:

```
FIRSTKEY 0xBEADBE598C183E6D23A4E365C9005802
YOURKEY   0xC214DF1E3A56526C745EBA34B2382138
MYKEY      0x1BD1046477B04D241AA70438200D4A50
MONKEY     0x758D2A29B30A495CC0EF4E3B8EE17268
```

Put a copy of the ENCKEYS file on the target installation directory.

3. Use the default key to encrypt password

From the GGSCI command prompt, encrypt a password using the default key.

```
GGSCI> encrypt password <password>
```

An example command and the encrypted response:

```
GGSCI> encrypt password ggstrn01
```

No key specified, using default key...

Encrypted password: AACAAAAAAAAAAAAIAOIWAOCOHTADHUIKG

4. Use ENCKEYS to encrypt password

From the GGSCI command prompt, encrypt a password using the user key named MYKEY.

```
GGSCI> encrypt password ggstrn01 encryptkey mykey
```

Sample response:

.....

Encrypted password: AACAAAAAAAAAAAAIAOALEPAVBWCKAPJXG

5. Encrypting trail data in the parameter file

Change some parameters to encrypt the trail data over TCP/IP using the MONKEY key.

```
RMTHOST <target>, MGRPORT 7809, ENCRYPT BLOWFISH, KEYNAME MONKEY
```

6. Encrypting passwords in the parameter file.

Change some parameters to use database password encryption using the encrypted password generated using the MYKEY key.

To do this, copy the encrypted password generated in step 4 using the MYKEY key and the ggstrn01 password and paste it into the parameter file as shown in the example below.

```
EXTRACT EXTORA
```

```
USERID "ggstrn01", PASSWORD AACAAAAAAAAAAAAIAOALEPAVBWCKAPJXG,  
encryptkey MYKEY
```


Exercise 14.

Configure a Data Pump for Windows/UNIX

.....

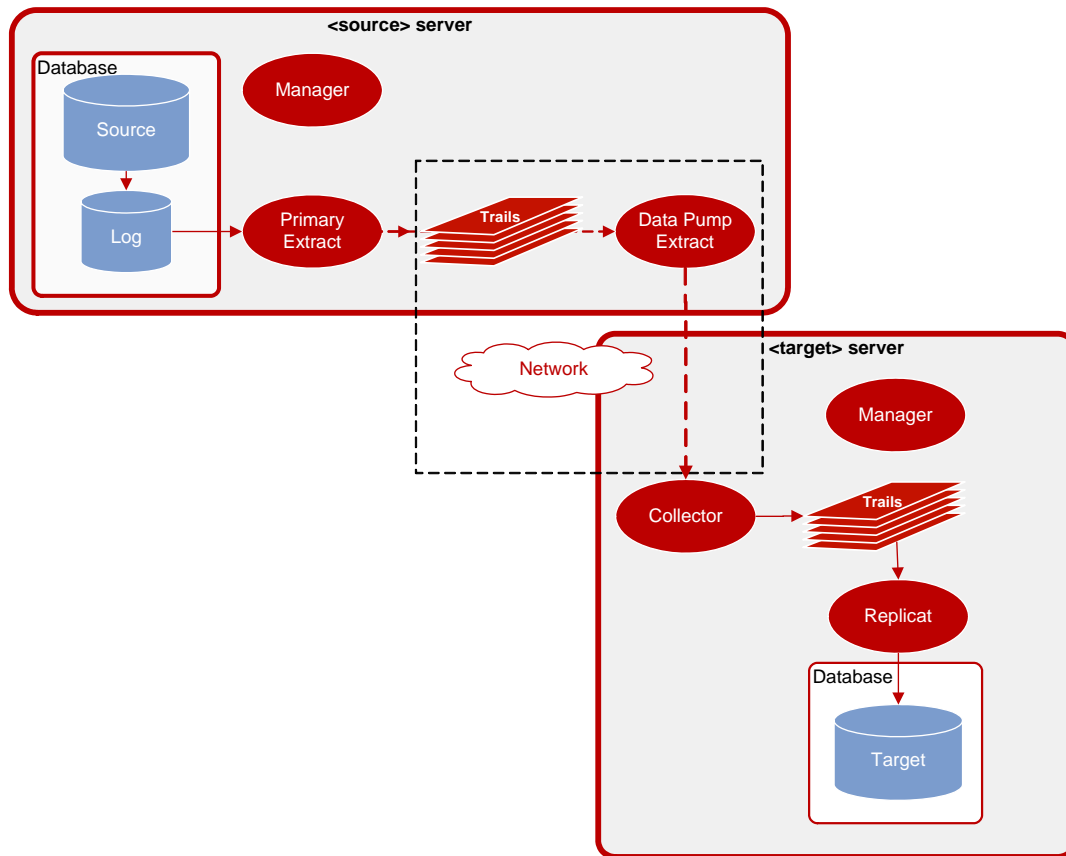
Objective

The goals of this exercise are to:

- Configure a change capture process that writes to a local extract trail.
- Configure a data pump that reads from the local extract trail and writes to a remote trail.

Data Pump Configuration

The dotted lines in the diagram illustrate the processes added for a configuration with a data pump. When using a data pump, the activity extracted from the transaction log is stored in a local trail. Then the data pump Extract moves the data from the local trail over the network to the remote trail in the target location.



.....

Prerequisites

This exercise assumes you successfully completed the previous lab on configuring change capture and replication between two databases on a Windows or UNIX platform.

The instructions for operations on the database are not specific. Please refer back to the previous labs if you need help with command syntax to log into and execute operations in your database.

Prepare the Environment

1. Prepare the source database

To allow the tables from the previous lab to be reused, perform the following steps on the <source> system.

- Stop the Extract from the previous lab.

```
Shell> cd <install location>
Shell> ggsci
GGSCI> STOP EXTRACT <Extract name>
```

Verify results

```
GGSCI> STATUS <Extract name>
```

- If you have not already done so, log on to the applicable database utility.

- For UNIX DB2:

```
Shell> db2
db2 => connect to <database>
```

- For zOS DB2:

```
Shell> odb2
```

- For MySQL:

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

- For Oracle:

```
Shell> sqlplus <login>/<password>
```

- For SQL Server 2000, run the SQL Server SQL Query Analyzer on the source system. Log in using the <login> user and <password> password.

- For SQL Server 2005, launch SQL Server Management Studio and enter the name of the local SQL Server instance. Depending on the type of authentication, you may also need to enter the <login> user and <password> password.

- For SQL/MX:

```
OSS> ./mxci
```

- For Sybase:

```
Shell> isql -U {login} -P {password}
```

- For Teradata:

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

- Substitute the correct syntax for the following generic commands to delete the data in the <source> database.

```
> delete from tcustmer;  
> delete from tcustord;  
> commit;
```

Verify results

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

2. Prepare the target database

Execute the following commands for the <target> system.

- Stop the Replicat from the previous lab.

```
Shell> cd <install location>  
Shell> ggsci  
GGSCI> STOP REPLICAT <Replicat name>
```

Verify results

```
GGSCI> STATUS <Replicat name>  
GGSCI> EXIT
```

- Logon to the database if necessary and delete the data in your <target> database.

```
> <dblogin sourcedb entries as appropriate>,  
    login <login>, password <password>  
  
> delete from tcustmer;
```

```
> delete from tcustord;
> commit;
```

Verify results

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

Configure initial Extract

3. Create the initial Extract parameter file

On the <source> system, define the parameters for an initial Extract named EIEX<unique id> to capture changes and route them to a local GoldenGate trail.

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

Note: The RMTHOST parameter is not included in the Extract parameter file because the trail for the data pump must be local.

```
--
-- Extract parameter file to capture
-- TCUSTMER and TCUSTORD changes
--
EXTRACT EIEX<unique id>
<appropriate login for your database>
EXTTRAIL ./dirdat/<exttrail>
<table statements appropriate for your database>;
```

Note: Record the two characters selected for your <exttrail>: _____. You will need this in step 5 for adding the EXTTRAIL and for adding the pump in step 7.

4. Add the Extract group

Execute the following commands on the <source> system.

```
GGSCI> ADD EXTRACT EIEX<unique id>, <TRANLOG, VAMTRAILSOURCE, etc.
depending on database>, BEGIN NOW
```

5. Define the GoldenGate trail

Execute the following commands on the <source> system to add the trail defined in the EIEX<unique id> parameters.

```
GGSCI> ADD EXTTRAIL ./dirdat/<exttrail>, EXTRACT EIEX<unique id>,
MEGABYTES 5
```

Configure data pump

6. Create the data pump Extract parameter file

Execute the following commands on the <source> system to define an Extract named EPMP<unique id> to route changes captured in the local trails to remote trails.

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

```
--
-- Data Pump parameter file
--
EXTRACT EPMP<unique id>
PASSTHRU
RMTHOST <target >, MGRPORT <port>
RMTTRAIL ./dirdat/<rmttrail>
<table statements appropriate for your database>;
```

Note: Record the two characters selected for your <rmttrail> identifier: _____. You will need this in step 8.

7. Add the data pump capture group

Execute the following commands on the <source> system.

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

8. Define the GoldenGate trail

Execute the following commands on the <source> system to add the trail declared in the PUMP parameters.

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

9. Start the data pump capture process

Execute the following commands on the <source> system.

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

Verify results:

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

Generate activity and verify results

10. Generate activity

On the <source> system, use the <database>insert.sql script from the earlier lab to generate activity in your recently emptied source tables.

11. Verify results

Verify the results in the source tables by substituting your database access utility's syntax for the following commands.

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

On the <target> system, verify the results in the trail created by the Extract pump by opening it in Logdump and paging through the records.

```
Shell> cd <install location>  
Shell> logdump  
Logdump> open ./dirdat/<trail id>000000  
Logdump> n  
Logdump> n
```

When you are finished:

```
Logdump> exit
```

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. The advantage of a data pump

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

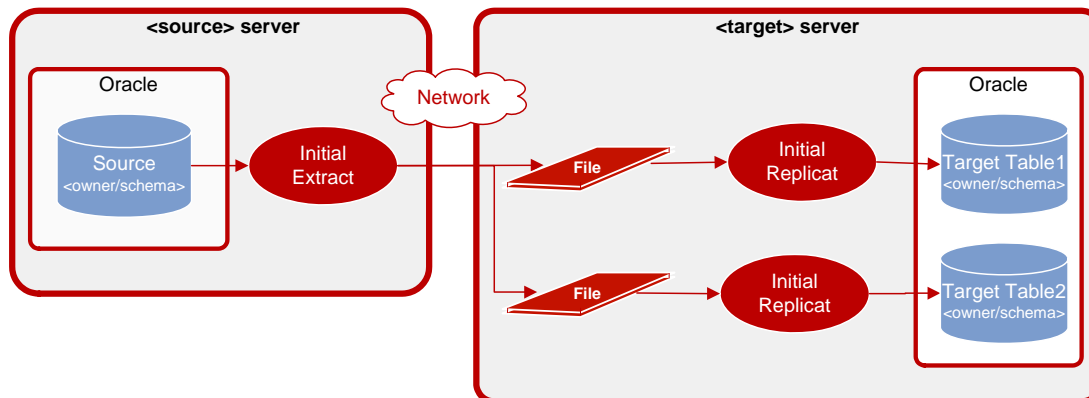
Exercise 15.**Initial Data Load using File to Replicat**

.....

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 load using file to Replicat

This method uses an Extract process to extract source data into flat files or trails on the target system. The initial load Replicat task then moves the data from the intermediate file to the target database.

Truncate target tables

Note: This exercise assumes that you have completed an initial data load using the direct method. If this is not the case, you may skip step 1.

1. Truncate your target tables

Execute the following commands on the <target> system.

```

Shell> sqlplus <login>/<password>
SQL> truncate table tcustmer;
SQL> truncate table tcustord;
  
```

.....


```
SQL> commit;
```

Initial data capture

2. Configure the initial load Extract parameter file

Execute the following commands on the <source> system to create an Extract named ELOD<unique id>².

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

```
--
-- GoldenGate Initial Data Capture
-- for TCUSTMER and TCUSTORD
--
SOURCEISTABLE
USERID <login>, PASSWORD <password>
RMTHOST <target>, MGRPORT <port>
RMTFILE ./dirdat/TCUSTMER.DAT, PURGE
TABLE <owner/schema>.TCUSTMER;
RMTFILE ./dirdat/TCUSTORD.DAT, PURGE
TABLE <owner/schema>.TCUSTORD;
```

3. Execute the initial load capture process

Start the initial load Extract directly from the UNIX command line using the following command.

```
Shell> cd <install location>
Shell> extract paramfile dirprm/elod<unique id>.prm reportfile
dirrpt/ELOD<unique id>.rpt
```

Verify the results:

```
Shell> ggsci
GGSCI> VIEW REPORT ELOD<unique id>
```

Initial data delivery

3. Configure the initial load delivery parameter file

Execute the following commands on the <target> system.

² The process names used in lab exercises, for example EINIBD1, are made up of 1) one character to identify the type of process (E for Extract, R for Replicat); 2) three characters to describe the type of process (INI or LOD for initial data load, ORA for capture from or delivery to an Oracle database, etc.) and 3) three characters usually made up of the student's initials plus a sequential number to uniquely identify multiple occurrences of that type of process.

○ Configure the data load parameters for TCUSTMER

```
GGSCI> EDIT PARAMS LOADTCUSTMER
```

```
--  
-- GoldenGate Initial Load Delivery for TCUSTMER  
--  
SPECIALRUN  
END RUNTIME  
USERID <login>, PASSWORD "<password>"  
ASSUMETARGETDEFS  
EXTFILE ./dirdat/TCUSTMER.dat  
MAP <owner/schema>.TCUSTMER, TARGET <owner/schema>.TCUSTMER;
```

○ Repeat this process for the TCUSTORD initial load parameter file.

```
GGSCI> EDIT PARAMS LOADTCUSTORD
```

```
--  
-- GoldenGate Initial Load Delivery for TCUSTORD  
--  
SPECIALRUN  
END RUNTIME  
USERID <login>, PASSWORD "<password>"  
ASSUMETARGETDEFS  
EXTFILE ./dirdat/TCUSTORD.DAT  
MAP <owner/schema>.TCUSTORD, TARGET <owner/schema>.TCUSTORD;
```

○ Exit GGSCI

```
GGSCI> EXIT
```

4. Execute the initial load delivery process

```
Shell> replicat paramfile dirprm/loadtcustmer.prm reportfile  
dirrpt/LOADTCUSTMER.rpt
```

```
Shell> replicat paramfile dirprm/loadtcustord.prm reportfile  
dirrpt/LOADTCUSTORD.rpt
```

Wait 10 seconds and then verify the results:

```
Shell> ggsci  
GGSCI> VIEW REPORT LOADTCUSTMER  
GGSCI> VIEW REPORT LOADTCUSTORD
```

Exercise 16.

Stored Procedures

• • • • •

Objective

Upon completion of this workshop, you will have a better understanding of using stored procedures and SQL queries with the SQLEXEC parameter.

During this lesson, you will learn how to:

- Create a Replicat parameter file that issues a SQL query to retrieve the name of the state from a state code look up table based on the two-character state code.
- Create a Replicat parameter file that calls a stored procedure to retrieve the name of the state from a state code look up table based on the two-character state code.

Prerequisites

This lab assumes that GoldenGate has been installed, the manager process started, and source tables created in previous labs. Since the lab uses two empty tables, there is no need to conduct an initial data load. It is assumed that supplemental logging is enabled for all source tables.

Preparing the environment

To prepare for this lab, you will need to truncate your source TCUSTMER table. You will create a new table similar to TCUSTMER, but the state column will be expanded to hold the name of the state.

1. Truncate the TCUSTMER table

Truncate your <source> TCUSTMER table.

2. Create new table

Perform the following steps on the <target> to create a new table and expand it.

- The script demo_ora_create_tcustmer_report.sql will be provided by your instructor. Use this to create a table on the <target> system named TCUSTMER_REPORT based on the TCUSTMER layout.
- Replace the column named STATE with a column named STATE_NAME

• • • • •

- Define the STATE_NAME as VARCHAR2(30)

3. Create state code table

On the <target> system, create and load the STATECODE table

4. Generate and execute source definitions

Perform the following on the <source> system.

- Create a parameter file for DEFGEN to generate source definitions for TCUSTOMER

- Execute the source definition generator

- Transfer the source.def file that is created to the <target> system.

Using SQL queries

5. Create and configure capture

On the <source> system, create and configure the Extract.

- Create an Extract group named EXTSQ

-
- Create an Extract parameter file that extracts data from the TCUSTMER table and saves the data in a trail named ./dirdat/SQ

-
-
-
- Define the SQ trails to be 10 megabytes in size.
-

6. Create and configure delivery

On the <target> system create and configure Replicat.

- Create a Replicat group named REPSQ which reads the SQ trail.
-

- Create a Replicat parameter file that:

- Maps TCUSTMER into TCUSTMER_REPORT.
 - Uses SQLEXEC on the MAP statement
 - Execute a SQL query named STATE_LOOKUP
 - Pass the input STATE code which returns the NAME from the STATECODE table
 - Get the value from the query and map the result to the STATE_NAME column TCUSTMER_REPORT
-
-
-
-
-
-

7. Start both Extract and Replicat processes.

Start the Extract on the <source> system.

Start the Replicat on the <target> system.

8. Generate inserts and validate results

Generate inserts into TCUSTMER on the <source> system. Validate the results in TCUSTMER_REPORT on the <target> system.

Using stored procedures

9. Create stored procedure

Using the concepts learned in the previous lab, create a new stored procedure that performs the same query that passes the state name back to the SQLEXEC to be used in target column mapping.

10. Stop Extract and Replicat

Stop the Extract (on the <source>) and Replicat (on the <target>) that were used in the earlier exercise.

11. Truncate data

Truncate the data in the <source> and <target> tables.

12. Create and configure capture group

Perform the following commands on the <source> system.

- Create an Extract group named EXTSP
-

- Create an Extract parameter file that extracts data from the TCUSTMER table and saves the data in a trail named ./dirdat/SP
-

-
-
- Define the SP trails to be 2 megabytes in size.
-

13. Create and configure delivery group

Perform the following commands on the <target> system.

- Create a Replicat group named REPSR which reads the SP trail.

- Create a SQL script that creates a stored procedure named STATE_NAME_LOOKUP that defines:
 - An input parameter named code_param as a varchar
 - An output parameter named state_name as a varchar
 - A select of the name from the STATECODE table where the code_param is equal to the two-character state column

- Create a Replicat parameter file that maps TCUSTOMER into TCUSTOMER_REPORT
 - Use SQLEXEC on the MAP statement
 - Execute the SPNAME STATE_NAME_LOOKUP
 - Pass the input PARAMS to the STATE column
 - Map the result of the procedure to the target column named STATE_NAME

Solutions: Stored Procedures

Preparing the environment

1. Truncate the TCUSTMER table

Truncate your <source> TCUSTMER table.

```
SQL> TRUNCATE TABLE TCUSTMER;
```

2. Create new table

Perform the following steps on the <target> to create a new table and expand it.

```
SQL> CREATE TABLE TCUSTMER_REPORT
(
    cust_code      VARCHAR2(4),
    name           VARCHAR2(30),
    city           VARCHAR2(20),
    state_name     VARCHAR(30),
    PRIMARY KEY (cust_code)
    USING INDEX );
```

If the instructor has provided sample scripts:

```
sqlplus login/password @demo_oracle_create_tcustmer_report.sql
```

3. Create state codes table

On the <target> system, create and load the STATECODE table

```
sqlplus login/password @demo_oracle_create_state_codes.sql
sqlplus login/password @demo_oracle_insert_state_codes.sql
```

4. Generate and execute source definitions

Perform the following on the <source> system.

- Generate source definitions for TCUSTMER.

```
GGSCI> EDIT PARAM DEFGEN
```

```
DEFSEFILE ./dirdef/MYDEFS.DEF
USERID <login>, PASSWORD <password>
TABLE <owner/schema>.TCUSTMER;
```

- Execute the source definition generator

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


- Transfer the source.def file that is created to the <target> system.

Execute the following commands on the <source> system to transfer the DEFGEN output to the <target> system..

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

Using SQL queries

5. Create and configure capture

On the <source> system, create and configure the Extract.

- Create an Extract group named EXTSQL

```
GGSCI> ADD EXTRACT EXTSQL, TRANLOG, BEGIN NOW
```

- Create an Extract parameter file that extracts data from the TCUSTMER table and saves the data in a trail named ./dirdat/SQ

```
GGSCI> EDIT PARAM EXTSQL
```

```
EXTRACT EXTSQL
USERID <user> PASSWORD <password>
RMTHOST <target>, MGRPORT <port>
RMTTRAIL ./dirdat/sq
TABLE <schema>.TCUSTMER;
```

- Define the SQ trails to be 10 megabytes in size.

```
GGSCI> ADD RMTTRAIL ./dirdat/sq, EXTRACT EXTSQL, megabytes 10
```

6. Create and configure delivery

On the <target> system, create and configure Replicat.

- Create a Replicat group named REPSQ which reads the sq trail.

```
GGSCI> ADD REPLICAT REPSQ, EXTTRAIL ./dirdat/sq
```

- Create a Replicat parameter file

```
GGSCI> EDIT PARAM REPSQ
```

```

REPLICAT REPSQ
SOURCEDEFS ./dirdef/MYDEFS.DEF
USERID <login>, PASSWORD <password>
MAP <owner/schema>.TCUSTMER, TARGET <owner/schema>.TCUSTMER_REPORT,
  SQLEXEC (ID state_lookup,
    QUERY "SELECT name desc_param FROM STATECODE WHERE STATE_CODE = :code_param",
    PARAMS (code_param = STATE)),
  COLMAP (USEDEFAULTS,
    STATE_NAME = @getval (state_lookup.desc_param));

```

7. Start both Extract and Replicat processes.

Start the Extract on the <source> system.

```
GGSCI> START EXTRACT *SQ
```

Start the Replicat on the <target> system.

```
GGSCI> START REPLICAT *SQ
```

8. Generate inserts and validate results

Generate inserts into TCUSTMER on the <source> system.

```
SQL> @demo_ora_insert
```

Validate the results in TCUSTMER_REPORT on the <target> system.

Using stored procedures

9. Create stored procedure

Using the concepts learned in the previous lab, create a new stored procedure that performs the same query that passes the state name back to the SQLEXEC to be used in target column mapping.

10. Stop Extract and Replicat

Stop the Extract and Replicat groups used in the previous lab.

On the <source>:

```
GGSCI> STOP EXTRACT *SQ
```

On the <target>:

```
GGSCI> STOP REPLICAT *SQ
```

11. Truncate data

Truncate the data in the <source> and <target> tables.

On the <source>:

```
SQLI> TRUNCATE TABLE TCUSTMER;
```

On the <target>:

```
SQLI> TRUNCATE TABLE TCUSTMER_REPORT;
```

12. Create and configure capture group

Perform the following commands on the <source> system.

- Create an Extract group named EXTSP

```
GGSCI> ADD EXTRACT EXTSP, TRANLOG, BEGIN NOW
```

- Create an Extract parameter file that extracts data from the TCUSTMER table and saves the data in a trail named ./dirdat/SP

```
GGSCI> EDIT PARAM EXTSP
```

```
EXTRACT EXTSP
USERID <login>, PASSWORD <password>
RMTHOST <target>, MGRPORT <port>
RMTTRAIL ./dirdat/sp
TABLE <owner/schema>.TCUSTMER;
```

- Define the SP trails to be 2 megabytes in size.

```
GGSCI> ADD RMTTRAIL ./dirdat/sp, megabytes 2, EXTRACT EXTSP
```

13. Create and configure delivery group

Perform the following commands on the <target> system.

- Create a Replicat group named REPSP which reads the SP trail.

```
GGSCI> ADD REPLICAT REPSP, EXTTRAIL ./dirdat/sp
```

- Create a SQL script that creates a stored procedure named STATE_NAME_LOOKUP

```
create or replace procedure state_name_lookup
(code_param in varchar2, state_name out varchar2)
IS
BEGIN
SELECT name INTO state_name FROM statecode
WHERE state_code = code_param;
END state_name_lookup;
```

If the instructor has provided sample scripts:

```
sqlplus userid/password @demo_state_lookup_proc.sql
```

- Create a Replicat parameter file that maps TCUSTMER into TCUSTMER_REPORT

```
GGSCI> EDIT PARAM REPSP
```

```
REPLICAT REPSP
SOURCEDEFS ./dirdef/MYDEFS.DEF
USERID <login>, PASSWORD <password>
MAP <owner/schema>.TCUSTMER, TARGET
    <owner/schema>.TCUSTMER_REPORT
    SQLEXEC (SPNAME state_name_lookup, PARAMS (code_param = state)),
    COLMAP (USEDEFAULTS,
        STATE_NAME =@GETVAL (state_name_lookup.state_name));
```

Instructor _____

[illegible]

[illegible]