Oracle GoldenGate 12*c*: Fundamentals for Oracle

Activity Guide

D84357GC10

Edition 1.0

January 2014

D85202

ORACLE

Author

Steve Friedberg

Technical Contributors and Reviewers

Mack Bell
Joseph deBuzna
Pete Daly
Joe Greenwald
Steven George
Susan Jang
Naoki Kato
Juan Quezada Nunez
Randy Richeson
Doug Reid
Ranbir Singh
Jinyu Wang
Volker Zell

Editors

Aju Kumar
Richard Wallis

Graphic Designer

Seema Bopaiah

Publishers

Pavithran Adka
Jayanthy Keshavamurthy
Srividya Rameshkumar

Table of Contents

Practices for Lesson 1: Introduction 1-1
Practices for Lesson 1 1-2
Practices for Lesson 2: Technology Overview 2-1
Practices for Lesson 2 2-2
Practices for Lesson 3: Oracle GoldenGate Architecture 3-1
Practices for Lesson 3 3-2
Practices for Lesson 4: Installing Oracle GoldenGate 4-1
Practices for Lesson 4: Overview 4-2
Practice 4-1: Preparation and Installation 4-3
Practice 4-2: Introduction to the GGSCI Command Interface 4-8
Practices for Lesson 5: Configuration Overview and Preparing the Environment 5-1
Practices for Lesson 5: Overview 5-2
Practice 5-1: Preparing Your Environment 5-3
Practice 5-2: Creating the GLOBALS Parameter File 5-9
Practice 5-3: Configuring and Starting Oracle GoldenGate Manager 5-10
Practice 5-4: Using the TranData option 5-13
Practices for Lesson 6: Configuring Change Capture 6-1
Practices for Lesson 6: Overview 6-2
`Practice 6-1: Setting Up the Extract and ExtTrail` 6-3
Practice 6-2: Setting Up an Extract Data Pump and Starting the Two Extracts 6-6
Practices for Lesson 7: Configuring Change Delivery 7-1
Practices for Lesson 7: Overview 7-2
Practice 7-1: Setting Up the Checkpoint Table on the Target System 7-3
Practice 7-2: Setting Up Replicat Delivery 7-5
Practice 7-3: Generating Data and Starting GoldenGate Processes 7-7
Practice 7-4: Stopping Processes and Checking Statistics 7-12
Practices for Lesson 8: Extract Trail and Files 8-1
Practices for Lesson 8: Overview 8-2
Practice 8-1: Using the logdump Utility 8-3
Practices for Lesson 9: Configuring Initial Load 9-1
Practices for Lesson 9: Overview 9-2
Practice 9-1: Setting Up the Initial Load by Using the File to Replicat Method 9-3
Practice 9-2: Setting Up the Initial Data Load by Using the Direct Load Method 9-8
Practice 9-3: Putting it All Together 9-13
Practice 9-4: Converting from Classic Mode to Integrated Mode 9-18
Practices for Lesson 10: Oracle GoldenGate Parameters 10-1
Practices for Lesson 10: Overview 10-2
Practice 10-1: Modifying Source Manager Parameters 10-3
Practice 10-2: Modifying the Target Manager Parameters 10-5
Practice 10-3: Modifying the Extract Parameters on the Source Database 10-7
Practices for Lesson 11: Data Selection and Filtering 11-1
Practices for Lesson 11: Overview 11-2
Practice 11-1: Increasing Performance by Splitting Replication Loads 11-3
Practice 11-2: Increasing Performance by Coordinated Applies 11-11

Practices for Lesson 12: Additional Transformation Topics....12-1
Practices for Lesson 12: Overview....12-2
Practice 12-1: Modifying an Existing Set of Macros and Using User Tokens....12-3
Practices for Lesson 13: Configuration Options....13-1
Practices for Lesson 13: Overview....13-2
Practice 13-1: Setting Up the Database and Source Definitions File....13-3
Practice 13-2: Oracle GoldenGate Encryption Using ENCKEYS....13-5
Practice 13-3: Generating Transactions and Validating Results....13-12
Practice 13-4: Encryption Using Wallets....13-18
Practices for Lesson 14: Bidirectional Replication....14-1
Practices for Lesson 14: Overview....14-2
Practice 14-1: Two-Way Active-Active Data Replication....14-3
Practice 14-2: Conflict Detection and Resolution....14-15
Practices for Lesson 15: DDL Replication....15-1
Practices for Lesson 15: Overview....15-2
Practice 15-1: DDL Replication Database Setup....15-3
Practice 15-2: DDL Replication Oracle GoldenGate Setup....15-7

Practices for Lesson 1: Introduction

Chapter 1

Practices for Lesson 1

There are no practices for this lesson.

Practices for Lesson 2: Technology Overview

Chapter 2

Practices for Lesson 2

There are no practices for this lesson.

Practices for Lesson 3: Oracle GoldenGate Architecture

Chapter 3

Practices for Lesson 3

There are no practices for this lesson.

Practices for Lesson 4: Installing Oracle GoldenGate

Chapter 4

Practices for Lesson 4: Overview

Practices Overview

In these practices, you become familiar with the lab environment that you will use for the duration of the class. You also install the Oracle GoldenGate software to be used with both the AMER and EURO databases. The software zip file is already downloaded and staged. Then you are briefly introduced to the command-line interface (GGSCI).

Practice 4-1: Preparation and Installation

Overview

There are two databases configured on your machine: AMER and EURO. These are physically different databases, each with its own structure and set of users. In the practices for this course, you sometimes need to connect to one of your databases by using SQL*Plus. The following steps describe how to connect to each of your databases and also provide more details about the users and tables that you will be working with.

Each database has administrative users, such as `system`, as well as other users created specifically for this course. The data tables that you create and populate in the AMER database that you use in this course are owned by the WEST user. Conversely, the data tables in the EURO database are owned by the EAST user.

Even though there is only one PC for each practice team, there are two host names defined in the `/etc/hosts` file: `easthost` and `westhost`. You should be able to ping both hosts, and you should use those names in place of `localhost` so that you can clearly indicate to yourself source and target hosts.

To connect to the AMER database as the `system` user by using SQL*Plus, enter the following command at the operating system prompt:

```
[OS_prompt ~] sqlplus system@amer
```

The password for all database accounts for this course is **`oracle_4U`**.

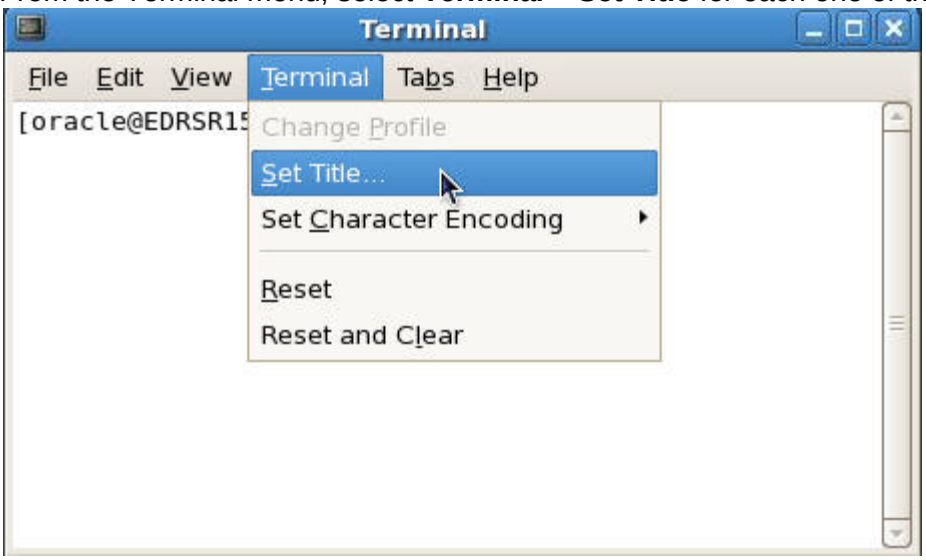
Operating System	
Type	**`Linux`**
<*userid*>	**`oracle`**
<*password*>	**`oracle`**
AMER Database (Source)	
<*login*>	**`west`**
<*password*>	**`oracle_4U`**
<*oracle_sid*>	**`amer`**
$GG_AMER_HOME	**`/u01/app/oracle/product/gg_amer`**
Manager <*port*>	**`15000`**

EUR0 Database (Target)	
<*login*>	**east**
<*password*>	**oracle_4U**
<*oracle_sid*>	**euro**
$GG_EURO_HOME	**/u01/app/oracle/product/gg_euro**
Manager <*port*>	**15001**
GoldenGate	
<*software location*>	**/stage**
GoldenGate <*install location - source*>	**/u01/app/oracle/product/gg_amer**
GoldenGate <*install location - target*>	**/u01/app/oracle/product/gg_euro**

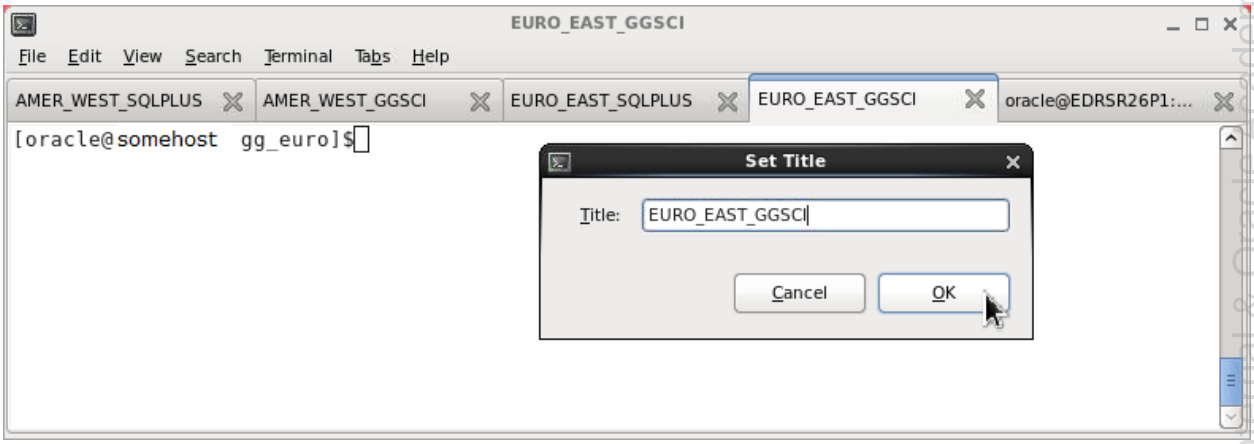
1. To set up your working environment, create terminal windows to make it easier to navigate between the source and target databases:
   a. Log in to your designated Linux machine with the following information:
      1) Username: **oracle**
      2) Password: **oracle**
2. Verify that a local copy of the documentation is accessible on your PC.
   a. Go to the Linux desktop.
   b. Start Adobe Reader. (Click its launcher icon.)
   c. Click **File > Open** and select the Release Notes in `/home/oracle/Documents`, that is, `ggate12c_release_notes_e40012.pdf`. This is only to verify that you have a working set of documentation.
   d. Close the Release Notes, but leave Adobe Reader running minimized. You will need it later, but not now.
3. Set up your terminal environment.
   a. Double-click the Terminal icon located on your Linux desktop.

   b. Expand it to full screen.
   c. Click **File > Open Tab** four times so that you have five open tabs.

d. From the Terminal menu, select **Terminal > Set Title** for each one of the tabs.



e. Set the name of the terminal tabs from left to right to be `AMER_WEST_SQLPLUS`, `AMER_WEST_GGSCI`, `EURO_EAST_SQLPLUS`, `EURO_EAST_GGSCI`, and nothing for the last one (which will dynamically show the current directory). It should look similar to this:



Leave these terminal tabs open for the entire course.

4. On the fifth (unnamed) tab, unzip the OGG installation files from the stage directory into the temp directory.

```
oracle@somehost ~
[OS_prompt ~] cd /tmp
[OS_prompt tmp] mkdir ogg
[OS_prompt tmp] cd ogg
[OS_prompt tmp] unzip /stage/1212*
[OS_prompt tmp] cd fbo*
[OS_prompt tmp] cd Disk1
[OS_prompt tmp] ls
install  response  runInstaller  stage
```

5. Run the Oracle Universal Installer (OUI) to create the AMER instance.
   a. At the OS prompt, run **`./runInstaller`** to launch the GUI.
   b. At the Installation Option panel, accept the default of Oracle 12c. Click Next.
   c. In the Installation Details panel, change the software location to **`/u01/app/oracle/product/gg_amer`**, leave the Database Location as its default, and change Manager Port to **15000**. Click Next.
   d. In the Summary panel, click Install. It takes about a minute.
   e. In the Finish panel, click Close.
6. Run the Oracle Universal Installer (OUI) again to create the EURO instance.
   a. At the OS prompt, re-run **`./runInstaller`** to launch the GUI a second time.
   b. In the Installation Option panel, accept the default of Oracle 12c. Click Next.
   c. In the Installation Details panel, change the software location to **`/u01/app/oracle/product/gg_euro`**, leave the Database Location at its default, and change Manager Port to **15001**. Click Next.
   d. In the Summary panel, click Install. It takes about a minute.
   e. In the Finish panel, click Close.
7. On the AMER_WEST_SQLPLUS tab, set the Oracle database environment variables for `amer`. Note the leading dot (period) before the `oraenv` command. Enter the following commands:

```
[OS_prompt ~] cd ~/labs
[OS_prompt labs] . oraenv
ORACLE_SID = [amer] ? amer
The Oracle base remains unchanged with value /u01/app/oracle
```

   Note: Because the current value is already `amer`, you could accept it without change by pressing Enter.
8. On the AMER_WEST_GGSCI tab, set the working directory and the Oracle database environment variables for `amer`. Note the leading dot (period) before the `oraenv` command. Enter the following commands:

```
[OS_prompt ~] cd $GG_AMER_HOME
[OS prompt gg_amer] pwd
/u01/app/oracle/product/gg_amer
[OS prompt gg_amer] . oraenv
ORACLE_SID = [amer] ? amer
The Oracle base remains unchanged with value /u01/app/oracle
```

   Note: Because the current value is already `amer`, you could accept it without change by pressing Enter.
9. On the EURO_EAST_SQLPLUS tab, set the Oracle database environment variables for `euro`. Note the leading dot (period) before the `oraenv` command. Enter the following commands:

```
[OS_prompt ~] cd ~/labs
[OS_prompt labs] . oraenv
ORACLE_SID = [amer] ? euro
The Oracle base remains unchanged with value /u01/app/oracle
```

Note: The first time, you must change the value to `euro`. But if you ever run the command again in the same window or on the same tab, you can accept `euro` without change by pressing Enter.

10. On the `EURO_EAST_GGSCI` tab, set the working directory and the Oracle database environment variables for `euro`. Note the leading dot (period) before the `oraenv` command. Enter the following commands:

```
[OS_prompt ~] cd $GG_EURO_HOME
[OS prompt gg_euro] pwd
/u01/app/oracle/product/gg_euro
[OS prompt gg_euro] . oraenv
ORACLE_SID = [amer] ? euro
The Oracle base remains unchanged with value /u01/app/oracle
```

Note: The first time, you must change the value to `euro`. But if you ever run the command again in the same window or on the same tab, you can accept `euro` without change by pressing Enter.

This completes Practice 4-1. Continue now with Practice 4-2.

Practice 4-2: Introduction to the GGSCI Command Interface

Overview

In this practice, you learn to invoke the Oracle GoldenGate command interface (GGSCI) and how to access online Help for all commands. This can be done using either the AMER database or the EURO database.

Assumptions

The interchangeability of the databases assumes that they are not currently running any GGSCI commands.

Tasks

1. How you invoke GGSCI depends on the settings in your profile. On the fifth tab `oracle@hostname`, from the `/home/oracle` directory, review the `.bashrc` file. Make certain that the `ORACLE_HOME`, `GG_AMER_HOME`, and `GG_EURO_HOME` system environment variables are set to the correct directories. The Oracle GoldenGate processes refer to them when connecting to the database. If the appropriate settings have been included, you can invoke GGSCI directly from the command line.

```
[OS_prompt] more ~/.bashrc
# .bashrc

# User specific aliases and functions
# alias ll='ls -l'
alias rlggsci='rlwrap ./ggsci'
alias rlsqlplus='rlwrap sqlplus'

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

ORACLE_BASE=/u01/app/oracle
ORACLE_HOME=${ORACLE_BASE}/product/12.1.0/dbhome_1
GG_AMER_HOME=${ORACLE_BASE}/product/gg_amer
GG_EURO_HOME=${ORACLE_BASE}/product/gg_euro
GG_DS_HOME=${ORACLE_BASE}/product/gg_director_server
GG_DC_HOME=${ORACLE_BASE}/product/gg_director_client
GG_MONI_HOME=${ORACLE_BASE}/product/gg_moni
JAVA_HOME=/usr/java/jdk1.7.0_40
PATH=${JAVA_HOME}/bin:${ORACLE_HOME}/bin:${GG_AMER_HOME}:${GG_EURO_HOME}:${PAT
H}:${HOME}/bin
LD_LIBRARY_PATH=${JAVA_HOME}/jre/lib/amd64/server/:${ORACLE_HOME}/lib

ORACLE_SID=amer
PATH=${ORACLE_HOME}/bin:${PATH}:${LD_LIBRARY_PATH}
CLASSPATH=${ORACLE_HOME}/jdbc/lib/ojdbc7.jar:${ORACLE_HOME}/jlib/orai18n.jar

export PATH
export ORACLE_BASE ORACLE_HOME ORACLE_SID GG_AMER_HOME GG_EURO_HOME
export GG_DC_HOME GG_DS_HOME GG_MONI_HOME
export CLASSPATH LD_LIBRARY_PATH JAVA_HOME
[OS_prompt]
```

Note: Make sure the PATH= statement is all on one line. Some of the Management Pack variables for Director and Monitor are set as well. They are not used in this course.

2. GGSCI does not have an easy native command-line editor, so a read-line wrap (`rlwrap`) package has been installed, and aliases set up to make it easier for the class. This is not part of the normal GGSCI function! Launch the GGSCI command interface by using either the alias (easier) or the official (harder) way. On the source `AMER_WEST_GGSCI` tab, enter the following commands:

```
[OS_prompt gg_amer] ./ggsci              either this or...
[OS_prompt gg_amer] rlggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.0.0 17185003 OGGCORE_12.1.2.0.0_PLATFORMS_130924.1316_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.
Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.

GGSCI (AMER) 1>
```

3. View a `HELP` summary for all commands:

```
GGSCI (AMER) 1> Help

GGSCI Command Summary:

Object:          Command:
SUBDIRS          CREATE
DATASTORE        ALTER, CREATE, DELETE, INFO, REPAIR
ER               INFO, KILL, LAG, SEND, STATUS, START, STATS, STOP
EXTRACT          ADD, ALTER, CLEANUP, DELETE, INFO, KILL,
                 LAG, REGISTER, SEND, START, STATS, STATUS, STOP
                 UNREGISTER
EXTTRAIL         ADD, ALTER, DELETE, INFO
GGSEVT           VIEW
JAGENT           INFO, START, STATUS, STOP
MANAGER          INFO, SEND, START, STOP, STATUS
MARKER           INFO
PARAMS           EDIT, VIEW
REPLICAT         ADD, ALTER, CLEANUP, DELETE, INFO, KILL, LAG, SEND,
                 START, STATS, STATUS, STOP, SYNCHRONIZE
REPORT           VIEW
RMTTRAIL         ADD, ALTER, DELETE, INFO
TRACETABLE       ADD, DELETE, INFO
TRANDATA         ADD, DELETE, INFO
SCHEMATRANDATA   ADD, DELETE, INFO
CHECKPOINTTABLE  ADD, DELETE, CLEANUP, INFO, UPGRADE
WALLET           CREATE, OPEN, PURGE
MASTERKEY        ADD, INFO, RENEW, DELETE, UNDELETE
CREDENTIALSTORE  ADD, ALTER, INFO, DELETE

Commands without an object:
(Database)       DBLOGIN, LIST TABLES, ENCRYPT PASSWORD, FLUSH SEQUENCE
                 MININGDBLOGIN
```

```
(DDL)            DUMPDDL
(Miscellaneous)  DEFAULTJOURNAL, FC, HELP, HISTORY, INFO ALL, OBEY,
                 SET EDITOR, SHELL, SHOW, VERSIONS, !
                 (note: type the word COMMAND after the ! to display the
                 ! help topic, for example: GGSCI (sys1)> help ! command

For help on a specific command, type HELP <command> <object>.
Example: HELP ADD REPLICAT

GGSCI (AMER) 2> Help All
```

The last three rows in the command summary (`Wallet, MasterKey, CredentialStore`) are new with 12*c*, and `JAGENT` is for Veridata.

4. View a `HELP` summary for a command or entity (some of the screenshots are too long to include):

```
GGSCI (AMER) 3> Help Add Extract
GGSCI (AMER) 4> Help Add ExtTrail
```

Unfortunately, there is no way to pipe the very long help output into `more`, or some kind of page-at-a-time pause command (though you could simply "`sh gedit help.txt`" and search and scroll).

5. View your command history:

```
GGSCI (AMER) 5> History

GGSCI Command History

    1: Help
    2: Help All
    3: Help Add Extract
    4: Help Add ExtTrail
    5: History

GGSCI (AMER) 6>
```

To repeat a command (for example, the third one), enter exclamation (bang) and the line number, as in the following example:

```
!3
```

Note that the up/down arrow recall of `rlwrap` (alias `rlggsci`) is not part of the normal function of the product, nor is it supported, but is still quite handy.

6. View a brief informational summary of all processes:

```
GGSCI (AMER) 6> Info All

Program     Status      Group       Lag at Chkpt  Time Since Chkpt
MANAGER     RUNNING

GGSCI (AMER) 7>
```

The Installer started the Manager for you. You will run this `Info All` command many times as the practices progress. When other processes are configured and running, the report becomes more interesting.

7. Do not exit GGSCI unless explicitly instructed to do so. (There is no harm; it is just a bother to keep restarting it.) Leave it running for the entire class.

This completes Practice 4-2. This completes the practices for Lesson 4. Stop here.

Practices for Lesson 5: Configuration Overview and Preparing the Environment

Chapter 5

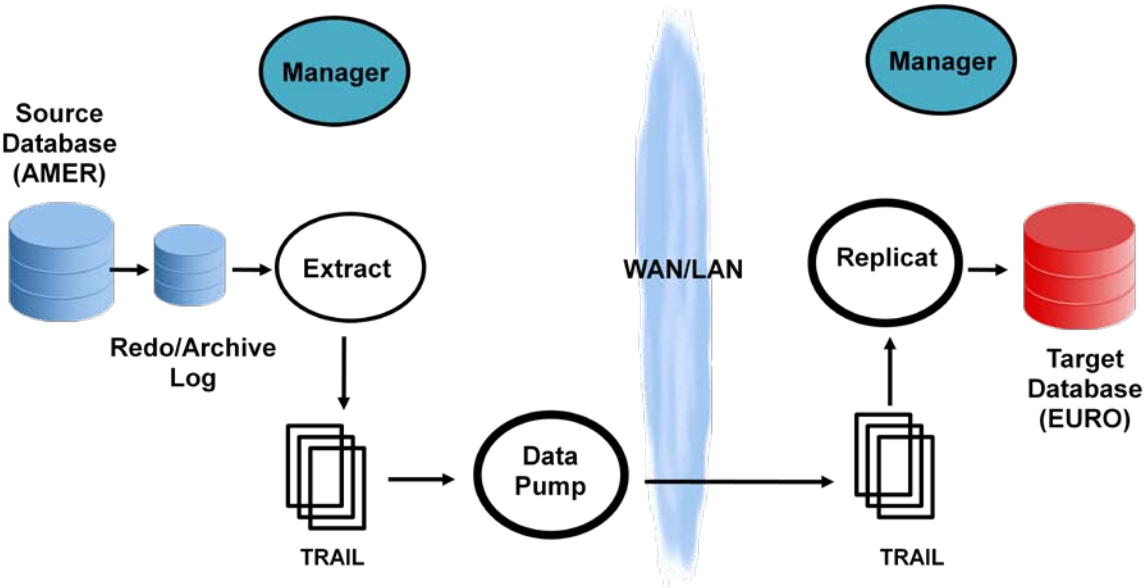
Practices for Lesson 5: Overview

Practices Overview

In these practices, you begin the process of setting up one-way data replication by creating and populating your databases. You also configure and start the Manager process on both instances of Oracle GoldenGate. Oracle GoldenGate provides log-based change data capture (CDC) and replication of committed database transactions. The software provides capture, routing, transformation, and delivery of transactional data across heterogeneous environments in real time.

Big Picture

The following figure depicts the simplified Oracle GoldenGate implementation that you create. In this practice, you create the objects that are shaded (source database, target database, and `mgr.prm` files):



Practice 5-1: Preparing Your Environment

Overview

In this practice, you set up the databases (AMER and EURO) and configure Oracle GoldenGate to work specifically with the Oracle databases.

Note: Some parts of the screenshot feedback may be in a smaller font for clarity. However, the commands that you should type are always indicated by full-sized, **bold** monospace font.

Assumptions

Two core databases are already installed: AMER and EURO. The AMER database (the source database) has the west/oracle_4U user already created, and the EURO database has the east/oracle_4U user also already created. In this practice, you run the source_database.sql script on the AMER database and the target_database.sql script on the EURO database. Both of these scripts create the following tables:

- ACCOUNT
- ACCOUNT_TRANS
- BRANCH
- TELLER
- TELLER_TRANS
- BRANCH_ATM

The data tables in the AMER database are owned by the WEST user. The data tables in the EURO database are owned by the EAST user.

You have already set the environment variables with oraenv for amer and euro. The SQL*Plus tabs are in the /home/oracle/labs working directory.

Tasks

1. Switch to the AMER_WEST_SQLPLUS tab. To create the source database for AMER, at the OS prompt, issue the following command:

```
[OS_prompt labs] ls Section5/sqlscripts/
seed_database.sql  source_database.sql  target_database.sql
[OS_prompt labs] sqlplus west@amer/oracle_4U
  @Section5/sqlscripts/source_database.sql

SQL*Plus: Release 12.1.0.1.0 Production on Wed Nov 20 13:42:30 2013
Copyright (c) 1982, 2013, Oracle.  All rights reserved.
Last Successful login time: Tue Nov 19 2013 12:39:20 +00:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options

Table dropped.
... many lines omitted for clarity ...
```

   As with many Oracle PL/SQL scripts, a first attempt to drop a non-existent table gives an "error" that the table or view does not exist. For these setup scripts, you can ignore those messages.

2. To populate the tables on the AMER database, run the following command from the same directory (/home/oracle/labs/Section5/sqlscripts):

```
[OS_prompt labs] sqlplus west@amer/oracle_4U
  @Section5/sqlscripts/seed_database.sql

SQL*Plus: Release 12.1.0.1.0 Production on Wed Nov 20 13:46:47 2013
Copyright (c) 1982, 2013, Oracle.  All rights reserved.
Last Successful login time: Wed Nov 20 2013 13:42:30 +00:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options

Table truncated.
... many lines omitted for clarity ...
```

3. To create the data tables on the EURO database, on the EURO_EAST_SQLPLUS tab, execute the following commands:

```
[OS_prompt labs] sqlplus east@euro/oracle_4U
  @Section5/sqlscripts/target_database.sql

SQL*Plus: Release 12.1.0.1.0 Production on Wed Nov 20 13:50:54 2013
Copyright (c) 1982, 2013, Oracle.  All rights reserved.
Last Successful login time: Wed Nov 06 2013 13:58:56 +00:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options

Table dropped.
... many lines omitted for clarity ...
```

Note that you do not populate the target tables with SQL scripts; populating (replicating) will be Oracle GoldenGate's job.

4. ***Don't do this step—just read it!*** Create an Oracle GoldenGate user that can connect to the source and target databases for transactional data. An Oracle GoldenGate user requires a database user with at least the following privileges:

User Privilege	Extract (Source Side)	Replicat (Target Side)
CREATE SESSION, ALTER SESSION	X	X
RESOURCE	X	X
SELECT ANY DICTIONARY	X	X
FLASHBACK ANY TABLE or FLASHBACK ON <*owner.table*>	X	
SELECT ANY TABLE or SELECT ON <*owner.table*>	X	X
INSERT, UPDATE, DELETE ON <*target tables*>		X
CREATE TABLE		X

User Privilege	Extract (Source Side)	Replicat (Target Side)
EXECUTE on DBMS_FLASHBACK package	X	

In addition to the above privileges, you would normally run:
```
SQL> EXEC DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE('GGUSER');
```

(Optional) To learn about the DBMS_GOLDENGATE_AUTH syntax, enter the following commands:

```
[OS_prompt labs] sqlplus / as sysdba
SQL> set pages 100
SQL> desc dbms_goldengate_auth
SQL> SELECT text FROM all_source
     WHERE name='DBMS_GOLDENGATE_AUTH';
(shows "help" syntax text at top by convention, may have to
scroll up to see it.)
```

Instead of doing all that work, the next step will "cheat" for purposes of this practice.

5. Create a DBA-level user (gguser/oracle_4U) on both the AMER and EURO databases. To make the practices easier in the classroom, you give ggsuser the DBA role (which is more than what would have been granted in the previous step). In a real-world situation, you would not grant that many privileges.

   For the AMER database, on the AMER_WEST_SQLPLUS tab:

```
[OS_prompt labs] sqlplus / as sysdba

SQL> sho parameter db_name

NAME            TYPE      VALUE
--------------- --------- ----------------
db_name         string    amer

SQL> CREATE TABLESPACE oggtabsp DATAFILE
     '/u01/app/oracle/oradata/amer/oggtabsp.dbf'  SIZE 50m
     AUTOEXTEND ON;
Tablespace created.

SQL> CREATE USER gguser  IDENTIFIED BY oracle_4U  DEFAULT
     TABLESPACE oggtabsp;
User created.

SQL> GRANT dba TO gguser;
Grant succeeded.

SQL> exit
```

   The separate tablespace is required only for the DDL replication in Practice 15; otherwise, the user could be anywhere.

For the EURO database, on the EURO_EAST_SQLPLUS tab:

```
[OS_prompt labs] sqlplus / as sysdba

SQL> sho parameter db_name

NAME            TYPE      VALUE
--------------- --------- ----------------
db_name         string    euro

SQL> CREATE TABLESPACE oggtabsp DATAFILE
     '/u01/app/oracle/oradata/euro/oggtabsp.dbf'  SIZE 50m
     AUTOEXTEND ON;
Tablespace created.

SQL> CREATE USER gguser  IDENTIFIED BY oracle_4U  DEFAULT
     TABLESPACE oggtabsp;
User created.

SQL> GRANT dba TO gguser;
Grant succeeded.

SQL> exit
```

6. On the AMER_WEST_SQLPLUS tab, log out of SQL sys and then log in again with the new DBA user, gguser. Minimal supplemental logging must be enabled at the database level to allow Oracle GoldenGate to properly capture updates to primary keys and chained rows. To enable supplemental logging at the database level, do the following:
   a. Log in to SQL*Plus as gguser/oracle_4U on AMER. Use rlsqlplus so that you get up/down arrow history and editing.
   b. Enable minimal supplemental logging by executing the following commands:
      ```
      ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
      ALTER DATABASE FORCE LOGGING;
      ```
   c. To ensure that the redo and archive logs contain supplemental log data, switch the logs by executing the following commands:
      ```
      ALTER SYSTEM SWITCH LOGFILE;
      ```
   d. Verify that supplemental logging is enabled at the database level via the following command. (**Note:** Output of the query must be YES or IMPLICIT.)
      ```
      SELECT supplemental_log_data_min, force_logging
        FROM v$database;
      ```

The results should look similar to this:

```
[OS_prompt labs] rlsqlplus gguser@amer/oracle_4U

SQL> set sqlprompt AMER_SQL>

AMER_SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
Database altered.

AMER_SQL> ALTER DATABASE FORCE LOGGING;
Database altered.

AMER_SQL> ALTER SYSTEM SWITCH LOGFILE;
System altered.

AMER_SQL> SELECT supplemental_log_data_min, force_logging
           FROM v$database;

SUPPLEME FORCE_LOGGING
-------- ---------------------------------
YES      YES

AMER_SQL>
```

7. On the EURO_EAST_SQLPLUS tab, log out of SQL sys and then log in again with the new user, gguser. Minimal supplemental logging must be enabled at the database level to allow Oracle GoldenGate to properly capture updates to primary keys and chained rows. To enable supplemental logging at the database level, perform the following:
   a. Log in to SQL*Plus as gguser/oracle_4U on EURO. Use rlsqlplus so that you get up/down arrow history and editing.
   b. Enable minimal supplemental logging by executing the following commands:
      ```
      ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
      ALTER DATABASE FORCE LOGGING;
      ```
   c. To ensure that the redo and archive logs contain supplemental log data, switch the logs by executing the following command:
      ```
      ALTER SYSTEM SWITCH LOGFILE;
      ```
   d. Verify that supplemental logging is enabled at the database level via the following command: (The output of the query must be YES or IMPLICIT.)
      ```
      SELECT supplemental_log_data_min, force_logging
        FROM v$database;
      ```

The results should look similar to this:

```
[OS_prompt labs] rlsqlplus gguser@euro/oracle_4U

SQL> set sqlprompt EURO_SQL>

EURO_SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
Database altered.

EURO_SQL> ALTER DATABASE FORCE LOGGING;
Database altered.

EURO_SQL> ALTER SYSTEM SWITCH LOGFILE;
System altered.

EURO_SQL> SELECT supplemental_log_data_min, force_logging
           FROM v$database;

SUPPLEME FORCE_LOGGING
-------- ---------------------------------
YES      YES

EURO_SQL>
```

This completes Practice 5-1. Continue now with Practice 5-2.

Practice 5-2: Creating the GLOBALS Parameter File

Overview

On the target side, Replicat checkpoints to a table in the database that is defined in a special parameter file named GLOBALS. The file name must be all UPPERCASE with no extension. In this practice, you create a GLOBALS parameter file located on the target database (EURO).

Tasks

1. On the EURO_EAST_GGSCI tab, create a GLOBALS parameter file by executing the following commands. The oraenv variables should already be set.

```
[OS_prompt ~] cd $GG_EURO_HOME
[OS_prompt gg_euro] gedit GLOBALS
```

2. Start gedit (or vi if you prefer) to open a new GLOBALS file located in the directory of your Oracle GoldenGate instance. Add the following runtime parameters to the GLOBALS file, which you use in a later practice:

```
CheckpointTable GGS_CHECKPOINT
```

3. Save and close the file.

   Note: There is no harm or benefit in creating a GLOBALS file on the source side as well; however, it is not necessary at this time. Other Oracle GoldenGate products, such as Monitor from the Management Pack, also make use of the GLOBALS file.

This completes Practice 5-2. Continue now with Practice 5-3.

Practice 5-3: Configuring and Starting Oracle GoldenGate Manager

Overview

In this practice, you configure and stop/restart the Manager process on both instances of Oracle GoldenGate. The Oracle GoldenGate Manager process performs several important tasks that you must configure:

- Listens for incoming connectivity requests from Extract or Extract data pump
- Performs housekeeping functions to delete any consumed Oracle GoldenGate trails

Tasks

1. The first task is to configure and stop/restart the Manager processes on both the source and target. On the `AMER_WEST_GGSCI` tab, GGSCI should still be started from a previous lab. If so, skip to step 2; if not, do the following:

```
[OS_prompt ~] cd $GG_AMER_HOME
[OS_prompt gg_amer] . oraenv
ORACLE_SID = [amer] ?
The Oracle base remains unchanged with value /u01/app/oracle
[OS_prompt gg_amer] rlggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.0.0 17185003 OGGCORE_12.1.2.0.0_PLATFORMS_130924.1316_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.
Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.

GGSCI (AMER) 1>
```

2. **(Optional)** If you want to change the text editor for this session, enter:

```
GGSCI (AMER)> Set Editor gedit
```

   If you do not do this `Set`, you get `vi` as the default editor for Linux. You cannot change the startup default editor.

3. At the GGSCI prompt, enter:

```
GGSCI (AMER)> Edit Param mgr
```

4. The text editor starts and opens a new `mgr.prm` file. Add the following runtime parameter to the `mgr.prm` file below the existing Port line (created by the installer):

```
Port 15000
PurgeOldExtracts ./dirdat/*, UseCheckpoints
```

   Do not save it yet. You have another line to add.

5. Using the Oracle GoldenGate Reference Manual (`ggate12c_reference_oracle_e29392.pdf`) located in `~/Documents`, what parameter is used to specify ports that Manager can allocate for dynamic communications?
   a. Add that parameter to the `mgr.prm` file, defining the `DynamicPortList` range to be 15100 through 15150.
   b. Save and close the file.

6. To make the new parameters take effect, at the GGSCI prompt, bounce (stop and restart) the Manager process by entering the following commands:

```
GGSCI (AMER)> Stop mgr
GGSCI (AMER)> Start mgr
```

To view the status of the Manager process, enter this command:

```
GGSCI (AMER)> Info mgr
```

The results should look similar to the following: (Your host name and PID will be different.)

```
GGSCI (AMER)> Stop Manager
Manager process is required by other GGS processes.
Are you sure you want to stop it (y/n)? y

Sending STOP request to MANAGER ...
Request processed.
Manager stopped.

GGSCI (AMER)> Start Manager
Manager started.

GGSCI (AMER)> Info Manager
Manager is running (IP port WESTHOST.15000, Process ID 7992).

GGSCI (AMER)>
```

Notice that you can substitute `mgr` for `Manager`, and you can add an exclamation mark (!) at the end of the `Stop` command to suppress the "Are you sure?" prompt (useful for scripts).

7. On the `EURO_EAST_GGSCI` tab, from the shell prompt positioned inside the Oracle GoldenGate directory, invoke GGSCI. Execute the following commands:

```
[OS_prompt ~] cd $GG_EURO_HOME
[OS_prompt gg_euro] . oraenv
ORACLE_SID = [euro] ? euro
The Oracle base remains unchanged with value /u01/app/oracle
[OS_prompt gg_euro] rlggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.0.0 17185003 OGGCORE_12.1.2.0.0_PLATFORMS_130924.1316_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.
Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.

GGSCI (EURO) 1>
```

8. Optionally, set the default editor. Edit the manager parameter file:

```
GGSCI (EURO)> Edit Param mgr
```

Add the following runtime parameters to the `mgr.prm` file:

```
Port 15001
PurgeOldExtracts ./dirdat/*, UseCheckpoints
```

Save and close the file.

9. To make the new parameters take effect, at the GGSCI prompt, bounce (stop and restart) the Manager process by entering the following commands:

```
GGSCI (EURO)> Stop mgr !
GGSCI (EURO)> Start mgr
```

Notice that the exclamation point suppresses the “Are you sure?” message. To view the status of the Manager process, enter the following command:

```
GGSCI (EURO)> Info Manager
```

The results should look similar to the following example. (Your host name and PID will be different.) Notice that the EURO port number is one greater than the AMER port number.

```
GGSCI (EURO)> Start Manager
Manager started.

GGSCI (EURO)> Info Manager
Manager is running (IP port EASTHOST.15001, Process ID 24513).

GGSCI (EURO)>
```

Do not exit GGSCI, leave it running for the entire class.

10. (Re)starting the Manager created an `MGR.rpt` in the `dirrpt` directory. Old reports get numbered 0–9. The current report is not very interesting yet, but it is a good habit to check the report often, especially for troubleshooting. You can optionally see it by entering the command:

```
GGSCI (EURO)> View Report mgr
```

If the report is very long and you want to scroll up or down, you can enter (being mindful of UPPER/lowercase file names):

```
GGSCI (EURO)> sh less dirrpt/MGR.rpt
```

This completes Practice 5-3. Continue now with Practice 5-4.

Practice 5-4: Using the `TranData` option

Overview

In this practice, you enable supplemental logging on the source database for the specified tables. Before you can start capturing real-time data, the Oracle database must be set to log the table key values whenever it logs a rows change, so that they are available to Oracle GoldenGate in Redo. By default, the database logs only those column values that change. This is required so Oracle GoldenGate can locate the correct row on the target for update and delete operations.

Tasks

1. You can add individual tables or use wildcards. On the AMER_WEST_GGSCI tab (the source server), at the GGSCI prompt, enter the following:

```
GGSCI (AMER)> DBLogin UserID gguser@amer, Password oracle_4U
GGSCI (AMER)> Add TranData WEST.ACCOUNT
GGSCI (AMER)> Add TranData WEST.*
```

   The results should be similar to the following:

```
GGSCI (AMER)> DBLogin UserID gguser@amer, Password oracle_4U
Successfully logged into database.

GGSCI (AMER)> Add TranData WEST.ACCOUNT

Logging of supplemental redo data enabled for table WEST.ACCOUNT.
TRANDATA for scheduling columns has been added on table 'WEST.ACCOUNT'.

GGSCI (AMER)> Add TranData WEST.*

Logging of supplemental redo log data is already enabled for table
WEST.ACCOUNT.

2013-12-02 13:14:07  WARNING OGG-00706  Failed to add supplemental log group
on table WEST.ACCOUNT due to ORA-32588: supplemental logging attribute primary
key exists SQL ALTER TABLE "WEST"."ACCOUNT" ADD SUPPLEMENTAL LOG DATA (PRIMARY
KEY, UNIQUE, FOREIGN KEY) COLUMNS  /* GOLDENGATE_DDL_REPLICATION */.

Logging of supplemental redo data enabled for table WEST.ACCOUNT_TRANS.
TRANDATA for scheduling columns has been added on table 'WEST.ACCOUNT_TRANS'.
Logging of supplemental redo data enabled for table WEST.BRANCH.
TRANDATA for scheduling columns has been added on table 'WEST.BRANCH'.
Logging of supplemental redo data enabled for table WEST.BRANCH_ATM.
TRANDATA for scheduling columns has been added on table 'WEST.BRANCH_ATM'.
Logging of supplemental redo data enabled for table WEST.TELLER.
TRANDATA for scheduling columns has been added on table 'WEST.TELLER'.
Logging of supplemental redo data enabled for table WEST.TELLER_TRANS.
TRANDATA for scheduling columns has been added on table 'WEST.TELLER_TRANS'.
GGSCI (AMER)>
```

Note that there is no harm in adding transaction data twice, as is the case for WEST.ACCOUNT. The warning can be ignored. If at this point you were to create a new table WEST.JUNK, it would *not* be picked up for adding transaction data even though you used the wildcard for the WEST schema. If you did want to include future tables, use Add SchemaTranData instead of TranData.

2. Execute the following GGSCI command to verify that supplemental logging is enabled on each table:

```
GGSCI (AMER)> Info TranData WEST.*
```

The results should be similar to the following:

```
GGSCI (AMER)> Info TranData WEST.*

Logging of supplemental redo log data is enabled for table WEST.ACCOUNT.
Columns supplementally logged for table WEST.ACCOUNT: ACCOUNT_NUMBER.

Logging of supplemental redo log data is enabled for table WEST.ACCOUNT_TRANS.
Columns supplementally logged for table WEST.ACCOUNT_TRANS: TRANS_NUMBER,
ACCOUNT_NUMBER, ACCOUNT_TRANS_TS.

Logging of supplemental redo log data is enabled for table WEST.BRANCH.
Columns supplementally logged for table WEST.BRANCH: BRANCH_NUMBER.

Logging of supplemental redo log data is enabled for table WEST.BRANCH_ATM.
Columns supplementally logged for table WEST.BRANCH_ATM: BRANCH_NUMBER,
ATM_TRANS_TS, TRANS_NUMBER, ATM_NUMBER.

Logging of supplemental redo log data is enabled for table WEST.TELLER.
Columns supplementally logged for table WEST.TELLER: TELLER_NUMBER.

Logging of supplemental redo log data is enabled for table WEST.TELLER_TRANS.
Columns supplementally logged for table WEST.TELLER_TRANS: TRANS_NUMBER,
TELLER_TRANS_TS, TELLER_NUMBER.

GGSCI (AMER)>
```

Leave all the GGSCI and SQL*Plus sessions running. Leave the five terminal tabs open.

This completes Practice 5-4. This completes the practices for Lesson 5. Stop here.

Practices for Lesson 6: Configuring Change Capture

Chapter 6

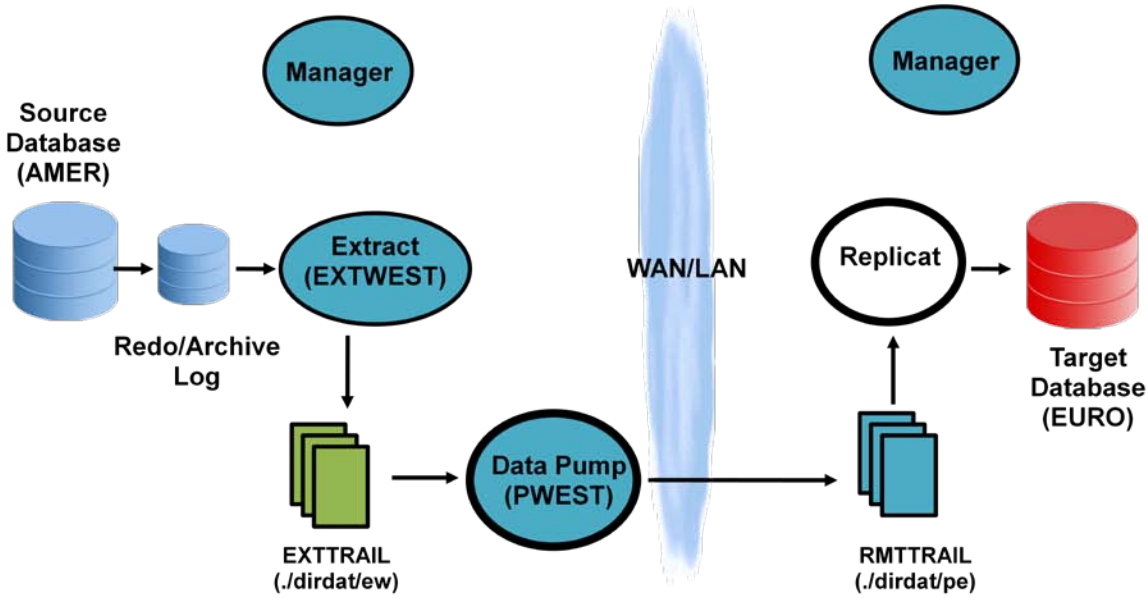
Practices for Lesson 6: Overview

Practices Overview

In these practices, you set up the Extract on the source (AMER) database, `ExtTrail`, data pump Extract, and `RmtTrail`.

Big Picture

The following diagram shows the new objects that are created: Extract, `ExtTrail`, data pump Extract, and `RmtTrail`.



The ew extract trail prefix is for ExtractWest. The pe remote trail prefix is for PumpEast. This is only a convention rather than a requirement. What is a requirement is the two-letter maximum name length.

Practice 6-1: Setting Up the Extract and `ExtTrail`

Overview

In this practice, you perform the following actions:

- Configure the primary Extract process that captures changes.
- Add the local trail that stores these changes.

Tasks

1. You should still be logged in to GGSCI in the AMER instance from the previous lab. If so, skip this task and proceed to task 2. If not, perform the following step. On the source server (on the AMER_WEST_GGSCI tab), confirm that you are in the proper installation directory and logged in to the database by using AMER SID. Start GGSCI by using the read-line utility.

```
[OS_prompt ~] cd $GG_AMER_HOME
[OS_prompt gg_amer] . oraenv
[OS_prompt gg_amer] rlggsci
GGSCI (AMER) 1> DBLogin UserID gguser@amer, Password oracle_4U
```

   Optionally, set the default text editor.

2. On the AMER_WEST_GGSCI tab, add the wallet and credential store information (password) for user gguser on both source and target systems. **WARNING:** Make sure that each `Alter CredentialStore` command is all on one line (cut and paste may make this difficult).

```
GGSCI (AMER)> Create Wallet
Created wallet at location 'dirwlt'.
Opened wallet at location 'dirwlt'.

GGSCI (AMER)> Add CredentialStore
Credential store created in ./dircrd/.

GGSCI (AMER)> Alter CredentialStore Add User gguser@amer
               Password oracle_4U Alias gguamer
Credential store in ./dircrd/ altered.

GGSCI (AMER)> Alter CredentialStore Add User gguser@euro
               Password oracle_4U Alias ggueuro
Credential store in ./dircrd/ altered.

GGSCI (AMER)> Info CredentialStore
Reading from ./dircrd/:
Domain: OracleGoldenGate
  Alias: gguamer
  Userid: gguser@amer

  Alias: ggueuro
  Userid: gguser@euro

GGSCI (AMER)>
```

The Wallet will be used for parameter files and also for DBLogin.

If you make a mistake entering wallet users, the command to remove a user is:

```
GGSCI> Alter CredentialStore Delete User username
```

3. On the fifth tab, oracle@hostname, at an OS prompt, copy the source wallet single sign-on files to the target system. Execute the two copies:

```
[OS_prompt ~] cp $GG_AMER_HOME/dircrd/* $GG_EURO_HOME/dircrd
[OS_prompt ~] cp $GG_AMER_HOME/dirwlt/* $GG_EURO_HOME/dirwlt
[OS_prompt ~] ls $GG_EURO_HOME/*/*.sso
/u01/app/oracle/product/gg_euro/dircrd/cwallet.sso
/u01/app/oracle/product/gg_euro/dirwlt/cwallet.sso
```

4. On the AMER_WEST_GGSCI tab, configure the change data capture extract parameter file from the GGSCI prompt by executing the Edit command:

```
GGSCI (AMER)> Edit Param extwest
```

Your chosen text editor opens the empty extwest.prm file. Add the following runtime parameters to the text file:

```
Extract extwest
ExtTrail ./dirdat/ew
UserIDAlias gguamer
Table WEST.*;
```

There are many other optional parameters that you could add to the Extract (some are very useful). Other "best practice" parameter additions are in Practice 10-3 "Modifying the Extract Parameters on the Source Database." Shown above is only the minimal required parameters. Save and close the file.

5. On the AMER_WEST_GGSCI tab, add the Extract group by executing the following command:

```
GGSCI (AMER)> Add Extract extwest, TranLog, Begin Now
```

This adds the extwest Extract to Oracle GoldenGate, specifying that it will read from Oracle Redo/Archive log, with an initial Redo checkpoint of the current timestamp (now).

6. On the AMER_WEST_GGSCI tab, add a local extract trail that links the trail to the extwest Extract:

```
GGSCI (AMER)> Add ExtTrail ./dirdat/ew, Extract extwest
```

This adds the dirdat/ew local extract trail, and links the trail to the extwest Extract group. When the Extract is started, the dirdat/ew000000 file is created. When that file is full, trail ew000001 is created, and so on through ew999999. Your results should be similar to the following:

```
GGSCI (AMER)> Edit Param extwest

GGSCI (AMER)> Add Extract extwest, TranLog, Begin Now
EXTRACT added.

GGSCI (AMER)> Add ExtTrail ./dirdat/ew, Extract extwest
EXTTRAIL added.

GGSCI (AMER)>
```

You can look in the `dirdat` directory to prove to yourself that nothing has yet been created.

This completes Practice 6-1. Continue now with Practice 6-2.

Practice 6-2: Setting Up an Extract Data Pump and Starting the Two Extracts

Overview

In this practice, you perform the following actions:

- Configure 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.

Tasks

1. On the AMER_WEST_GGSCI tab, at the GGSCI prompt on the source server, configure the Extract data pump to read from the `dirdat/ew` local extract trail, transmit the data to the target server (EURO), and write it to the `dirdat/pe` remote trail:

   ```
   GGSCI> Edit Param pwest
   ```

2. Your text editor of choice opens the empty `pwest.prm` file. Add the following runtime parameters to the text file:

   ```
   Extract pwest
   RmtHost easthost, MgrPort 15001
   RmtTrail ./dirdat/pe
   Passthru
   Table WEST.*;
   ```

   Make sure you include the one trailing semicolon. Save and close the file.

3. Add the Extract group by executing the following GGSCI commands:

   ```
   GGSCI (AMER)> Add Extract pwest, ExtTrailSource ./dirdat/ew
   EXTRACT added.

   GGSCI (AMER)> Add RmtTrail ./dirdat/pe, Extract pwest
   RMTTRAIL added.

   GGSCI (AMER)>
   ```

4. Start the primary Extract process, `extwest`, by issuing the following command:

   ```
   GGSCI (AMER)> Start Extract extwest
   ```

5. Start the data pump Extract process, `pwest`, by issuing the following command:

   ```
   GGSCI (AMER)> Start Extract pwest
   ```

6. To verify the status of all Extracts and Replicats (ER), execute the command:

   ```
   GGSCI (AMER)> Info ER *
   ```

   Your results should look similar to the following:

```
GGSCI (AMER)> Start Extract extwest

Sending START request to MANAGER ...
EXTRACT EXTWEST starting

GGSCI (AMER)> Start Extract pwest

Sending START request to MANAGER ...
EXTRACT PWEST starting

GGSCI (AMER)> Info ER *

EXTRACT    EXTWEST   Last Started 2013-12-02 14:01   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:04 ago)
Process ID           24884
Log Read Checkpoint  Oracle Redo Logs
                     2013-12-02 14:01:54  Seqno 231, RBA 11571712
                     SCN 0.5867261 (5867261)

EXTRACT    PWEST     Last Started 2013-12-02 14:01   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:02 ago)
Process ID           24885
Log Read Checkpoint  File ./dirdat/ew000000
                     First Record  RBA 1417

GGSCI (AMER)>
```

Both processes should display Status as "RUNNING." If you see either "STOPPED" or "ABENDED," view the report log (`View Report <extract_name>`), fix the error, and restart the process. Use `Info All` for a one-line summary of process status without the details.

7. Enter the `Info` command again and see which, if any, timestamps and numbers are increasing. Some of them should be. Other database activity unrelated to this lab may also cause the numbers to increase.

   ```
   GGSCI (AMER)> Info ER *
   ```

8. Look in the source `$GG_AMER_HOME/dirdat` directory and see if any local trails are started. They should be. You can "look" using the GUI Desktop Linux bar: Applications > System Tools > File Browser; or you can use the following command:

   ```
   GGSCI (AMER)> sh ls dirdat
   ```

   You examine the content of these trails later in Practice 8 using `logdump`.

9. Look in the target `$GG_EURO_HOME/dirdat` directory and see if any remote trails are started. They should be. You can "look" using the Desktop File Browser or by entering:

   ```
   GGSCI (EURO)> sh ls dirdat
   ```

10. Using the Desktop File Browser, look in the source `$GG_AMER_HOME/dirrpt` directory and see if any reports are started. They should be. Note that the reports (`*.rpt`) are all UPPERCASE file names even though the parameter (`*.prm`) files are lowercase file names. Look at the content of an Extract report. Note that it includes the content of the associated parameter file at the time that the process was started.

This completes Practice 6-2. This completes the practices for Lesson 6. Stop here.

Practices for Lesson 7: Configuring Change Delivery

Chapter 7

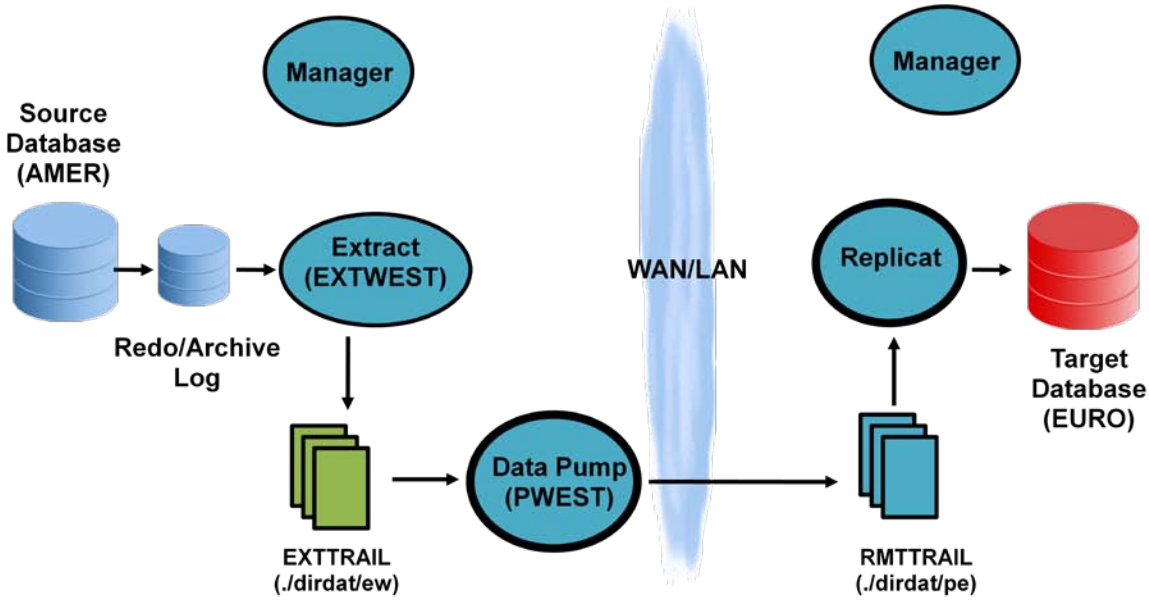
Practices for Lesson 7: Overview

Practices Overview

In these practices, you will configure change delivery. The components of change delivery include:

- Setting up the checkpoint table on the target system
- Creating a named group that includes the Replicat process and the checkpoint tables
- Configuring the Replicat group by adding parameters
- Starting the Replicat group
- Generating database activity and verifying the results

Big Picture



Practice 7-1: Setting Up the Checkpoint Table on the Target System

Overview

In this practice, you will create a checkpoint table on the target system.

Assumptions

It is assumed that you still have the five terminal tabs labeled and open from the previous labs. Replicat checkpoints to this table in the database ensure that the Replicat checkpoint is part of the transaction, and is used to ensure data integrity. For this step, you will create a special table on the target server.

Tasks

1. GGSCI should still be running on the EURO_EAST_GGSCI tab from the previous labs. If so, skip to the next task 2. If not, do the following task:

```
[OS_prompt ~] cd $GG_EURO_HOME
[OS_prompt gg_euro] . oraenv
ORACLE_SID = [euro] ? euro
[OS_prompt gg_euro] rlggsci
GGSCI (EURO) 1>
```

2. To create the checkpoint table, on the target server (on the EURO_EAST_GGSCI tab), from the installation directory, execute the following commands:

```
GGSCI (EURO)> DBLogin UserIDAlias ggueuro
Successfully logged into database.

GGSCI (EURO)> Add CheckpointTable
No checkpoint table specified. Using GLOBALS specification (GGS_CHECKPOINT)...
Successfully created checkpoint table GGS_CHECKPOINT.

GGSCI (EURO)> Info CheckpointTable
No checkpoint table specified. Using GLOBALS specification (GGS_CHECKPOINT)...
Checkpoint table GGS_CHECKPOINT created 2013-12-02 14:21:06.

GGSCI (EURO)> List Tables gg*
GGUSER.GGS_CHECKPOINT
GGUSER.GGS_CHECKPOINT_LOX
Found 2 tables matching list criteria.

GGSCI (EURO)>
```

There is no harm in doing a `DBLogin` twice. There is no `DBLogout`; you stay logged in until you `Exit` the session or `DBLogin` somewhere else. If you want to check to see if you are DB logged in anywhere, issue the `versions` command to look for the database. Unfortunately, it does not tell you *which* database you are logged in to. `Versions` will fail if you are not logged in to any database.

If you accidentally try to add a checkpoint table to the same schema twice, GGSCI will warn you, but no harm is done.

This completes Practice 7-1. Continue now with Practice 7-2.

Practice 7-2: Setting Up Replicat Delivery

Overview

In this practice, you will set up the Replicat delivery component.

Tasks

1. On the target server EURO_EAST_GGSCI tab, make sure that the Wallet containing the user IDs and passwords came over intact from the source host during the previous lab.

```
GGSCI (EURO)> Info CredentialStore
Reading from ./dircrd/:
Domain: OracleGoldenGate

  Alias: gguamer
  Userid: gguser@amer

  Alias: ggueuro
  Userid: gguser@euro

GGSCI (EURO)>
```

2. On the target EURO_EAST_GGSCI tab, execute the following command:

```
GGSCI (EURO)> Edit Param reast
```

   Your text editor of choice opens the empty `reast.prm` file. (Remember, if you prefer some editor other than the UNIX default of `vi`, you can issue the `Set Editor gedit` command for this session.) Add the following runtime parameters to the text file:

```
Replicat reast
AssumeTargetDefs
DiscardFile ./dirrpt/reast.dsc, Purge
--HandleCollisions
--End Runtime
UserIDAlias ggueuro
Map WEST.*, Target EAST.*;
```

   The double-dashed (commented-out) lines are optional. Save and close the file.

3. On the target EURO_EAST_GGSCI tab, add the Replicat by executing the following command:

```
GGSCI (EURO)> Add Replicat reast, ExtTrail ./dirdat/pe
REPLICAT added.

GGSCI (EURO)>
```

4. Start the Replicat process, and then verify the results:

```
GGSCI (EURO)> Start Replicat reast
Sending START request to MANAGER ...
REPLICAT REAST starting

GGSCI (EURO)> Info Replicat reast

REPLICAT   REAST     Last Started 2013-12-02 14:59   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:04 ago)
Process ID           25935
Log Read Checkpoint  File ./dirdat/pe000000
                     First Record  RBA 0

GGSCI (EURO)>
```

   Your details (dates, times, and PIDs) will be different. If the Replicat is not `Status RUNNING`, then examine the reports.

This completes Practice 7-2. Continue now with Practice 7-3.

Practice 7-3: Generating Data and Starting GoldenGate Processes

Overview

In this practice, you will start the data generation process and begin real-time data capture.

Assumptions

For the accounts created when you created the database, random transactions are executed: point-of-sale debit card transactions, ATM deposits, ATM withdrawals, deposits with branch tellers, and withdrawals at branch tellers. Another script will be used to simulate a real-time banking system.

Tasks

1. On the AMER_WEST_GGSCI tab, view the status of the `extwest` Extract and the `pwest` Data Pump Extract:

```
GGSCI (AMER)> Info ER *

EXTRACT    EXTWEST   Last Started 2013-12-02 14:01   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:01 ago)
Process ID           24884
Log Read Checkpoint  Oracle Redo Logs
                     2013-12-03 07:05:07  Seqno 234, RBA 26998272
                     SCN 0.5911278 (5911278)

EXTRACT    PWEST     Last Started 2013-12-02 14:01   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:04 ago)
Process ID           24885
Log Read Checkpoint  File ./dirdat/ew000000
                     First Record  RBA 1417

GGSCI (AMER)>
```

   Notice the RBA numbers for both of the Extracts. Write them here:

   _____

2. On the AMER_WEST_SQLPLUS tab, if SQL*Plus is running, exit and return to the OS prompt.

```
SQL> exit
[OS_prompt ~]
```

   Then from the `~/labs/Section7/sqlscripts` directory, run the transaction generator by executing the following commands:

```
[OS_prompt labs] ls Section7/sqlscripts/
trans_branch.sql
[OS_prompt labs] rlsqlplus west/oracle_4U
  @Section7/sqlscripts/trans_branch.sql

SQL*Plus: Release 12.1.0.1.0 Production on Tue Dec 3 07:15:19 2013
Copyright (c) 1982, 2013, Oracle.  All rights reserved.
Last Successful login time: Wed Nov 20 2013 13:46:47 +00:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
With Partitioning, OLAP, Advanced Analytics and Real Application Testing
options

'THEBEFORECOUNTFORWEST.BRANCHIS'||COUNT(*)
--------------------------------------------------------------------------------
The BEFORE count for west.branch is 40

1 row created.

...many lines omitted for clarity...
SQL>
```

Note the BEFORE count for the BRANCH table.

3. Return to the AMER_WEST_GGSCI tab and verify the results.
   Note: The RBA numbers should be increasing for the Extract, but the pump should stabilize.

```
GGSCI (AMER)> Info ER *

EXTRACT    EXTWEST   Last Started 2013-12-02 14:01   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:01 ago)
Process ID           24884
Log Read Checkpoint  Oracle Redo Logs
                     2013-12-03 07:05:07  Seqno 234, RBA 26998272
                     SCN 0.5911278 (5911278)

EXTRACT    PWEST     Last Started 2013-12-02 14:01   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:04 ago)
Process ID           24885
Log Read Checkpoint  File ./dirdat/ew000000
                     First Record  RBA 1417

GGSCI (AMER)> !

EXTRACT    EXTWEST   Last Started 2013-12-02 14:01   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:04 ago)
Process ID           24884
Log Read Checkpoint  Oracle Redo Logs
                     2013-12-03 07:21:49  Seqno 235, RBA 4100096
                     SCN 0.5912039 (5912039)

EXTRACT    PWEST     Last Started 2013-12-02 14:01   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:07 ago)
Process ID           24885
Log Read Checkpoint  File ./dirdat/ew000000
                     2013-12-03 07:15:19.000000  RBA 2299

GGSCI (AMER)>
```

4. On the EURO_EAST_GGSCI tab, view the Replicat statistics by executing the following commands:

```
GGSCI (EURO)> Info Replicat reast

REPLICAT   REAST     Last Started 2013-12-02 14:59   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:07 ago)
Process ID           25935
Log Read Checkpoint  File ./dirdat/pe000000
                     2013-12-03 07:15:19.000118  RBA 2360

GGSCI (EURO)> Info ER *

REPLICAT   REAST     Last Started 2013-12-02 14:59   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:00 ago)
Process ID           25935
Log Read Checkpoint  File ./dirdat/pe000000
                     2013-12-03 07:15:19.000118  RBA 2360

GGSCI (EURO)>
```

Notice that the RBA in EURO is now pretty close to the RBA in AMER.

5. On the AMER_WEST_GGSCI tab, issue the Info command again. When the RBA value reported for Log Read Checkpoint calms down (it will never completely stop incrementing), Extract has captured all outstanding Redo records. (You can see it "stop" within the 10 second info refresh polling window. On a production system, DML to tables other than the ones you are interested in will cause the RBA number to continue to increment.)

```
GGSCI (AMER)> Info extwest

EXTRACT    EXTWEST   Last Started 2013-12-02 14:01   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:07 ago)
Process ID           24884
Log Read Checkpoint  Oracle Redo Logs
                     2013-12-03 07:30:08  Seqno 235, RBA 4248064
                     SCN 0.5912346 (5912346)

EXTRACT    PWEST     Last Started 2013-12-02 14:01   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:07 ago)
Process ID           24885
Log Read Checkpoint  File ./dirdat/ew000000
                     2013-12-03 07:15:19.000000  RBA 2299

GGSCI (AMER)> !
Info extwest

EXTRACT    EXTWEST   Last Started 2013-12-02 14:01   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:01 ago)
Process ID           24884
Log Read Checkpoint  Oracle Redo Logs
                     2013-12-03 07:30:08  Seqno 235, RBA 4248064
                     SCN 0.5912346 (5912346)

EXTRACT    PWEST     Last Started 2013-12-02 14:01   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:02 ago)
Process ID           24885
Log Read Checkpoint  File ./dirdat/ew000000
                     2013-12-03 07:15:19.000000  RBA 2299

GGSCI (AMER)>
```

If there is any Oracle Redo activity, especially on other tables besides the ones Extract is monitoring, the RBA value will continue to change, because Extract must read Redo even if it has no pertinent data to capture.

Execute the following GGSCI command:

```
GGSCI (AMER)> Lag extwest

Sending GETLAG request to EXTRACT EXTWEST ...
Last record lag: 2 seconds.
At EOF, no more records to process.

GGSCI (AMER)>
```

When Extract returns "`At EOF, no more records to process`," then Extract has captured all outstanding redo records.

6. On the `AMER_WEST_SQLPLUS` tab, execute the following command to get the row counts for the source `BRANCH` table:

```
SQL> set sqlprompt AMER_SQL>
AMER_SQL> SELECT count(*) FROM west.branch;

  COUNT(*)
----------
        43

AMER_SQL>
```

It is not critical that your numbers exactly match the numbers above, but it is critical that your number be three more than when you started.

7. On the `EURO_EAST_SQLPLUS` tab, start a SQL*Plus session for the `east` schema by executing the following command (remember that `rlsqlplus` gives you edit history):

```
[OS_prompt labs] rlsqlplus east@euro/oracle_4U
```

If `rlsqlplus` is already running, then simply connect to the new user:

```
EURO_SQL> conn east@euro/oracle_4U
```

Get the row counts for the `BRANCH` table:

```
SQL> set sqlprompt EURO_SQL>
EURO_SQL> SELECT count(*) FROM east.branch;

  COUNT(*)
----------
         3

EURO_SQL>
```

Why do the counts not match? When you started, the source table had rows and the target was empty. If both tables were empty, this would not be a problem. If the target table was pre-populated outside of Oracle GoldenGate, this would not be a problem. If you were only interested in records from this point-in-time on, this would not be a problem. However, if you want all rows to be initially brought over by Oracle GoldenGate so the counts would match, then you would need to run an initial load before this lab. For the moment, assume that this count-mismatch is not a problem and you will get to the initial load in a later practice.

This completes Practice 7-3. Continue now with Practice 7-4.

Practice 7-4: Stopping Processes and Checking Statistics

Overview

In this practice, you will see the reports that are generated midstream and at the completion of a replication.

Assumptions

You have successfully completed the previous practices in this lab.

Tasks

1. On the fifth tab (`oracle@hostname`), use the text browser of your choice, such as `more`, `less`, or `gedit`, to look at the three reports to see if there are any statistics about rows processed:

```
[OS_prompt] more $GG_AMER_HOME/dirrpt/EXTWEST.rpt
[OS_prompt] more $GG_AMER_HOME/dirrpt/PWEST.rpt
[OS_prompt] more $GG_EURO_HOME/dirrpt/REAST.rpt
```

   The reports are interesting (barely) but contain no statistics. This is because the processes are still running.

2. On the source `AMER_WEST_GGSCI` tab, check for statistics:

```
GGSCI (AMER)> Stats extwest

Sending STATS request to EXTRACT EXTWEST ...
Start of Statistics at 2013-12-03 07:55:47.
Output to ./dirdat/ew:
Extracting from WEST.BRANCH to WEST.BRANCH:

...many lines omitted for clarity...

*** Latest statistics since 2013-12-03 07:15:22 ***
     Total inserts                                 4.00
     Total updates                                 2.00
     Total deletes                                 1.00
     Total discards                                0.00
     Total operations                              7.00
End of Statistics.

GGSCI (AMER)>
```

3. On the target EURO_EAST_GGSCI tab, check for statistics:

```
GGSCI (EURO)> Stats reast

Sending STATS request to REPLICAT REAST ...
Start of Statistics at 2013-12-03 07:58:36.
Replicating from WEST.BRANCH to EAST.BRANCH:

...many lines omitted for clarity...

*** Latest statistics since 2013-12-03 07:15:33 ***
     Total inserts                                 4.00
     Total updates                                 2.00
     Total deletes                                 1.00
     Total discards                                0.00
     Total operations                              7.00
End of Statistics.

GGSCI (EURO)>
```

These statistics should match the source and the target.

4. On the EURO_EAST_GGSCI tab, stop the target processes:

```
GGSCI (EURO)> Stop *
```

5. On the AMER_WEST_GGSCI tab, stop the source processes:

```
GGSCI (AMER)> Stop *
```

Note: Stopping with an asterisk does not stop the Manager.

6. On the fifth tab (oracle@hostname), look at the three reports again to see if there are now any statistics about rows processed:

```
[OS_prompt] more $GG_AMER_HOME/dirrpt/EXTWEST.rpt
[OS_prompt] more $GG_AMER_HOME/dirrpt/PWEST.rpt
[OS_prompt] more $GG_EURO_HOME/dirrpt/REAST.rpt
```

Instead of more, you can use any text editor or less. The reports now contain plenty of statistics. Find the section that has similar statistics as the Stats command gave:

```
***********************************************************************
                 Oracle GoldenGate Capture for Oracle
 Version 12.1.2.0.0 17185003 OGGCORE_12.1.2.0.0_PLATFORMS_130924.1316_FBO
   Linux, x64, 64bit (optimized), Oracle 12c on Sep 25 2013 02:47:30
Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.
                  Starting at 2013-12-02 14:01:16
***********************************************************************
...many lines omitted for clarity...
***********************************************************************
*                  ** Run Time Statistics **                          *
***********************************************************************
Report at 2013-12-03 08:01:23 (activity since 2013-12-03 07:15:22)
Output to ./dirdat/ew:
From Table WEST.BRANCH:
       #                  inserts:         4
       #                  updates:         2
       #                  deletes:         1
       #                 discards:         0
...many lines omitted for clarity...
```

Each of the three reports contains slightly different information, but all three should contain the above common statistics.

This completes Practice 7-4. This completes the practices for Lesson 7. Stop here.

Practices for Lesson 8: Extract Trail and Files

Chapter 8

Practices for Lesson 8: Overview

Practices Overview

In this practice, you will be able to use the `logdump` utility to look at records in an Oracle GoldenGate trail file.

In this practice, 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
- Use the `logdump` help feature

Practice 8-1: Using the logdump Utility

Tasks

1. On the AMER_WEST_GGSCI tab, exit to the OS prompt, navigate to the source installation directory $GG_AMER_HOME, and invoke *logdump*:

```
GGSCI (AMER)> Exit
[OS_prompt gg_amer] pwd
/u01/app/oracle/product/gg_amer
[OS_prompt gg_amer] ./logdump

Oracle GoldenGate Log File Dump Utility for Oracle
Version 12.1.2.0.0 17185003 OGGCORE_12.1.2.0.0_PLATFORMS_130924.1316
Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.

Logdump 1 > help
```

   Issue the help command to display a list of all commands available in *logdump*. Similar to GGSCI, there is no good way to make help or any other command pause.

2. In *logdump*, open the <*install_path*>/dirdat/<*trail_id*>000000 trail file: (Verify the name of the trail file in the dirdat directory.)

```
Logdump 2 > open dirdat/ew000000
Current LogTrail is
/u01/app/oracle/product/gg_amer/dirdat/ew000000
```

Identifying the File Headers

3. View the first record. Use the n (next) command:

```
Logdump 3 > n

2013/12/02 14:01:18.018.941 FileHeader           Len  1409 RBA 0
Name: *FileHeader*
 3000 030c 3000 0008 4747 0d0a 544c 0a0d 3100 0002 | 0...0...GG..TL..1...
 0004 3200 0004 2000 0000 3300 0008 02f2 12c1 11e4 | ..2... ...3.........
 a17d 3400 0037 0035 7572 693a 4544 5253 5232 3650 | .}4..7.5uri:EDRSR26P
 313a 3a75 3031 3a61 7070 3a6f 7261 636c 653a 7072 | 1::u01:app:oracle:pr
 6f64 7563 743a 6767 5f61 6d65 723a 4558 5457 4553 | oduct:gg_amer:EXTWES
 5436 0000 1300 112e 2f64 6972 6461 742f 6577 3030 | T6....../dirdat/ew00
 3030 3030 3700 0001 0138 0000 0400 0000 0039 ff00 | 00007....8.......9..

Logdump 4 >
```

   Note that this is the trail file header record.

4. Turn on the FILEHEADER ON option and view the record again:

```
Logdump 4 > fileheader on
Logdump 5 > pos 0
Reading forward from RBA 0
Logdump 6 > n
```

   Scroll down and notice how you can view the detail on the trail, the machine being used, the Extract that produced the trail, the database information, and so on.

```
... Many lines omitted for clarity ...

GroupID x31 '1' MachineInfo      Info x00  Length  114
 3100 0072 3000 0007 0005 4c69 6e75 7831 0000 0b00 | 1..r0.....Linux1....
 0945 4452 5352 3236 5031 3200 001f 001d 322e 362e | .EDRSR26P12.....2.6.
 3339 2d32 3030 2e32 342e 312e 656c 3675 656b 2e78 | 39-200.24.1.el6uek.x
 3836 5f36 3433 0000 2500 2323 3120 534d 5020 5361 | 86_643..%.##1 SMP Sa
 7420 4a75 6e20 3233 2030 323a 3339 3a30 3720 4544 | t Jun 23 02:39:07 ED
 5420 3230 3132 3400 0008 0006 7838                | T 20124.....x8

GroupID x32 '2' DatabaseInfo     Info x00  Length  360
 3200 0168 3000 0002 0007 3100 0006 0004 414d 4552 | 2..h0.....1.....AMER
 3200 0006 0004 616d 6572 3300 0004 0000 0000 3400 | 2.....amer3.......4.
 0002 000c 3500 0002 0001 3600 00e7 00e5 4f72 6163 | ....5.....6.....Orac
 6c65 2044 6174 6162 6173 6520 3132 6320 456e 7465 | le Database 12c Ente
 7270 7269 7365 2045 6469 7469 6f6e 2052 656c 6561 | rprise Edition Relea
 7365 2031 322e 312e 302e 312e 3020 2d20 3634 6269 | se 12.1.0.1.0 - 64bi
 7420 5072 6f64 7563 7469 6f6e 0a50 4c2f 5351 4c20 | t Production.PL/SQL
 5265 6c65 6173 6520 3132 2e31 2e30 2e31 2e30 202d | Release 12.1.0.1.0 -
 2050 726f 6475 6374 696f 6e0a 434f 5245 0931 322e |  Production.CORE.12.
 312e 302e 312e 3009 5072 6f64 7563 7469 6f6e 0a54 | 1.0.1.0.Production.T
 4e53 2066 6f72 204c 696e 7578 3a20 5665 7273 696f | NS for Linux: Versio
 6e20 3132 2e31 2e30 2e31 2e30 202d 2050 726f 6475 | n 12.1.0.1.0 - Produ
 6374 696f 6e0a 4e4c 5352 544c 2056 6572 7369 6f6e | ction.NLSRTL Version
 2031 322e 312e 302e 312e 3020 2d20 5072 6f64 7563 |  12.1.0.1.0 - Produc
 7469 6f6e 0a37 0000 0440 0000 0038 0000 0c00 0a31 | tion.7...@...8.....1
 322e 312e 302e 312e 3039 0000 0400 0000 013a 0000 | 2.1.0.1.09.......:..
 0200 003b 0000 0400 0000 013c 0000 1400 0000 1014 | ...;.......<........
 1414 1414 1414 1414 1414 1411 1414 143d 0000 0500 | ..............=....

... Many lines omitted for clarity ...
```

Using the Head and Detail Attributes when Displaying a Record

5. View the first next record:

```
Logdump 7 > n

2013/12/03 07:15:19.000.000 Insert               Len    28 RBA 1417
Name: WEST.BRANCH
After  Image:                                             Partition 4   G  b
 0000 000a 0000 0000 0000 0000 0064 0001 000a 0000 | .............d......
 0000 0000 0000 292f                               | ......)/

Logdump 8 >
```

This is a standard trail record. A database type of NUMBER is not as easily readable as a CHAR or VARCHAR. If you are curious, this record is the line

```
INSERT INTO west.branch VALUES (100, 10543);
```

from the previous lab's `trans_branch.sql`, the "`0064`" in hex is the `branch_number` 100 in decimal, and "`292f`" in hex above equals 10543 in decimal (the new `branch_zip`).

6. Position back to **RBA 0** in the file, set the header and detail attributes to on, and view the same trail record again. (Make sure you do `next` or n twice.)

```
Logdump 8 > pos 0
Reading forward from RBA 0
Logdump 9 > ghdr on
Logdump 10 > detail on
Logdump 11 > n
Logdump 12 > n


___________________________________________________________________

Hdr-Ind    :     E  (x45)     Partition  :     .  (x04)
UndoFlag   :     .  (x00)     BeforeAfter:     A  (x41)
RecLength  :    28  (x001c)   IO Time    : 2013/12/03 07:15:19.000.000
IOType     :     5  (x05)     OrigNode   :   255  (xff)
TransInd   :     .  (x00)     FormatType :     R  (x52)
SyskeyLen  :     0  (x00)     Incomplete :     .  (x00)
AuditRBA   :        235       AuditPos   : 4057616
Continued  :     N  (x00)     RecCount   :     1  (x01)

2013/12/03 07:15:19.000.000 Insert               Len    28 RBA 1417
Name: WEST.BRANCH
After  Image:                                             Partition 4   G  b
 0000 000a 0000 0000 0000 0000 0064 0001 000a 0000 | .............d......
 0000 0000 0000 292f                               | ......)/
Column     0 (x0000), Len    10 (x000a)
Column     1 (x0001), Len    10 (x000a)

Logdump >
```

Note the difference in the display from the output in the last step. (*The trail record header area and detail on the columns have been added to the display.)*

Exit `logdump`.

7. *Restart* `logdump`. *Notice that the* `history` *and the last prompt number is preserved in* `~/logdump.hst`, *but the session information (for example, which trail file) is not preserved. That is, you can display the commands you ran last session to open a trail file, but the file is not currently opened. Prove this by entering* `next`. *Exit* `logdump` *again.*

8. For more information about the `logdump` utility, which reference manual would you use?

_____ (**Hint:** Look in `~/Documents`).

This completes Practice 8-1. This completes the practices for Lesson 8. Stop here.

Practices for Lesson 9: Configuring Initial Load

Chapter 9

Practices for Lesson 9: Overview

Practices Overview

In these practices, you will set up the initial load of data by using two different methods:

- File to Replicat
- Direct Load

Practice 9-1: Setting Up the Initial Load by Using the File to Replicat Method

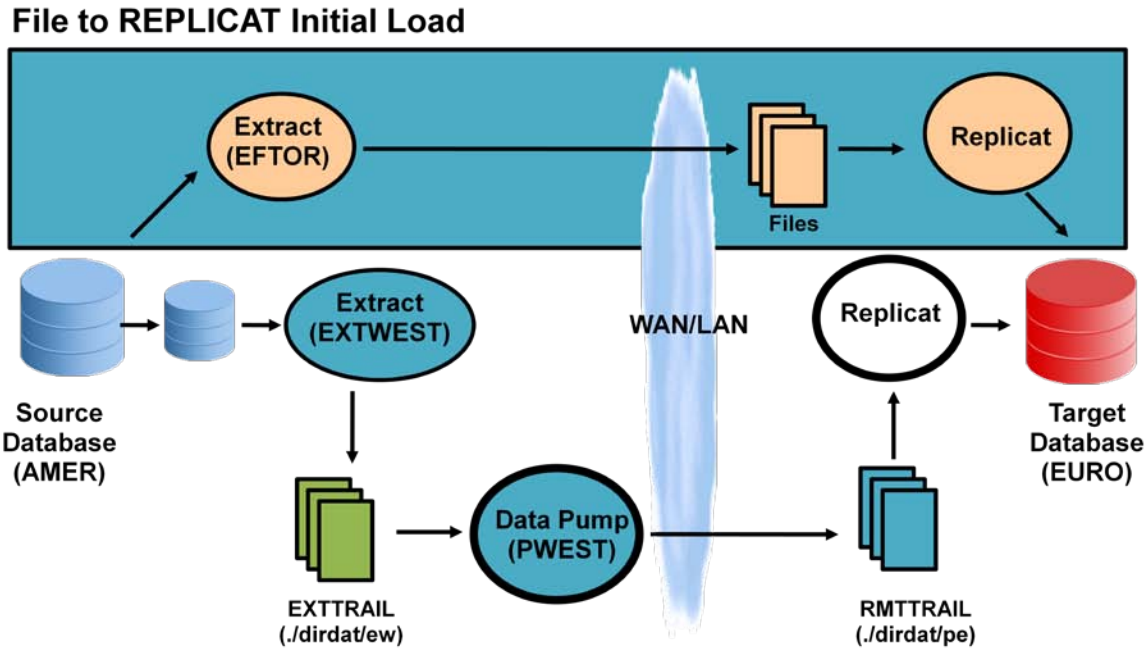
Overview

In this practice, you will:

- Configure a task to load the initial data from two source tables: `ACCOUNT` and `BRANCH`
- Configure the delivery of the data to the target
- Execute the initial load of table data

This File to Replicat 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.

Big Picture



The Extract naming convention is `eftor` (Extract File To Replicat).

Tasks

1. GGSCI for the source was stopped in the previous `logdump` lab. Start GGSCI on the `AMER_WEST_GGSCI` tab. Configure the initial load Extract parameter file by executing the following commands on the source system to create an Extract named `eftor`:

```
[OS_prompt gg_amer] rlggsci
GGSCI (AMER) 1> DBLogin UserIDAlias gguamer
GGSCI (AMER) 2> Set Editor gedit
GGSCI (AMER) 3> Edit Param eftor
```

The editor of your choice opens the `eftor.prm` file. Add the following runtime parameters to the text file:

```
SourceIsTable
UserIDAlias gguamer
RmtHost easthost, MgrPort 15001
RmtFile ./dirdat/ACCOUNT.DAT, Purge
Table WEST.ACCOUNT;
RmtFile ./dirdat/BRANCH.DAT, Purge
Table WEST.BRANCH;
```

Save and close the file.

2. On the fifth tab, `oracle@hostname`, execute the initial load capture process by starting the initial load Extract directly at a shell prompt from the source installation directory by using the following command (all on one line, and all lowercase):

```
[OS_prompt] cd $GG_AMER_HOME
[OS_prompt gg_amer] ./extract paramfile dirprm/eftor.prm
   reportfile dirrpt/eftor.rpt
[OS_prompt gg_amer]
```

Notes

- There is no message returned to the command line, neither good nor bad. That information is in the source report.
- If for any reason you need to rerun the extract (for example, you made a mistake), a new report will not overlay the old one. You must first delete the old report, and then rerun the extract.

3. To view the report, navigate to the AMER `dirrpt` directory and execute the following commands:

```
[OS_prompt gg_amer] more dirrpt/eftor.rpt

2013-12-03 11:59:43  INFO    OGG-02095  Successfully set environment variable
ORACLE_SID=amer.

**********************************************************************
                 Oracle GoldenGate Capture for Oracle
 Version 12.1.2.0.0 17185003 OGGCORE_12.1.2.0.0_PLATFORMS_130924.1316_FBO
   Linux, x64, 64bit (optimized), Oracle 12c on Sep 25 2013 02:47:30
Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.
                  Starting at 2013-12-03 11:59:43
**********************************************************************

Operating System Version: Linux
Version #1 SMP Sat Jun 23 02:39:07 EDT 2012, Release 2.6.39-
200.24.1.el6uek.x86_64
Node: HOSTNAME
Machine: x86_64
--More--(37%)
```

4. Scroll to the end of the report to view the number of inserts for the ACCOUNT and BRANCH tables: (Your results may vary slightly.)

```
***********************************************************************
*                   ** Run Time Statistics **                         *
***********************************************************************

Report at 2013-12-03 11:59:51 (activity since 2013-12-03 11:59:45)
Output to ./dirdat/ACCOUNT.DAT:

From Table WEST.ACCOUNT:
       #                   inserts:      1000
       #                   updates:         0
       #                   deletes:         0
       #                  discards:         0

Output to ./dirdat/BRANCH.DAT:

From Table WEST.BRANCH:
       #                   inserts:        43
       #                   updates:         0
       #                   deletes:         0
       #                  discards:         0

REDO Log Statistics
  Bytes parsed                   0
  Bytes output               95618

[OS_prompt gg_amer]
```

Note: This did not put the rows in the remote *table*; it put them in a remote *file*. You can see some of it by entering:

```
[OS_prompt gg_amer] strings $GG_EURO_HOME/dirdat/ACCOUNT.DAT
```

5. On the EURO_EAST_GGSCI tab, GGSCI should still be running. Configure the initial load delivery parameter file by executing the following commands on the **target** system:

   a. Configure the data load parameters for ACCOUNT by executing the following command: (The file name starts with a lowercase L as in “load account.”)

   ```
   GGSCI (EURO)> Edit Param lacct
   ```

   Your editor of choice opens the empty `lacct.prm` file. Add the following runtime parameters to the text file:

```
SpecialRun
End Runtime
UserIDAlias ggueuro
AssumeTargetDefs
ExtFile ./dirdat/ACCOUNT.DAT
Map WEST.ACCOUNT, Target EAST.ACCOUNT;
```

   Save and close the file.

   b. On the EURO_EAST_GGSCI tab, repeat this process for the BRANCH Initial Load parameter file:

   ```
   GGSCI (EURO)> Edit Param lbranch
   ```

   Your editor of choice opens the empty `lbranch.prm` file. Add the following runtime parameters to the text file:

```
SpecialRun
End Runtime
UserIDAlias ggueuro
HandleCollisions
AssumeTargetDefs
ExtFile ./dirdat/BRANCH.DAT
Map WEST.BRANCH, Target EAST.BRANCH;
```

Note: This differs from the `lacct` file in that it has `HandleCollisions` (that is, "ignore duplicates") to accommodate the three pre-existing rows from the previous lab.

Save and close the file. Leave GGSCI running.

6. On the fifth tab, `oracle@hostname`, execute the Initial Load delivery process by executing the following commands at the shell prompt from the target installation directory (each Replicat all on one line, and all lowercase):

```
[OS_prompt gg_amer] cd $GG_EURO_HOME
[OS_prompt gg_euro] ./replicat paramfile dirprm/lacct.prm
   reportfile dirrpt/lacct.rpt
[OS_prompt gg_euro] ./replicat paramfile dirprm/lbranch.prm
   reportfile dirrpt/lbranch.rpt
[OS_prompt gg_euro]
```

Notes

- There is no message returned to the command line, neither good nor bad. That information is in the target report.
- If for any reason you need to rerun the replicat (for example, you made a mistake), a new report will not overlay the old one. You must first delete the old report, and then rerun the replicat.

Wait for ten seconds and then verify the results in the target report files:

```
[OS_prompt gg_euro] more dirrpt/lacct.rpt
[OS_prompt gg_euro] more dirrpt/lbranch.rpt
```

Results from `lacct.rpt`:

```
... Many lines omitted for clarity ...
2013-12-03 11:59:51.502486 Insert             Len    28 RBA 93026
Name: WEST.ACCOUNT
_____________________________________________________________________
Reading ./dirdat/ACCOUNT.DAT, current RBA 93117, 1000 records
Report at 2013-12-03 12:23:55 (activity since 2013-12-03 12:23:55)
From Table WEST.ACCOUNT to EAST.ACCOUNT:
       #                   inserts:      1000
       #                   updates:         0
       #                   deletes:         0
       #                  discards:         0

Last log location read:
     FILE:      ./dirdat/ACCOUNT.DAT
     RBA:       93117
     TIMESTAMP: 2013-12-03 11:59:51.502486
     EOF:       NO
     READERR:   400

... Many lines omitted for clarity ...
```

The number of inserts should be 1000.

Results from `lbranch.rpt`:

```
... Many lines omitted for clarity ...
2013-12-03 11:59:51.575484 Insert             Len    28 RBA 5148
Name: WEST.BRANCH
_____________________________________________________________________
Reading ./dirdat/BRANCH.DAT, current RBA 5238, 43 records
Report at 2013-12-03 12:24:08 (activity since 2013-12-03 12:24:08)
From Table WEST.BRANCH to EAST.BRANCH:
       #                   inserts:        43
       #                   updates:         0
       #                   deletes:         0
       #                  discards:         0
       #          insert collisions:         3

Last log location read:
     FILE:      ./dirdat/BRANCH.DAT
     RBA:       5238
     TIMESTAMP: 2013-12-03 11:59:51.575484
     EOF:       NO
     READERR:   400

... Many lines omitted for clarity ...
```

The number of inserts should be 40 or 43, depending on how you think about the collisions.

This completes Practice 9-1. Continue now with Practice 9-2.

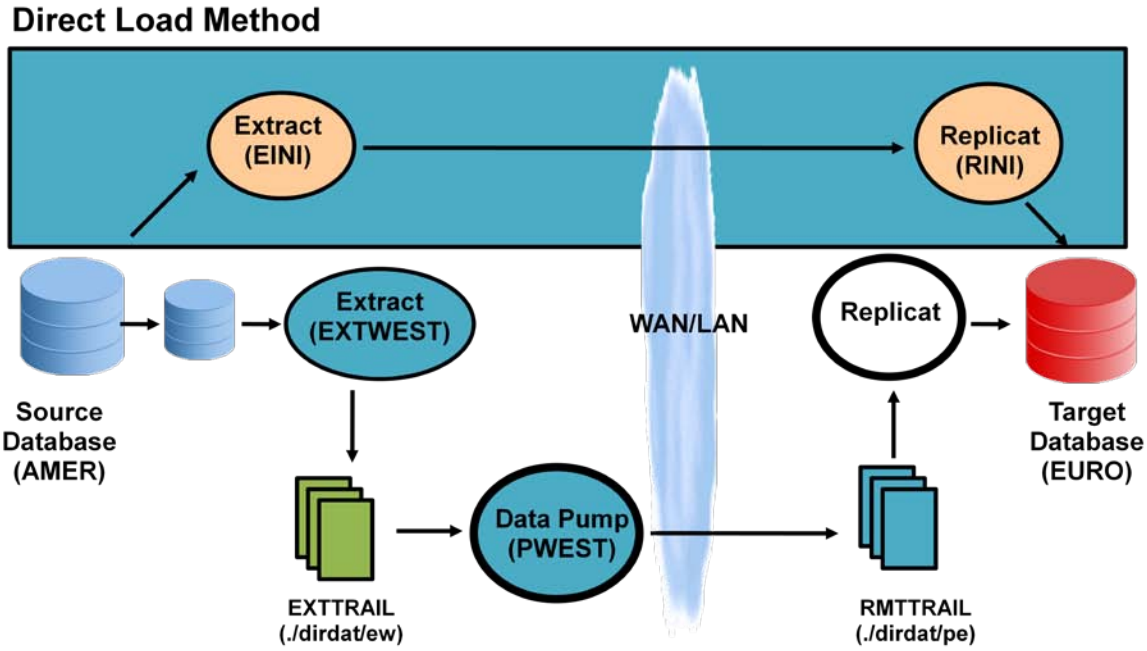
Practice 9-2: Setting Up the Initial Data Load by Using the Direct Load Method

Overview

In this practice, you will configure:

- A task to load the initial data from all of the source tables
- The delivery of the data to the target

Big Picture



Assumptions

For this practice, target database instantiation will be performed via Oracle GoldenGate tasks. An Extract Task is a special process that, once started, reads from each source table row by row. This data is transmitted to the Replicat task on the target that inserts each row of data into the empty database tables. The Task will automatically shut down once all rows are read, transmitted, and inserted.

This exercise assumes that:

- SQL*Plus is still running on the `EURO_EAST_SQLPLUS` tab. If not, start it by using **`rlsqlplus`**.
- GGSCI is still running in both the `AMER_WEST_GGSCI` and `EURO_EAST_GGSCI` tabs. If not, start it with **`rlggsci`**, and then:
  `DBLogin UserID gguser, Password oracle_4U`

Tasks

1. Truncate the target ACCOUNT and BRANCH tables.

   Note: This exercise assumes that you have completed an initial data load by using the File to Replicat method. If this is not the case, you may skip to step 2.

   Execute the following commands on the target system EURO_EAST_SQLPLUS tab:

```
SQL> conn east@euro/oracle_4U
Connected.
SQL> set sqlprompt EURO_SQL>
EURO_SQL> TRUNCATE TABLE account;
Table truncated.

EURO_SQL> TRUNCATE TABLE branch;
Table truncated.

EURO_SQL> COMMIT;
Commit complete.

EURO_SQL>
```

2. To add the initial load capture batch task group, on the source system AMER_WEST_GGSCI tab, in the installation directory, execute the following command:

```
GGSCI (AMER)> Add Extract eini, SourceIsTable
```

3. Configure the initial load capture parameter file by executing the following command on the source system AMER_WEST_GGSCI tab:

```
GGSCI (AMER)> Edit Param eini
```

   The editor of your choice opens the empty eini.prm file. Add the following runtime parameters to the text file:

```
Extract eini
RmtHost easthost, MgrPort 15001
RmtTask Replicat, Group rini
UserIDAlias gguamer
Table WEST.*;
```

   Save and close the file.

4. Add the initial load delivery batch task by executing the following command on the target system EURO_EAST_GGSCI tab:

```
GGSCI (EURO)> Add Replicat rini, SpecialRun
```

5. Configure the initial load delivery parameter file by executing the following command on the target database EURO_EAST_GGSCI tab:

```
GGSCI (EURO) > Edit Param rini
```

The editor of your choice opens the empty rini.prm file. Add the following runtime parameters to the text file:

```
Replicat rini
AssumeTargetDefs
UserIDAlias ggueuro
DiscardFile ./dirrpt/rini.dsc, Purge
Map WEST.*, Target EAST.*;
```

Save and close the file.

6. On the AMER_WEST_GGSCI tab, execute the initial load process in the source system:

```
GGSCI (AMER)> Start Extract eini

Sending START request to MANAGER ...
EXTRACT EINI starting

GGSCI (AMER)>
```

Note that Start * will not start this kind of an initial load; you need to name it explicitly.

7. This task displays differently than other Extracts. Enter the commands:

```
GGSCI (AMER)> Info All
GGSCI (AMER)> Info Extract *
GGSCI (AMER)> Info Extract eini
```

Notice that this initial load Extract does not show up in Info All, nor Info with an asterisk, but does show up in Info Extract eini. It will show with a Status of STOPPED (which means that it is already completed). If you are *very* fast, you may briefly see Status STARTING.

8. On the target EURO_EAST_GGSCI tab, view the report by executing the following command:

```
GGSCI (EURO)> View Report rini
... Many lines omitted for clarity ...
***********************************************************************
*                   ** Run Time Statistics **                         *
***********************************************************************
Report at 2013-12-03 13:48:06 (activity since 2013-12-03 13:48:00)

From Table WEST.ACCOUNT to EAST.ACCOUNT:
       #                   inserts:      1000
       #                   updates:         0
       #                   deletes:         0
       #                  discards:         0
From Table WEST.BRANCH to EAST.BRANCH:
       #                   inserts:        43
       #                   updates:         0
       #                   deletes:         0
       #                  discards:         0
From Table WEST.TELLER to EAST.TELLER:
       #                   inserts:       800
       #                   updates:         0
       #                   deletes:         0
       #                  discards:         0

... Many lines omitted for clarity ...
```

Make a note of the number of inserts from the initial load.

Note: If you entered `List Tables WEST.*` on the source AMER_WEST_GGSCI tab, you would see that there are *six* tables, but only *three* of them came over in the initial load, the three that had rows in them. The other three tables, ACCOUNT_TRANS, TELLER_TRANS, and BRANCH_ATM existed on source and target, but were empty. The truncated tables are examined by Oracle GoldenGate and then wisely ignored; they do not show up in the statistics, nor in the report.

9. On the source database AMER_WEST_SQLPLUS tab, verify that the count for the tables matches: (The numbers should match in both databases.)

```
AMER_SQL> SELECT count(*) FROM account;

  COUNT(*)
----------
      1000

AMER_SQL> SELECT count(*) FROM branch;

  COUNT(*)
----------
        43

AMER_SQL> SELECT count(*) FROM teller;

  COUNT(*)
----------
       800
```

and then on the target database EURO_EAST_SQLPLUS tab:

```
EURO_SQL> SELECT count(*) FROM account;

  COUNT(*)
----------
      1000

EURO_SQL> SELECT count(*) FROM branch;

  COUNT(*)
----------
        43

EURO_SQL> SELECT count(*) FROM teller;

  COUNT(*)
----------
       800
```

Leave SQL*Plus running.

10. Because you will not need the initial load process groups again, delete them from both Oracle GoldenGate instances. To unregister them from the databases, GGSCI needs to be logged in to the databases. There is no harm if you log in to the database more than once.

On the source AMER_WEST_GGSCI tab:

```
GGSCI (AMER)> DBLogin UserIDAlias gguamer
GGSCI (AMER)> Delete Extract eini
```

On the target EURO_EAST_GGSCI tab:

```
GGSCI (EURO)> DBLogin UserIDAlias ggueuro
GGSCI (EURO)> Delete Replicat rini
```

This completes Practice 9-2. Continue now with Practice 9-3.

Practice 9-3: Putting it All Together

Overview

In this practice, you will restart the Extract and Replicat from earlier practices, and run a transaction generator to verify that all the data flows from source tables to target tables.

Assumptions

- You successfully completed the previous practices.
- The source and target tables in the WEST and EAST schemas are now identical as a result of the successful initial load.
- GGSCI and SQL*Plus are all running on both source and targets.
- DBLogin is still in effect on both GGSCI sessions.

Tasks

1. On the source system AMER_WEST_GGSCI tab, start all the processes and make sure they are all running:

```
GGSCI (AMER)> Start ER *

Sending START request to MANAGER ...
EXTRACT EXTWEST starting

Sending START request to MANAGER ...
EXTRACT PWEST starting

GGSCI (AMER)> Info All

Program     Status      Group       Lag at Chkpt  Time Since Chkpt
MANAGER     RUNNING
EXTRACT     RUNNING     EXTWEST     00:00:00      00:00:05
EXTRACT     RUNNING     PWEST       00:01:00      00:00:02
GGSCI (AMER)>
```

2. On the target system EURO_EAST_GGSCI tab, start all the processes and make sure they are all running:

```
GGSCI (EURO)> Start ER *

Sending START request to MANAGER ...
REPLICAT REAST starting

GGSCI (EURO)> Info All

Program     Status      Group       Lag at Chkpt  Time Since Chkpt
MANAGER     RUNNING
REPLICAT    RUNNING     REAST       00:00:00      00:00:05

GGSCI (EURO)>
```

3. On the source AMER_WEST_SQLPLUS tab, from ~/labs/Section9/sqlscripts, run the transaction generator as the owner of the source schema:

```
AMER_SQL> show user
USER is "WEST"
AMER_SQL>
  @/home/oracle/labs/Section9/sqlscripts/trans_generator.sql

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
[OS_prompt sqlscripts]
```

The PL/SQL procedure takes a few seconds to complete.

4. Make sure that your Oracle GoldenGate processes are still running.
   - On the source system GGSCI, gather information:

```
GGSCI (AMER)> !
Info All

Program     Status      Group       Lag at Chkpt  Time Since Chkpt
MANAGER     RUNNING
EXTRACT     RUNNING     EXTWEST     00:00:00      00:00:01
EXTRACT     RUNNING     PWEST       00:00:00      00:00:00

GGSCI (AMER)>
```

   - On the target system GGSCI, gather information:

```
GGSCI (EURO)> !
Info All

Program     Status      Group       Lag at Chkpt  Time Since Chkpt
MANAGER     RUNNING
REPLICAT    RUNNING     REAST       00:00:00      00:00:04

GGSCI (EURO)>
```

5. Examine the row count on both the source and the target tables:
   a. On the source database AMER_WEST_SQLPLUS tab:

```
[OS_prompt labs] rlsqlplus west@amer/oracle_4U
SQL*Plus: Release 12.1.0.1.0 Production on Wed Dec 4 07:58:18 2013
SQL> set sqlprompt AMER_SQL>

AMER_SQL> SELECT count(*) FROM account;
COUNT(*)
----------
1060

AMER_SQL> SELECT count(*) FROM account_trans;
COUNT(*)
----------
1500

AMER_SQL> SELECT count(*) FROM teller_trans;
COUNT(*)
----------
559

AMER_SQL> SELECT count(*) FROM branch_atm;
COUNT(*)
----------
563

AMER_SQL>
```

   b. On the target database EURO_EAST_SQLPLUS tab:

```
[OS_prompt labs] rlsqlplus east@euro/oracle_4U
SQL*Plus: Release 12.1.0.1.0 Production on Wed Dec 4 07:58:18 2013
SQL> set sqlprompt EURO_SQL>

EURO_SQL> SELECT count(*) FROM account;
COUNT(*)
----------
1060

EURO_SQL> SELECT count(*) FROM account_trans;
COUNT(*)
----------
1500

EURO_SQL> SELECT count(*) FROM teller_trans;
COUNT(*)
----------
559

EURO_SQL> SELECT count(*) FROM branch_atm;
COUNT(*)
----------
563

EURO_SQL>
```

   Source and target row counts should match. Your numbers may be slightly different. (There is a random number generator as part of the transaction generator function.)

6. Examine the statistics. Five of the six tables had activity:
   a. On the source system, on the AMER_WEST_GGSCI tab, gather statistics:

```
GGSCI (AMER)> Stats pwest
Sending STATS request to EXTRACT PWEST ...
Start of Statistics at 2013-12-04 12:55:44.
Output to ./dirdat/pe:

Extracting from WEST.ACCOUNT_TRANS to WEST.ACCOUNT_TRANS:
*** Total statistics since 2013-12-04 12:11:20 ***
     Total inserts                          1500.00

Extracting from WEST.BRANCH_ATM to WEST.BRANCH_ATM:
*** Total statistics since 2013-12-04 12:11:20 ***
     Total inserts                           563.00

Extracting from WEST.ACCOUNT to WEST.ACCOUNT:
*** Total statistics since 2013-12-04 12:11:20 ***
     Total inserts                            60.00

Extracting from WEST.TELLER_TRANS to WEST.TELLER_TRANS:
*** Total statistics since 2013-12-04 12:11:20 ***
     Total inserts                           559.00

Extracting from WEST.BRANCH to WEST.BRANCH:
*** Total statistics since 2013-12-04 12:11:20 ***
     Total deletes                             3.00
...many lines have been omitted for clarity...

End of Statistics.

GGSCI (AMER)>
```

b. On the target system, on the EURO_EAST_GGSCI tab, gather statistics:

```
GGSCI (EURO)> Stats reast
Sending STATS request to REPLICAT REAST ...
Start of Statistics at 2013-12-04 12:59:20.

Replicating from WEST.ACCOUNT_TRANS to EAST.ACCOUNT_TRANS:
*** Total statistics since 2013-12-04 12:11:22 ***
     Total inserts                          1500.00

Replicating from WEST.BRANCH_ATM to EAST.BRANCH_ATM:
*** Total statistics since 2013-12-04 12:11:22 ***
     Total inserts                           563.00

Replicating from WEST.ACCOUNT to EAST.ACCOUNT:
*** Total statistics since 2013-12-04 12:11:22 ***
     Total inserts                            60.00

Replicating from WEST.TELLER_TRANS to EAST.TELLER_TRANS:
*** Total statistics since 2013-12-04 12:11:22 ***
     Total inserts                           559.00

Replicating from WEST.BRANCH to EAST.BRANCH:
*** Total statistics since 2013-12-04 12:11:22 ***
     Total deletes                             3.00
...many lines have been omitted for clarity...

End of Statistics.

GGSCI (EURO)>
```

Notice that data pump pwest is from WEST to WEST, but the Replicat reast is from WEST to EAST. Also notice that the Replicat is lagging 2 seconds behind the pump. Your counts and times may be slightly different; the important thing is that source WEST matches target EAST.

This completes Practice 9-3. Continue now with Practice 9-4.

Practice 9-4: Converting from Classic Mode to Integrated Mode

Overview

The process for converting the Extract and Replicat from Classic Mode to Integrated Mode is slightly different on a production database than on a test or development database due to the lack of other log traffic. On the classroom one-CPU machines, you will not be able to see anything very different.

Assumptions

- The Extract, Data Pump, and Replicat are all working from the previous lab
- No other practice depends on this conversion being completed; it is, therefore, optional.

Tasks

1. Register the Extract with the database. On the AMER_WEST_GGSCI tab, enter:

```
GGSCI (AMER)> Stop extwest
Sending STOP request to EXTRACT EXTWEST ...
Request processed.

GGSCI (AMER)> Register Extract extwest Database
Extract EXTWEST successfully registered with database at SCN 5988879.

GGSCI (AMER)> Start extwest
EXTRACT EXTWEST starting

GGSCI (AMER)>
```

   Your SCN will be different.

2. On the source database AMER_WEST_SQLPLUS tab, enter the following DML:

```
AMER_SQL> INSERT INTO west.account VALUES (7000,7000);
AMER_SQL> COMMIT;
AMER_SQL> INSERT INTO west.account VALUES (8000,8000);
AMER_SQL> COMMIT;
```

   The goal is to create some database traffic to increase the SCN value from what it was when the Extract was registered. In a busy production environment, this would happen from other traffic automatically.

3. On the target database EURO_EAST_SQLPLUS tab, enter the following query:

```
EURO_SQL> SELECT * FROM east.account
           WHERE account_number > 6000;

ACCOUNT_NUMBER ACCOUNT_BALANCE
-------------- --------------
      7000           7000
      8000           8000
```

This just makes sure that the GGSCI processes are still working.

4. You want to convert the target process before you convert the source process. On the target system EURO_EAST_GGSCI tab, enter the following commands:

```
GGSCI (EURO)> Stop reast
Sending STOP request to REPLICAT REAST ...
Request processed.

GGSCI (EURO)> Alter Replicat reast, Integrated
REPLICAT (Integrated) altered.

GGSCI (EURO)>
```

NOTE: Do *not* restart it yet!

5. When all GGSCI processes are stopped, convert the source Extract (not the data pump). On the source system AMER_WEST_GGSCI tab, enter the following commands:

```
GGSCI (AMER)> Stop extwest
Sending STOP request to EXTRACT EXTWEST ...
Request processed.

GGSCI (AMER)> Alter Extract extwest, Upgrade Integrated TranLog
Extract EXTWEST successfully upgraded to integrated capture.

GGSCI (AMER)> Start *
Sending START request to MANAGER ...
EXTRACT EXTWEST starting
EXTRACT PWEST is already running.

GGSCI (AMER)>
```

6. Restart the replicat. On the target system EURO_EAST_GGSCI tab, enter the following command:

```
GGSCI (EURO)> Start *
```

7. Make sure that the replication is still working. On the source database AMER_WEST_SQLPLUS tab, enter the following DML:

```
AMER_SQL> INSERT INTO west.account VALUES (9000,9000);
AMER_SQL> COMMIT;
```

8. On the target database EURO_EAST_SQLPLUS tab, enter the following query:

```
EURO_SQL> SELECT * FROM east.account
           WHERE account_number > 6000;

ACCOUNT_NUMBER ACCOUNT_BALANCE
-------------- ---------------
      7000            7000
      8000            8000
      9000            9000
```

Make sure that the last row was added.

9. The only indication that the Replicat is now integrated (as opposed to classic) is in:

```
GGSCI (EURO)> View Report reast
```

where it says:

```
2013-12-04 13:39:41  INFO  OGG-02527  Integrated Replicat does not populate a
trace table.

2013-12-04 13:39:41  INFO  OGG-02545  Parameter GROUPTRANSOPS is ignored by
Integrated Replicat when parallelism is greater than 1.

2013-12-04 13:39:48  INFO  OGG-02528  REPLICAT REAST successfully registered
with database as inbound server OGG$REAST.

2013-12-04 13:39:50  INFO  OGG-02530  Integrated replicat successfully
attached to inbound server OGG$REAST.

... Many lines omitted for clarity ...
```

10. The only indication that the Extract is now integrated (as opposed to classic) is in:

```
GGSCI (AMER)> Info extwest
```

where it says:

```
Log Read Checkpoint  Oracle Integrated Redo Logs
```

and in:

```
GGSCI (AMER)> View Report extwest
```

where it says:

```
2013-12-04 13:35:27  INFO  OGG-02068  Integrated capture successfully attached
to logmining server OGG$CAP_EXTWEST using OGGCapture API.

2013-12-04 13:35:27  INFO  OGG-02086  Integrated Dictionary will be used.

... Many lines omitted for clarity ...
```

This completes Practice 9-4. This completes the practices for Lesson 9. Stop here.

Practices for Lesson 10: Oracle GoldenGate Parameters

Chapter 10

Practices for Lesson 10: Overview

Practices Overview

In this practice, you will modify the existing Manager and Extract processes by using more advanced parameters.

During this practice, you will modify:

- The source system Manager parameters
- The target system Manager parameters
- The Extract parameters and restart the Extract process

You can check your work by looking in `~/labs/Section10/solutions` for answers.

Practice 10-1: Modifying Source Manager Parameters

Tasks

1. Following the instructions in the Oracle GoldenGate documentation (`~/Documents`), modify the Manager parameter file on the source system `AMER_WEST_GGSCI` tab. You will find some of the information in Section *3.5 Recommended Manager Parameters* of *Administering Oracle GoldenGate* (`ggate12c_admin_e29388.pdf`), and other information scattered alphabetically in Section *2.1.2 Summary of Manager Parameters* of *Reference for Oracle GoldenGate* (`ggate12c_reference_oracle_e29392.pdf`).

   ```
   GGSCI (AMER)> Edit Param mgr
   ```

   - Have an Extract group automatically start when the Manager process is started.
     Answer: _____
   - Have all Extract groups automatically restart after abending. Manager should retry the operation three times, waiting one minute between each attempt.
     Answer: _____
   - Report the current lag for each Extract/Replicat group every hour to the application log file.
     Answer: _____
   - Write an informational message to the application log file if the lag for any Extract/Replicat group exceeds 30 minutes.
     Answer: _____
   - Write a critical message to the application log file if the lag for any Extract/ Replicat group exceeds 45 minutes.
     Answer: _____

2. Activate the changes by bouncing (stopping, and starting, or restarting) the Manager process:

   ```
   GGSCI (AMER)> Stop Manager
   Manager process is required by other GGS processes.
   Are you sure you want to stop it (y/n)? y

   Sending STOP request to MANAGER ...
   Request processed.
   Manager stopped.

   GGSCI (AMER)> Start Manager
   Manager started.

   GGSCI (AMER)>
   ```

   Note: You do not have to stop any of the Extracts or Replicats to bounce the Manager. After the Extracts are running, they are independent of the Manager.

3. Verify that the Manager has restarted:

```
GGSCI (AMER)> Info mgr
Manager is running (IP port westhost.15000, Process ID 12655).

GGSCI (AMER)>
```

Your host name and PID will be different.

This completes Practice 10-1. Continue now with Practice 10-2.

Practice 10-2: Modifying the Target Manager Parameters

Tasks

1. Following the instructions in the Oracle GoldenGate documentation (found in the same places as Practice 10-1, Task 1), modify the Manager parameter file on the target `EURO_EAST_GGSCI` tab with the following parameters:

```
GGSCI (EURO)> Edit Param mgr
```

   - Use ports in the range of 15151 through 15199 when starting Oracle GoldenGate processes. See footnote for checking port availability.[1]
     Answer: _____
   - Have all Replicat groups that have a group name of `R*` automatically start when the Manager process is started.
     Answer: _____
   - Have all Replicat groups automatically restart after abending. Manager should retry the operation three times, waiting one minute between each attempt.
     Answer: _____
   - Delete the remote trail files after the Replicat has finished processing them and the trail has been kept for at least two hours.
     Answer: _____
   - Report the current lag for each Extract/Replicat group every hour to the application log file.
     Answer: _____
   - Write an informational message to the application log file if the lag for any Extract/Replicat group exceeds 30 minutes.
     Answer: _____
   - Write a critical message to the application log file if the lag for any Extract/ Replicat group exceeds 45 minutes.
     Answer: _____

[1] **Port Availability:** The basic UNIX command to check for a single free port is:

```
netstat -na | grep LISTEN | grep 15001
```

but it gets trickier when trying to check for a range. Using regex (regular expressions), this will search a range of 15151-15199 inclusive:

```
netstat -na | grep LISTEN | egrep 1515[1-9]\|151[6-9][0-9]
```

2. Activate the changes by bouncing (stopping, and starting, or restarting) the Manager to make the changes take effect:

```
GGSCI (EURO)> Stop Manager
Manager process is required by other GGS processes.
Are you sure you want to stop it (y/n)? y

Sending STOP request to MANAGER ...
Request processed.
Manager stopped.

GGSCI (EURO)> Start Manager
Manager started.

GGSCI (EURO)>
```

Note: You do not have to stop any of the Extracts or Replicats to bounce the Manager. After they are running, they are independent of the Manager.

Note: If you do **Stop mgr !** (by adding an exclamation point!), it will not prompt you for "Are you sure?"

3. Verify that the Manager has started:

```
GGSCI (EURO)> Info mgr
Manager is running (IP port easthost.15001, Process ID 22613).

GGSCI (EURO)>
```

Your host name and PID will be different.

This completes Practice 10-2. Continue now with Practice 10-3.

Practice 10-3: Modifying the Extract Parameters on the Source Database

Tasks

1. Using the Oracle GoldenGate Reference Manual (online or the `ggate12c_reference_oracle_e29392.pdf` file located in the `~/Documents` directory), find the parameters that perform the following and add them to the `extwest.prm` file before the `Table` listings. Modify the Extract parameter file on the source database `AMER_WEST_GGSCI` tab for the following:

   ```
   GGSCI (AMER)> Edit Param extwest
   ```

   - Controls whether or not statistics generated by the `REPORT` parameter are reset when a new process report is created
   - Generates interim runtime statistics in a process report. Set this parameter to report at one minute after midnight daily.
   - Forces report files to age on a regular schedule, instead of when a process starts. Set this report roll-over parameter to occur at one minute after midnight daily.
   - Reports a count of transaction records processed since startup. Set this report parameter to occur every 60 seconds, and report the number of operations per second and the change in rate.
   - Writes a message to the report file after processing every 1000 records.
   - Only extracts data from the `ACCOUNT_TRANS` table if the `account_trans_type` is equal to `"CR"`
     Answer: _____

   You can check your answers against a sample file in `~/labs/Section10/solutions/amer/extwest.prm`. Look for these parameters later in the macro best practices settings `MACRO #bpsettings` in Practice 12-4: “Modifying an Existing Set of Macros and User Tokens.”

2. Activate the changes by stopping, and starting or restarting the Extract to have the changes take effect. Then verify that the Extract has started:

```
GGSCI (AMER)> Stop Extract extwest
Sending STOP request to EXTRACT EXTWEST ...
Request processed.

GGSCI (AMER)> Start Extract extwest
Sending START request to MANAGER ...
EXTRACT EXTWEST starting

GGSCI (AMER)> Info ER *
EXTRACT    EXTWEST   Last Started 2013-12-04 14:24   Status RUNNING
Checkpoint Lag       00:00:06 (updated 00:00:00 ago)
Process ID           12774
Log Read Checkpoint  Oracle Integrated Redo Logs
                     2013-12-04 14:24:47
                     SCN 0.6020566 (6020566)

EXTRACT    PWEST     Last Started 2013-12-04 07:41   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:06 ago)
Process ID           5429
Log Read Checkpoint  File ./dirdat/ew000005
                     2013-12-04 14:24:33.234481  RBA 1479

GGSCI (AMER)>
```

3. Clean up the Oracle GoldenGate lab environment by executing the following GGSCI commands on the target server EURO_EAST_GGSCI tab:

```
GGSCI (EURO)> Stop ER *
GGSCI (EURO)> DBLogin UserIDAlias ggueuro
GGSCI (EURO)> Delete ER *                  (confirm the deletion)
GGSCI (EURO)> Delete CheckpointTable       (only on the EURO db)
GGSCI (EURO)> Stop mgr !
GGSCI (EURO)> Exit
OS_prompt> rm $GG_EURO_HOME/dirdat/*
OS_prompt> rm $GG_EURO_HOME/dirrpt/*
```

Do not type the *(red comments)*; those are only for clarification. Shown below is an example of the dialog from the target (EURO) server:

```
GGSCI (EURO)> Stop ER *
Sending STOP request to REPLICAT REAST ...
Request processed.

GGSCI (EURO)> DBLogin UserIDAlias ggueuro
Successfully logged into database.

GGSCI (EURO)> Delete ER *
Are you sure you want to delete all groups? y
2013-12-04 14:30:54  INFO  OGG-02529  Successfully unregistered REPLICAT REAST
inbound server OGG$REAST from database.
Deleted REPLICAT REAST.

GGSCI (EURO)> Delete CheckpointTable
No checkpoint table specified, using GLOBALS specification (GGS_CHECKPOINT)...
This checkpoint table may be required for other installations.  Are you sure
you want to delete this checkpoint table? y
Successfully deleted checkpoint table GGS_CHECKPOINT.

GGSCI (EURO)> Stop mgr !
Sending STOP request to MANAGER ...
Request processed.
Manager stopped.

GGSCI (EURO)> Exit
[OS_prompt gg_euro]$ rm $GG_EURO_HOME/dirdat/*
[OS_prompt gg_euro]$ rm $GG_EURO_HOME/dirrpt/*
```

4. This cleanup script above will be run in future labs as well. You can find it as a shell script in `~/labs/setup/kill-all-ogg.sh`. `Exit` in any GGSCI sessions on all terminal tabs, then run `kill-all-ogg` for the AMER source server. (It will also re-kill the EURO processes.) From the fifth tab `oracle@hostname` OS prompt, enter:

   `[OS_prompt]` **`~/labs/setup/kill-all-ogg.sh`**

   It is normal for Oracle shell scripts to warn when trying to drop a non-existent object. This is also a good example of a shell script that invokes GGSCI commands. The reason you run this from the fifth tab (and not from either of the GGSCI tabs) is that the script may leave the `oraenv` and `$ORACLE_SID` set to other than what you want.

   You can check your work by looking in `~/labs/Section10/solutions` for answers.

This completes Practice 10-3. This completes the practices for Lesson 10. Stop here.

Practices for Lesson 11: Data Selection and Filtering

Chapter 11

Practices for Lesson 11: Overview

Practices Overview

In these practices, you will increase the performance of Oracle GoldenGate replication by configuring multiple replication streams running in parallel. One of the techniques used to achieve parallelism is based on the `@RANGE` function, which computes a hash value of the columns specified in the input. The `@RANGE` function, combined with the `FILTER` option, allows the Extract/Replicat processes to split the flow of replicated data into two or more streams.

Assumptions

- You completed the lab practices for all previous lessons.
- Nothing is running in GGSCI at the moment.
- You have two configured Extract processes (`extwest` and `pwest`) and one configured Replicat process (`reast`). All processes are currently stopped.

Practice 11-1: Increasing Performance by Splitting Replication Loads

Overview

In this practice, you create a new source table and a stored procedure that populates the source table with random data. The content of the table is replicated using an Extract process, which writes three remote trail files, each consumed on the target environment by three dedicated Replicat processes.

Tasks

1. Create the source and target table `range_split`. Its definition is stored in the `range_split.sql` file. On the fifth tab, `oracle@hostname`, at an OS prompt, navigate to the Section 11 directory, and view the script:

   ```
   [OS_prompt] cat ~/labs/Section11/sqlscripts/range_split.sql
   ```

   ```
   DROP   TABLE range_split;
   CREATE TABLE range_split
   (   ROW_ID      NUMBER NOT NULL PRIMARY KEY,
       split_value NUMBER NOT NULL ,
       split_text  VARCHAR2(128) NOT NULL ,
       split_date TIMESTAMP DEFAULT systimestamp );
   ```

2. You also will create the `populate_range_split` stored procedure, defined in `populate_range_split.sql`. View the PL/SQL code:

   ```
   [OS] cat ~/labs/Section11/sqlscripts/populate_range_split.sql
   ```

   ```
   CREATE OR REPLACE PROCEDURE populate_range_split
   (  num_rows        IN NUMBER
    , commit_interval IN NUMBER  )
   ...many lines omitted for clarity...
   ```

   The stored procedure takes two parameters:

   - The number of rows to be created
   - The commit interval (in number of rows)

   The PL/SQL stored procedure defines a cursor used to generate random strings, then loops until the number of rows to be created has been reached, generating random numbers and random strings and storing them in the `RANGE_SPLIT` table.

3. On the `AMER_WEST_SQLPLUS` tab, connect to the source Oracle database by using SQL*Plus. If you are already connected as `west@amer`, skip to the next task.

   ```
   [OS_prompt labs] rlsqlplus west@amer/oracle_4U
   SQL*Plus: Release 12.1.0.1.0 Production on Thu Dec 5 08:28:43 2013
   Copyright (c) 1982, 2013, Oracle.  All rights reserved.
   Last Successful login time: Wed Dec 04 2013 07:58:18 +00:00
   Connected to:
   Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit Production
   With Partitioning, OLAP, Advanced Analytics, Real Application Testing options
   SQL> set sqlprompt AMER_SQL>
   ```

4. Use the @ command in SQL*Plus to create the RANGE_SPLIT table:

```
AMER_SQL> @Section11/sqlscripts/range_split.sql
Table created.
```

As always, if the initial table to be dropped does not exist, you get an error that you can ignore. As long as the script says `Table created` at the end, it worked.

5. Load and compile the stored procedure by using the @ command into the `west` schema:

```
AMER_SQL> @Section11/sqlscripts/populate_range_split.sql
Procedure created.
```

6. Change to the source AMER_WEST_GGSCI tab, and make sure that the GGSCI command interpreter and manager are started:

```
[OS prompt ~] cd $GG_AMER_HOME
[OS prompt gg_amer] rlggsci
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.0.0 17185003 OGGCORE_12.1.2.0.0_PLATFORMS_130924.1316_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.
Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.

GGSCI (AMER) 1> Start mgr
Manager started.
```

It may be helpful to eventually define a `startup.oby` "obey" script file that contains the half-dozen or so common GGSCI housekeeping tasks, such as starting the manager, doing `DBLogin`, setting the default editor, and so on.

7. You must set the additional logging for the new RANGE_SPLIT table:

```
GGSCI (AMER) 2> DBLogin UserIDAlias gguamer
Successfully logged into database.

GGSCI (AMER) 3> Add TranData WEST.RANGE_SPLIT
Logging of supplemental redo data enabled for table WEST.RANGE_SPLIT.
TRANDATA for scheduling columns has been added on table 'WEST.RANGE_SPLIT'.

GGSCI (AMER) 4> Info TranData WEST.R*
```

8. Edit the parameters for `defgen` and add the RANGE_SPLIT table:

```
GGSCI (AMER) 5> Edit Param defsrc
DefsFile ./dirdef/rangesplit.def, Purge
UserIDAlias gguamer
Table WEST.RANGE_SPLIT;
```

9. Run the defgen utility from the OS command shell to generate the table definitions. You could exit GGSCI, run defgen, and then restart GGSCI. However, it is easier to just run defgen from the sh command inside of GGSCI. Note that defgen is *NOT* a GGSCI command. The response should look similar to:

```
GGSCI (AMER)> sh ./defgen paramfile dirprm/defsrc.prm
```

```
...many lines omitted for clarity...
*******************************************************************
**            Running with the following parameters              **
*******************************************************************
DefsFile ./dirdef/rangesplit.def, Purge
UserIDAlias gguamer
Table WEST.RANGE_SPLIT;
Retrieving definition for WEST.RANGE_SPLIT

Definitions generated for 1 table in ./dirdef/rangesplit.def
```

10. Copy the rangesplit.def file to the EURO target environment, under the dirdef directory:

```
GGSCI (AMER)> sh cp ./dirdef/*.def  $GG_EURO_HOME/dirdef
GGSCI (AMER)> sh ls $GG_EURO_HOME/dirdef
rangesplit.def
```

In a production environment, you might need to use scp or sftp instead of cp if the target is a remote host.

11. Change to the EURO_EAST_SQLPLUS tab and connect to the target EURO database by using SQL*Plus to create the target RANGE_SPLIT table. If you are already connected as east@euro, skip to the next task.

```
[OS_prompt labs] rlsqlplus east@euro/oracle_4U
```

12. Create the empty target RANGE_SPLIT table:

```
EURO_SQL> @Section11/sqlscripts/range_split.sql
Table created.
```

As before, an attempt to drop a non-existent table results in an error that you can safely ignore.

13. On the AMER_WEST_GGSCI tab, create three Extract processes (erangea, erangeb, and erangec), which use the FILTER command combined with the @RANGE function to create three separate trail files. These files will be processed by three separate Replicat processes, thus achieving parallelism and significantly improving load performance. Note where the files differ by a, b, c, and by 1, 2, 3:

```
GGSCI (AMER)> Edit Param erangea
```

```
Extract erangea
UserIDAlias gguamer
RmtHost easthost, MgrPort 15001
RmtTrail ./dirdat/ea
Table WEST.RANGE_SPLIT, Filter (@RANGE (1, 3));
```

14. Save the parameter file `erangea` and create the second parameter file:

```
GGSCI (AMER)> Edit Param erangeb
```

```
Extract erangeb
UserIDAlias gguamer
RmtHost easthost, MgrPort 15001
RmtTrail ./dirdat/eb
Table WEST.RANGE_SPLIT, Filter (@RANGE (2, 3));
```

15. Save the parameter file `erangeb` and create the third parameter file:

```
GGSCI (AMER)> Edit Param erangec
```

```
Extract erangec
UserIDAlias gguamer
RmtHost easthost, MgrPort 15001
RmtTrail ./dirdat/ec
Table WEST.RANGE_SPLIT, Filter (@RANGE (3, 3));
```

16. Add the three Extract processes:

```
GGSCI (AMER)> Add Extract erangea, TranLog, Begin Now
EXTRACT added.
GGSCI (AMER)> Add Extract erangeb, TranLog, Begin Now
EXTRACT added.
GGSCI (AMER)> Add Extract erangec, TranLog, Begin Now
EXTRACT added.
```

17. Add the three remote trail definitions:

```
GGSCI (AMER)> Add RmtTrail ./dirdat/ea, Extract erangea
RMTTRAIL added.
GGSCI (AMER)> Add RmtTrail ./dirdat/eb, Extract erangeb
RMTTRAIL added.
GGSCI (AMER)> Add RmtTrail ./dirdat/ec, Extract erangec
RMTTRAIL added.
```

18. Assess the current situation of your Extract processes by entering `Info All`:

```
GGSCI (AMER)> Info All
```

```
Program     Status      Group       Lag at Chkpt  Time Since Chkpt
MANAGER     RUNNING
EXTRACT     STOPPED     ERANGEA     00:00:00      00:02:35
EXTRACT     STOPPED     ERANGEB     00:00:00      00:02:28
EXTRACT     STOPPED     ERANGEC     00:00:00      00:02:20
```

19. Start the three new Extract processes:

```
GSCI (AMER)> Start Extract erange*

Sending START request to MANAGER ...
EXTRACT ERANGEA starting

Sending START request to MANAGER ...
EXTRACT ERANGEB starting

Sending START request to MANAGER ...
EXTRACT ERANGEC starting
```

20. Issue `Info All` once again, to verify that no process is in an ABENDED status:

```
GGSCI (AMER)> Info All

Program     Status      Group       Lag at Chkpt  Time Since Chkpt
MANAGER     RUNNING
EXTRACT     RUNNING     ERANGEA     00:00:00      00:05:53
EXTRACT     RUNNING     ERANGEB     00:00:00      00:05:46
EXTRACT     RUNNING     ERANGEC     00:00:00      00:05:38
```

21. On the EURO_EAST_GGSCI tab, start GGSCI:

```
[OS_prompt ~] cd $GG_EURO_HOME
[OS_prompt gg_euro] rlggsci
```

22. Do basic housekeeping in preparation for configuration. Start the Manager, log in to the database, and make sure that there is a Checkpoint table.

```
GGSCI (EURO) 1> DBLogin UserIDAlias ggueuro
GGSCI (EURO) 2> Start Manager
GGSCI (EURO) 3> Info CheckpointTable
```

If the checkpoint table exists, then skip the next line; otherwise create the checkpoint table.

```
GGSCI (EURO) 4> Add CheckpointTable
GGSCI (EURO) 5> Set Editor gedit                ← optional
```

Optional: If this housekeeping seems repetitive for each time you start a new GGSCI session, look at `~/labs/setup/startup.oby`.

23. Create three Replicat processes (`rrangea, rrangeb`, and `rrangec`), which source their data from the trail files created by the three Extract processes created previously.

```
GGSCI (EURO)> Edit Param rrangea

Replicat rrangea
UserIDAlias ggueuro
SourceDefs ./dirdef/rangesplit.def
DiscardFile ./dirrpt/rrangea.dsc, Append
Map WEST.RANGE_SPLIT, Target EAST.RANGE_SPLIT;
```

24. Save the parameter file `rrangea` and create the second parameter file:

```
GGSCI (EURO)> Edit Param rrangeb
```

```
Replicat rrangeb
UserIDAlias ggueuro
SourceDefs ./dirdef/rangesplit.def
DiscardFile ./dirrpt/rrangeb.dsc, Append
Map WEST.RANGE_SPLIT, Target EAST.RANGE_SPLIT;
```

25. Save the parameter file `rrangeb` and create the third parameter file:

```
GGSCI (EURO)> Edit Param rrangec
```

```
Replicat rrangec
UserIDAlias ggueuro
SourceDefs ./dirdef/rangesplit.def
DiscardFile ./dirrpt/rrangec.dsc, Append
Map WEST.RANGE_SPLIT, Target EAST.RANGE_SPLIT;
```

26. Add the three Replicat processes, connecting them to their respective trail files:

```
GGSCI (EURO)> Add Replicat rrangea, exttrail ./dirdat/ea
REPLICAT added.
GGSCI (EURO)> Add Replicat rrangeb, exttrail ./dirdat/eb
REPLICAT added.
GGSCI (EURO)> Add Replicat rrangec, exttrail ./dirdat/ec
REPLICAT added.
```

27. Start the Replicat processes and enter `info all` to verify that they started successfully:

```
GGSCI (EURO)> Start Replicat rrange*
Sending START request to MANAGER ...
REPLICAT RRANGEA starting

Sending START request to MANAGER ...
REPLICAT RRANGEB starting

Sending START request to MANAGER ...
REPLICAT RRANGEC starting

GGSCI (EURO)> Info All
Program     Status      Group       Lag at Chkpt  Time Since Chkpt
MANAGER     RUNNING
REPLICAT    RUNNING     RRANGEA     00:00:00      00:00:02
REPLICAT    RUNNING     RRANGEB     00:00:00      00:00:02
REPLICAT    RUNNING     RRANGEC     00:00:00      00:00:02
```

Check the status of the source GGSCI processes as well. It *may* have a problem… Why?

28. Go to the `AMER_WEST_SQLPLUS` tab, and execute the stored procedure which populates the table `WEST.RANGE_SPLIT`:

```
AMER_SQL> exec populate_range_split(500000,1000);
```

You will not see anything while it executes. This is normal. Keep reading the next steps.

29. The stored procedure inserts half a million rows, committing every one thousand rows. It takes a few minutes to complete. During its execution, you can monitor the Extract and Replicat processes to verify that data is being replicated. Click the AMER_WEST_GGSCI tab where you are running the source Oracle GoldenGate installation and use the statistics command to show total DMLs on the Extract process:

```
GGSCI (AMER)> Stats Extract erangea
Sending STATS request to EXTRACT ERANGEA ...

Start of Statistics at 2013-12-05 14:43:37.

Output to ./dirdat/ea:

Extracting from WEST.RANGE_SPLIT to WEST.RANGE_SPLIT:

...many lines omitted for clarity...

*** Latest statistics since 2013-12-05 14:43:14 ***
    Total inserts                          92846.00
    Total updates                              0.00
    Total deletes                              0.00
    Total discards                             0.00
    Total operations                       92846.00

End of Statistics.
```

30. Go to the EURO_EAST_GGSCI tab for the target environment, and display statistics for the Replicat processes:

```
GGSCI (EAST)> Stats Replicat rrangea
Sending STATS request to REPLICAT RRANGEA...

Start of Statistics at 2013-12-05 14:44:50.

Replicating from WEST.RANGE_SPLIT to EAST.RANGE_SPLIT:

...many lines omitted for clarity...

*** Latest statistics since 2013-12-05 14:43:17 ***
    Total inserts                          28964.00
    Total updates                              0.00
    Total deletes                              0.00
    Total discards                             0.00
    Total operations                       28964.00

End of Statistics.
```

```
GGSCI (EURO)> Stats Replicat rrangeb
[omitted similar output…]
GGSCI (EURO)> Stats Replicat rrangec
[omitted similar output…]
```

31. A short time after the stored procedure finishes its execution, the replication ends (not to be confused with the Replicat which keeps running until you stop it). You can assess the status of your replicated rows by viewing the reports on the Extract and Replicat processes from GGSCI:

```
GGSCI (AMER)> Stop ER *
GGSCI (AMER)> View Report erangea
```

```
GGSCI (EURO)> Stop ER *
GGSCI (EURO)> View Report rrangea
```

The reports will not contain final stats until you stop the processes with `Stop ER *` or equivalent.

32. View the report produced for all Extract processes (`erangea`, `erangeb`, and `erangec`) and all Replicat processes (`rrangea`, `rrangeb`, and `rrangec`) and make sure that replication was successful at both ends, and no rows were missed (`discards: 0`). Were all the Extracts identical in count? Why or why not?

This completes Practice 11-1. Continue now with Practice 11-2.

Practice 11-2: Increasing Performance by Coordinated Applies

Overview

This practice modifies the previous practice by changing the `@RANGE()` to a coordinated replicat `ThreadRange()`, which is new with 12*c*.

Tasks

1. Stop all previous Extract and Replicat processes:
   - On the source `AMER_WEST_GGSCI` tab, enter:
     ```
     GGSCI (AMER)> Stop ER *
     ```
   - On the target `EURO_EAST_GGSCI` tab, enter:
     ```
     GGSCI (EURO)> Stop ER *
     ```
2. Truncate both source and target tables in SQL*Plus as you will reuse them:
   a) On the source database `AMER_WEST_SQLPLUS` tab, enter:
   ```
   AMER_SQL> TRUNCATE TABLE west.range_split;
   ```
   b) On the target database `EURO_EAST_SQLPLUS` tab, enter:
   ```
   EURO_SQL> TRUNCATE TABLE east.range_split;
   ```
3. On the `AMER_WEST_GGSCI` tab, create a new parameter file `ecord` on the source:
   ```
   GGSCI (AMER)> Edit Param ecord
   ```
   ```
   Extract ecord
   UserIDAlias gguamer
   RmtHost easthost, MgrPort 15001
   RmtTrail ./dirdat/ed
   Table WEST.RANGE_SPLIT;
   ```
   Save and close the file. Note that there is nothing in the Extract that indicates anything "coordinated" about the apply. The only difference between this Extract and the previous three was removing the Table clause:
   ```
   , Filter (@RANGE (n, 3))
   ```
   and the fact that you needed three of them. Now we only need one Extract.
4. On the `EURO_EAST_GGSCI` tab, create a new parameter file `rcord` on the target:
   ```
   GGSCI (EURO)> Edit Param rcord
   ```
   ```
   Replicat rcord
   UserIDAlias ggueuro
   SourceDefs ./dirdef/rangesplit.def
   DiscardFile ./dirrpt/rrangea.dsc, Append
   Map WEST.RANGE_SPLIT, Target EAST.RANGE_SPLIT
     ThreadRange (1-3, ROW_ID);
   ```
   The replicat group name must be five characters or less, or you get the following error:
   ```
   ERROR: Invalid Coordinated Replicat group name.  Must be at most
   5 characters..
   ```

This is because it will create `rcord001`, `rcord002`, …, `rcord`*`nnn`*, and so on. Note the new map clause:

`ThreadRange (1-`*`nnn`*`, `*`column`*`)`

and the fact that you only need one Replicat. Save and close the file.

5. On the AMER_WEST_GGSCI tab, add and start the source processes:

```
GGSCI (AMER)> Add Extract ecord, TranLog, Begin Now
EXTRACT added.

GGSCI (AMER)> Add RmtTrail ./dirdat/ed, Extract ecord
RMTTRAIL added.

GGSCI (AMER)> Start ER ec*
Sending START request to MANAGER ...
EXTRACT ECORD starting
```

6. On the EURO_EAST_GGSCI tab, add and start the target processes (all on one line; no breaks):

```
GGSCI (EURO)> Add Replicat rcord, Coordinated, ExtTrail
                ./dirdat/ed, MaxThreads 3
REPLICAT (Coordinated) added.

GGSCI (EURO)> Start ER rc*
Sending START request to MANAGER ...
REPLICAT RCORD starting

GGSCI (EURO) 41> Info rcord

REPLICAT   RCORD     Last Started 2013-12-05 12:04   Status RUNNING
COORDINATED          Coordinator                      MAXTHREADS 3
Checkpoint Lag       00:00:00 (updated 00:00:00 ago)
Process ID           18375
Log Read Checkpoint  File ./dirdat/ed000000
                     First Record  RBA 1415

GGSCI (EURO)>
```

7. Generate DML traffic. On the AMER_WEST_SQLPLUS source database tab, enter:

```
AMER_SQL> exec populate_range_split(10000,1000);
```

8. On the target EURO_EAST_GGSCI tab, view the process and thread information. Enter:

```
GGSCI (EURO)> Info rcord, Detail

REPLICAT   RCORD     Last Started 2013-12-06 08:29   Status RUNNING
COORDINATED          Coordinator                      MAXTHREADS 3
Checkpoint Lag       00:00:00 (updated 00:00:02 ago)
Process ID           18375
Log Read Checkpoint  File ./dirdat/ed000000
                     2013-12-06 12:15:08.818086  RBA 3169467

Lowest Log BSN value: <NULL>

Active Threads:
ID  Group Name PID   Status   Lag at Chkpt  Time Since Chkpt
1   RCORD001   18382 RUNNING  00:00:00      00:00:02
2   RCORD002   18383 RUNNING  00:00:00      00:00:02
3   RCORD003   18384 RUNNING  00:00:00      00:00:02

Current directory    /u01/app/oracle/product/gg_euro

Report file          /u01/app/oracle/product/gg_euro/dirrpt/RCORD.rpt
Parameter file       /u01/app/oracle/product/gg_euro/dirprm/rcord.prm
Checkpoint file      /u01/app/oracle/product/gg_euro/dirchk/RCORD.cpr
Checkpoint table     GGS_CHECKPOINT
Process file         /u01/app/oracle/product/gg_euro/dirpcs/RCORD.pcr
Error log            /u01/app/oracle/product/gg_euro/ggserr.log

GGSCI (EURO)> Info rcord002

REPLICAT   RCORD002  Last Started 2013-12-06 08:29   Status RUNNING
COORDINATED          Replicat Thread                  Thread 2
Checkpoint Lag       00:00:00 (updated 00:00:06 ago)
Process ID           18383
Log Read Checkpoint  File ./dirdat/ed000000
                     2013-12-06 08:30:52.000182  RBA 3169467

GGSCI (EURO)> Info rcord003

REPLICAT   RCORD003  Last Started 2013-12-06 08:29   Status RUNNING
COORDINATED          Replicat Thread                  Thread 3
Checkpoint Lag       00:00:00 (updated 00:00:00 ago)
Process ID           18384
Log Read Checkpoint  File ./dirdat/ed000000
                     2013-12-06 08:30:52.000182  RBA 3169467
GGSCI (EURO)>
```

9. On the target EURO_EAST_GGSCI tab, view the Replicat statistics for all threads. Enter:

```
GGSCI (EURO)> Stats rcord

Sending STATS request to REPLICAT RCORD ...
Coordinated Replicat Statistics:

Thread Lag Gap:                                          00:00:00 (updated
00:00:09 ago)
Coordinated Total DDLs:                                  0
Coordinated Total PK-update transactions:                0
Coordinated Total EMI transactions:                      0
Total transactions with user-requested coordination:     0
Average Coordination Time:                               00:00:00

Start of Statistics at 2013-12-06 12:21:15.
Replicating from WEST.RANGE_SPLIT to EAST.RANGE_SPLIT:

...many lines omitted for clarity...

*** Latest statistics since 2013-12-06 08:30:54 ***
     Total inserts                           10000.00
     Total updates                               0.00
     Total deletes                               0.00
     Total discards                              0.00
     Total operations                        10000.00

End of Statistics.

GGSCI (EURO)> Stats rcord002

Sending STATS request to REPLICAT RCORD002 ...
Start of Statistics at 2013-12-06 12:21:23.
Replicating from WEST.RANGE_SPLIT to EAST.RANGE_SPLIT:

...many lines omitted for clarity...

*** Latest statistics since 2013-12-06 08:30:54 ***
     Total inserts                            3346.00
     Total updates                               0.00
     Total deletes                               0.00
     Total discards                              0.00
     Total operations                         3346.00

End of Statistics.
```

```
GGSCI (EURO)> Stats rcord003

Sending STATS request to REPLICAT RCORD003 ...
Start of Statistics at 2013-12-06 12:21:26.
Replicating from WEST.RANGE_SPLIT to EAST.RANGE_SPLIT:

...many lines omitted for clarity...

*** Latest statistics since 2013-12-06 08:30:54 ***
     Total inserts                            3322.00
     Total updates                               0.00
     Total deletes                               0.00
     Total discards                              0.00
     Total operations                         3322.00

End of Statistics.

GGSCI (EURO)>
```

Notice that the division of labor among the three threads is *approximately* equal but not exactly.

This completes Practice 11-2. This completes the practices for Lesson 11. Stop here.

Practices for Lesson 12: Additional Transformation Topics

Chapter 12

Practices for Lesson 12: Overview

Practices Overview

In this practice, you will learn more about the structure and parameters that make up Oracle GoldenGate macros and user tokens. These will be used in subsequent labs.

Practice 12-1: Modifying an Existing Set of Macros and Using User Tokens

Overview

In this practice, you will create three macros. Oracle GoldenGate Macros provide functionality for sharing parameters or other runtime configuration settings across multiple components and externalizing complex configuration settings to streamline parameter file contents.

The best practice is to create a file or series of files as a macro library and store them in a specific folder (for example, *$OGG_HOME*/dirmac).

1. On the fifth oracle@hostname tab, at the OS prompt, edit the ~/labs/Section12/macrolib_shell.mac file in the text editor of your choice (for example, vi, gedit). Referring to the *Oracle GoldenGate Windows and UNIX Reference Guide* (~/Documents/ggate12c_reference_oracle_e29392.pdf), fill in the ***<blanks>*** with the appropriate values to complete the three macros:
   - #dbconnect
   - #bpsettings
   - #funcsmap

   <a> The keyword that starts a macro body

   <b> The keyword that finishes a macro

   <c> Reset the statistics when a new report is generated

   <d> Generates a report every day at one minute after midnight

   <e> Close the current report file and create a new one daily at one minute after midnight

   <f> The keyword that maps records between different source and target columns

   <g> The function that is used to return information about the Oracle GoldenGate environment

   <h> The function used to identify a user token

```
MACRO #dbconnect
<a>
UserID gguser, Password <encrypted_pswd> AES256, EncryptKey MyKey3
<b>;

MACRO #bpsettings
BEGIN
<c>
<d>
<e>
ReportCount Every 60 Seconds, Rate
END;

MACRO #funcsmap
PARAMS (#src_table, #target_table)
BEGIN
   MAP #src_table, TARGET #target_table,
    <f> (usedefaults,
            gg_commit_ts = <g> ( 'GGHEADER' , 'COMMITTIMESTAMP' ),
            lag_extract_ms = <h> ( 'TKN-EXTLAG-MSEC' ),
            lag_replicat_ms = @GETENV ( 'LAG' , 'MSEC' ),
            src_db_name = @TOKEN ( 'TKN-SRC-DBNAME' ),
            src_db_version = @TOKEN ( 'TKN-SRC-DBVERSION' ),
            src_txn_csn = @TOKEN ( 'TKN-TXN-CSN' )
       );
END;
```

After all of the parameters have been set correctly, save and close the file. It is very important that the token quotes be 'single, straight' quotes. Verify that it has the correct parameters by comparing it with the `macrolib.mac` file, which is also located in the `~/labs/Section12` directory.

2. On the `AMER_WEST_GGSCI` source tab, create a directory named `dirmac`. Copy the `macrolib.mac` file to this directory:

```
GGSCI (AMER)> sh mkdir dirmac
GGSCI (AMER)> sh cp ~/labs/Section12/macrolib.mac dirmac
GGSCI (AMER)> sh ls dirmac
macrolib.mac
GGSCI (AMER)>
```

3. On the EURO_EAST_GGSCI target tab, perform the same steps. Create a directory named dirmac. Copy the macrolib.mac file to this directory.

```
GGSCI (EURO)> sh mkdir dirmac
GGSCI (EURO)> sh cp ~/labs/Section12/macrolib.mac dirmac
GGSCI (EURO)> sh ls dirmac
macrolib.mac
GGSCI (EURO)>
```

Note: Both the macros and user tokens will be used in subsequent labs.

This completes Practice 12-1. This completes the practices for Lesson 12. Stop here.

Practices for Lesson 13: Configuration Options

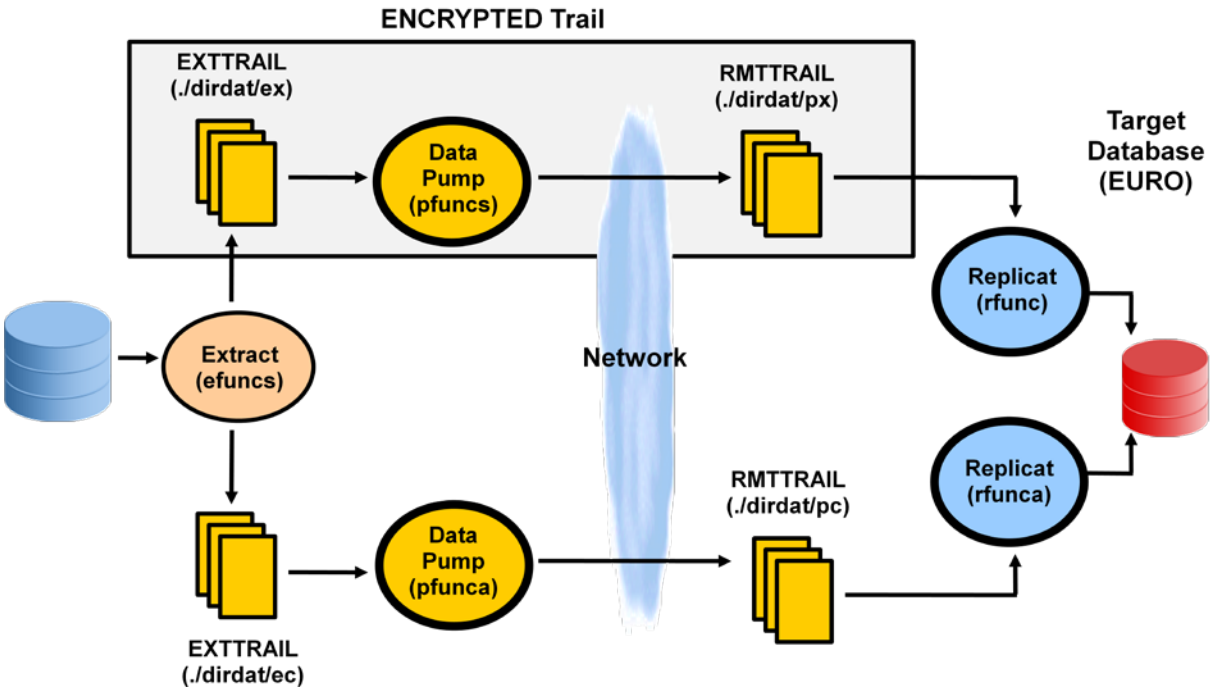
Chapter 13

Practices for Lesson 13: Overview

Practices Overview

In this practice, you will set up a configuration as depicted in the following graphic. You will also use the macros and tokens created in the previous practice, including password encryption, trail encryption, and SQLEXEC.

Big Picture



Practice 13-1: Setting Up the Database and Source Definitions File

Overview

In this practice, you set up the new tables in the databases and create a `defgen` file.

Assumptions

You have successfully completed the previous lab where you created the macros. If the SQL*Plus tabs are still signed on as the proper users, you can just execute the script; no need to exit/restart.

Tasks

Adding Transaction Data

1. On the source database `AMER_WEST_SQLPLUS` tab, execute the following command:

   ```
   [OS_prompt labs] sqlplus west@amer/oracle_4U
                    @Section13/sqlscripts/source_database.sql
   ```

   As always, the first time you see the "`ORA-00942: table or view does not exist`" warning in a SQL script, it can probably be safely ignored.

2. On the target database `EURO_EAST_SQLPLUS` tab, execute the following command:

   ```
   [OS_prompt labs] sqlplus east@euro/oracle_4U
                    @Section13/sqlscripts/target_database.sql
   ```

3. On the source system `AMER_WEST_GGSCI` tab, configure the database to log table key values by issuing the following commands. If you are still logged in from a previous lab, skip the first three lines and just continue with the `Add TranData` step.

   ```
   [OS_prompt] cd $GG_AMER_HOME
   [OS_prompt gg_amer] rlggsci
   GGSCI (AMER) 1> DBLogin UserIDAlias gguamer
   GGSCI (AMER) 2> Add TranData WEST.*
   ```

   Note: Supplemental logging has already been enabled for the existing tables in the database and will cause many `WARNING OGG-00706` messages that can be ignored. Verify that supplemental logging has been enabled for the new tables: `WEST.CUST_ZIP`, `WEST.WSHOP_ENCRYPT`, and `WEST.WSHOP_FUNCS`.

Generating Source Table Definitions

4. Because the target tables differ from the source, a source definitions file must be generated for use in Replicat to perform the table name, column name, and data type conversions. The `defgen` utility is used to create this file. To create the `defgen` file, on the source `AMER_WEST_GGSCI` tab, execute the following command:

   ```
   GGSCI (AMER)> Edit Param defgen
   ```

   The text editor of your choice opens the empty `defgen.prm` file. Add the following runtime parameters to the text file:

   ```
   DefsFile ./dirdef/section13.defs, Purge
   UserIDAlias gguamer
   Table WEST.wshop_encrypt;
   Table WEST.wshop_funcs;
   Table WEST.cust_zip;
   ```

   You could have used a wildcard for `Table WEST.wshop*`. Save and close the file.

5. On the `AMER_WEST_GGSCI` tab, execute the OS command:

   ```
   GGSCI (AMER)> sh ./defgen paramfile ./dirprm/defgen.prm
   ...many lines omitted for clarity...

   ************************************************************************
   **            Running with the following parameters                   **
   ************************************************************************
   DefsFile ./dirdef/section13.defs, Purge
   UserIDAlias gguamer
   Table WEST.wshop_encrypt;
   Retrieving definition for WEST.WSHOP_ENCRYPT.
   Table WEST.wshop_funcs;
   Retrieving definition for WEST.WSHOP_FUNCS.
   Table WEST.cust_zip;
   Retrieving definition for WEST.CUST_ZIP.

   Definitions generated for 3 tables in ./dirdef/section13.defs.

   GGSCI (AMER)>
   ```

   The `defgen` utility will report runtime data to the screen. If any errors are reported, fix them and rerun the utility.

6. On the source `AMER_WEST_GGSCI` tab, copy the `dirdef/section13.defs` file to the same location on the target server:

   ```
   GGSCI (AMER)> sh cp dirdef/section13.defs $GG_EURO_HOME/dirdef
   ```

7. On the `EURO_EAST_GGSCI` tab, verify that the `section13` definition file is there:

   ```
   GGSCI (EURO)> sh ls -al dirdef/s*
   -rw-r----- 1 oracle oinstall 2098 Dec 6  dirdef/section13.defs
   ```

This completes Practice 13-1. Continue now with Practice 13-2.

Practice 13-2: Oracle GoldenGate Encryption Using ENCKEYS

Overview

Oracle GoldenGate offers three types of encryption: Oracle GoldenGate Trail, Data Transmission, and password. In this practice, you will use the AES256 FIPS-compliant encryption to encrypt the password for the database login.

Tasks

1. Before data encryption can occur, encryption keys must be generated using the `keygen` utility.

   To run `keygen` and create keys that will be used for password and data transmission encryption, on the fifth `oracle@hostname` tab Linux shell session, navigate to the `$GG_AMER_HOME` directory. Using the `keygen` utility, create five keys, with one having 64 bits, one with 128 bits, and three having 256 bits. Then open the file with the text editor of your choice:

   ```
   [OS_prompt] cd $GG_AMER_HOME
   [OS_prompt gg_amer] ./keygen 64 1 > ENCKEYS
   [OS_prompt gg_amer] ./keygen 128 1 >> ENCKEYS
   [OS_prompt gg_amer] ./keygen 256 3 >> ENCKEYS
   [OS_prompt gg_amer] gedit ENCKEYS
   ```

   The double greater-than ">>" appends the standard output to a file.

2. Name the keys `MyKey1` through `MyKey5` by adding the prefixes. Your hex numbers will be different. Save and close the file:

   ```
   MyKey1  0x449FFE730EF5634F
   MyKey2  0x1F1914181FDE5B2048A11E5B0F712377
   MyKey3  0xBEE3A6170DCA34671DBF8D12524DD403312ECE240529CA5C37A7540C7F790642
   MyKey4  0xB177611C9714FB1DC1A6936BA8CAE2359B0D642EF3FB1609CAD9914DA13BA45E
   MyKey5  0x4EB616219242DA1B110026107AAB910530AC224BC874E40E0421B9113114A008
   ```

3. Copy this file to the same location on the target (`GG_EURO_HOME`) system:

   ```
   [OS_prompt gg_amer] cp ENCKEYS $GG_EURO_HOME
   ```

   Verify that the file is there in the target installation directory:

   ```
   [OS_prompt gg_amer] more $GG_EURO_HOME/ENCKEYS
   MyKey1  0x449FFE730EF5634F
   MyKey2  0x1F1914181FDE5B2048A11E5B0F712377
   MyKey3  0xBEE3A6170DCA34671DBF8D12524DD403312ECE240529CA5C37A7540C7F790642
   MyKey4  0xB177611C9714FB1DC1A6936BA8CAE2359B0D642EF3FB1609CAD9914DA13BA45E
   MyKey5  0x4EB616219242DA1B110026107AAB910530AC224BC874E40E0421B9113114A008
   [OS_prompt gg_amer]
   ```

Password Encryption Using Oracle GoldenGate Default Encryption

4. Having clear text database login credentials stored in edit files may be deemed a security violation. To solve this issue, Oracle GoldenGate provides a mechanism for encrypting the database access password stored in its configuration files.

   The `Encrypt Password` GGSCI command is used to accomplish this task. Two different levels of encryption are offered: AES and Blowfish.

   The `Encrypt Password <password>` GGSCI command uses the default Oracle GoldenGate encryption key to encrypt the provided text. The returned encrypted text is then copied to the parameter file in place of the clear text password. For correct decryption, the default key must be specified in the parameter file as well.

   Navigate to the `$GG_AMER_HOME/dirmac` directory to retrieve the file that was completed in Practice 12 (`macrolib.mac`).

   To encrypt the password used by the Oracle GoldenGate macro `#dbconnect`, perform the following:

   a. On the source `AMER_WEST_GGSCI` tab, execute the following command: `Encrypt Password oracle_4U`. If you are still logged in to GGSCI from a previous lab, skip the first two steps and just continue with the `Encrypt` command.

```
[OS_prompt ~] cd $GG_AMER_HOME
[OS_prompt gg_amer] rlggsci

GGSCI 1> Encrypt Password oracle_4U AES256 EncryptKey MyKey3
Encrypted password:
AADAAAAAAAAAAAJAMEHJTEOAXDFHGGSILHRGKAXIFJUGMCIEDECEUCXFAFWEWADFECLCEEUFAIVAJH
QEIEKETECCOCJBAGJJGGAGYFRBVDLCHDIB
Algorithm used:  AES256

GGSCI 2>
```

   b. Copy the encrypted password output (not the label prompt, just the hex password itself) into the clipboard with no line breaks.

   c. Use your text editor of choice to edit the existing `AMER` copy of the `macrolib.mac` file.

```
GGSCI (AMER)> sh gedit dirmac/macrolib.mac
```

   d. In the `#dbconnect` macro, replace <*encrypted_password*> with the copied value.

```
MACRO #dbconnect
BEGIN
Userid gguser, Password
AADAAAAAAAAAAAJAMEHJTEOAXDFHGGSILHRGKAXIFJUGMCIEDECEUCXFAFWEWADFECLCEEUFAIVAJH
QEIEKETECCOCJBAGJJGGAGYFRBVDLCHDIB AES256, EncryptKey MyKey3
END;

MACRO #bpsettings
BEGIN
... Many lines omitted for clarity ...
```

   Note: All the way from `Userid` through `MyKey3` is all one line with no line breaks.

   e. Save the file and exit.

f. Replace the same value in the `macrolib.mac` file on the EURO database (`/u01/app/oracle/product/gg_euro/dirmac/macrolib.mac`).

Setting Oracle GoldenGate Change Data Capture (efuncs)

5. On the source server AMER_WEST_GGSCI tab, configure the change data capture extract from a GGSCI prompt:

```
GGSCI (AMER)> Edit Param efuncs
```

The editor of your choice opens the empty `efuncs.prm` file. Add the following runtime parameters to the text file:

```
NoList
Include ./dirmac/macrolib.mac
List
Extract efuncs
EncryptTrail AES256 KeyName MyKey3
ExtTrail ./dirdat/ex

#dbconnect ()
#bpsettings ()
Table WEST.WSHOP_ENCRYPT;

NoEncryptTrail
ExtTrail ./dirdat/ec
Table WEST.WSHOP_ENCRYPT;
Table WEST.CUST_ZIP;
Table WEST.WSHOP_FUNCS, TOKENS (
    TKN-EXTLAG-MSEC = @GETENV ( 'LAG' , 'MSEC' ),
    TKN-SRC-DBNAME = @GETENV ( 'DBENVIRONMENT' , 'DBNAME' ),
    TKN-SRC-DBVERSION = @GETENV ( 'DBENVIRONMENT' , 'DBVERSION' ),
    TKN-TXN-CSN = @GETENV ( 'TRANSACTION' , 'CSN' )
);
```

Save and close the file.

6. On the source server AMER_WEST_GGSCI tab, add the Extract group and two local Extract Trails by executing the following GGSCI commands:

```
GGSCI (AMER)> DBLogin UserIDAlias gguamer
Successfully logged into database.

GGSCI (AMER)> Add Extract efuncs, TranLog, Begin Now
EXTRACT added.

GGSCI (AMER)> Add ExtTrail ./dirdat/ex, Extract efuncs
EXTTRAIL added.

GGSCI (AMER)> Add ExtTrail ./dirdat/ec, Extract efuncs
EXTTRAIL added.

GGSCI (AMER)>
```

Notice this Extract group will have *two* trails: one encrypted (ex) and one plain text (ec).

Setting Up the First GoldenGate Extract Data Pump (pfuncs)

7. On the source server AMER_WEST_GGSCI tab, configure the first Extract Data Pump to read from the local Extract Trail ./dirdat/ex, transmit the data to the target server, and write it to the Remote Trail ./dirdat/px:

```
GGSCI (AMER) > Edit Param pfuncs
```

Your text editor opens the empty pfuncs.prm file. Add the following runtime parameters to the text file:

```
Extract pfuncs
RmtHost easthost, MgrPort 15001, Compress
RmtTrail ./dirdat/px
Passthru
Table WEST.*;
```

Save and close the file.

8. On the source server AMER_WEST_GGSCI tab, add the Extract group by executing the following commands:

```
GGSCI (AMER)> Add Extract pfuncs, ExtTrailsource ./dirdat/ex
EXTRACT added.

GGSCI (AMER)> Add RmtTrail ./dirdat/px, Extract pfuncs
RMTTRAIL added.

GGSCI (AMER) >
```

Setting Up the Second Oracle GoldenGate Extract Data Pump (pfunca)

9. On the source server AMER_WEST_GGSCI tab, configure the second Extract Data Pump to read from the local Extract Trail ./dirdat/ec, transmit the data to the target server, and write it to the Remote Trail ./dirdat/pc:

```
GGSCI (AMER) > Edit Param pfunca
```

The text editor of your choice opens the empty pfunca.prm file. Add the following runtime parameters to the text file:

```
Extract pfunca
RmtHost easthost, MgrPort 15001, Compress
RmtTrail ./dirdat/pc
Passthru
Table WEST.*;
```

10. On the source server AMER_WEST_GGSCI tab, add the Extract group by executing the following commands:

```
GGSCI (AMER)> Add Extract pfunca, ExtTrailsource ./dirdat/ec
GGSCI (AMER)> Add RmtTrail ./dirdat/pc, Extract pfunca
```

Creating the Oracle GoldenGate Checkpoint Table

On the target server, Replicat checkpoints to a special table created in the database. This checkpoint table ensures that the Replicat checkpoint is part of the transaction and is used to ensure data integrity. If it does not already exist, create the checkpoint table:

11. On the target server EURO_EAST_GGSCI tab, start GGSCI and execute the following startup commands. If you are still logged in to GGSCI from a previous lab, skip this task and just continue with the next task.

```
[OS_prompt gg_euro] rlggsci

GGSCI (EURO) 1> DBLogin UserIDAlias ggueuro
Successfully logged into database.

GGSCI (EURO) 2> Info CheckpointTable
No checkpoint table specified, using GLOBALS specification (GGS_CHECKPOINT)...
Checkpoint table GGS_CHECKPOINT does not exist.

GGSCI (EURO) 3> Add CheckpointTable
No checkpoint table specified, using GLOBALS specification (GGS_CHECKPOINT)...
Successfully created checkpoint table GGS_CHECKPOINT.

GGSCI (EURO) 4> Start mgr
```

Setting Up Oracle GoldenGate Delivery (Two Replicats: rfunca and rfunc)

12. On the target server EURO_EAST_GGSCI tab, configure the Replicat Delivery components:

    ```
    GGSCI (EURO) > Edit Param rfunca
    ```

    The editor of your choice opens the empty `rfunca.prm` file. Add the following runtime parameters to the text file:

    ```
    NoList
    Include ./dirmac/macrolib.mac
    List
    Replicat rfunca
    #dbconnect ()
    SourceDefs ./dirdef/section13.defs
    DiscardFile ./dirrpt/rfunca.dsc, Purge
    #bpsettings ()
    #funcsmap (WEST.WSHOP_FUNCS, EAST.WSHOP_FUNCS)
    Map WEST.CUST_ZIP, TARGET EAST.CUST_CITY_STATE,
    SQLEXEC (ID ZIPLKUP,
             QUERY ' SELECT zip_city, zip_state FROM east.zip_lookup WHERE
                      zip = :vzip ',
             PARAMS (vzip = cust_zip)),
    ColMap (usedefaults,
            cust_city = @GETVAL (ZIPLKUP.zip_city),
            cust_state = @GETVAL (ZIPLKUP.zip_state)
    );
    ```

    Note the QUERY has 'single, straight' quotes. Save and close the file.

13. On the EURO_EAST_GGSCI tab, execute the command:

    ```
    GGSCI (EURO) > Edit Param rfunc
    ```

    The text editor of your choice opens the empty `rfunc.prm` file. Add the following runtime parameters to the text file:

    ```
    Replicat rfunc
    UserIDAlias ggueuro
    AssumeTargetDefs
    DiscardFile ./dirrpt/rfunc.dsc, Purge
    DecryptTrail AES256 KeyName MyKey3
    Map WEST.*, Target EAST.*;
    ```

    Save and close the file.

14. On the EURO_EAST_GGSCI tab, add the Replicats by executing the following commands:

    ```
    GGSCI (EURO)> Add Replicat rfunc,  ExtTrail ./dirdat/px
    GGSCI (EURO)> Add Replicat rfunca, ExtTrail ./dirdat/pc
    ```

15. Ensure on both systems that the manager is up and running (if not, start the Manager):

    ```
    GGSCI (both)> Info mgr
    GGSCI (both)> Start mgr
    ```

16. In the GG_AMER_HOME window, start all the extracts:

```
GGSCI (AMER)> Start ER ef*

Sending START request to MANAGER ...
EXTRACT EFUNCS starting

GGSCI (AMER)> Start ER pf*

Sending START request to MANAGER ...
EXTRACT PFUNCA starting

Sending START request to MANAGER ...
EXTRACT PFUNCS starting

GGSCI (AMER)>
```

Verify that all process from this practice are running by executing the `Info ER *` command. You do not care if other processes from previous labs (such as `erange*`) are stopped.

17. In the GG_EURO_HOME window, start all the Replicats:

```
GGSCI (EURO) > Start ER rf*

Sending START request to MANAGER ...
REPLICAT RFUNC starting

Sending START request to MANAGER ...
REPLICAT RFUNCA starting

GGSCI (EURO) >
```

Verify that all process are running by executing the `Info ER *` command. You do not care if other processes from previous labs (such as `rrange*`) are stopped.

Note: The Replicats may already be running if the `mgr.prm` was not deleted in an earlier lab.

This completes Practice 13-2. Continue now with Practice 13-3.

Practice 13-3: Generating Transactions and Validating Results

Tasks

1. On the AMER_WEST_SQLPLUS tab, at the SQL prompt, generate source database transactions by executing the following command:

```
[OS_prompt labs] sqlplus west@amer/oracle_4U
                 @Section13/sqlscripts/trans_generator.sql
```

2. Verify that the data applied to the target wshop_encrypt table from the encrypted Oracle GoldenGate Trails is the same as the source data. On the AMER_WEST_SQLPLUS tab, at the shell prompt, start SQL*Plus and query all rows from the wshop_encrypt table:

```
[OS_prompt labs] rlsqlplus west@amer/oracle_4U

SQL*Plus: Release 12.1.0.1.0 Production on Mon Dec 9 12:30:54 2013
Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> set sqlprompt AMER_SQL>
AMER_SQL> SELECT * FROM wshop_encrypt;

ROW_NUMBER ROW_TEXT
---------- -------------------------
         1 AWUpbEBePRhIKhRuKBJsrweLW
         2 yoHddMMKXlGpQLuKSDTRqSSoX
         3 ziJvSXUJxEiNqapSPAxdoVUNL
         4 JnweLVnqmYfjfhjLXeCUAlTTP
         5 UGoIkLswswiqxoqHtaFXBaYcf
         6 JVFgfYHDSUFyjjQbSWeXBfXca
         7 XntsrAQJOQqREvPDwvXZYVsbB
         8 ZcDzHjttaYQHUciWpAnMqVSav
         9 bDsUBZNWwrHADHqOhMQrlSlXN
        10 lLNtcVEHsclSGwUHPUfqNEhAd

10 rows selected.

AMER_SQL>
```

Your text will be different, but you should have 10 rows of 25 encrypted characters each.

3. On the EURO_EAST_SQLPLUS tab, at the shell prompt, start `rlsqlplus` and query the `wshop_encrypt` table:

```
[OS_prompt labs] rlsqlplus east@euro/oracle_4U

SQL*Plus: Release 12.1.0.1.0 Production on Mon Dec 9 12:30:54 2013
Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> set sqlprompt EURO_SQL>
EURO_SQL> SELECT * FROM wshop_encrypt;

ROW_NUMBER ROW_TEXT
---------- ------------------------
         1 AWUpbEBePRhIKhRuKBJsrweLW
         2 yoHddMMKXlGpQLuKSDTRqSSoX
         3 ziJvSXUJxEiNqapSPAxdoVUNL
         4 JnweLVnqmYfjfhjLXeCUAlTTP
         5 UGoIkLswswiqxoqHtaFXBaYcf
         6 JVFgfYHDSUFyjjQbSWeXBfXca
         7 XntsrAQJOQqREvPDwvXZYVsbB
         8 ZcDzHjttaYQHUciWpAnMqVSav
         9 bDsUBZNWwrHADHqOhMQrlSlXN
        10 lLNtcVEHsclSGwUHPUfqNEhAd

10 rows selected.

EURO_SQL>
```

Each table should contain 10 rows of encrypted equivalent data.

Validating the Token, Macro, and SQLEXEC Information

4. In the GG_AMER_HOME terminal window, view the `efuncs.prm` parameter file. Four user-defined tokens will be added to each record captured for this table.
   - `TKN-EXTLAG-MSEC` uses the @GETENV function (all GoldenGate function names begin with the @ character) to get the Extract database lag from the GoldenGate environment.
   - `TKN-SRC-DBNAME` gets the source database name from the database environment.
   - `TKN-SRC-DBVERSION` gets the source database version from the database environment.
   - `TKN-TXN-CSN` gets the commit sequence number for the transaction from the database.

5. On the EURO_EAST_GGSCI tab, view the rfunca.prm parameter file, especially the SQLEXEC query:

```
GGSCI (EURO)> View Param rfunca

NOLIST
INCLUDE ./dirmac/macrolib.mac
LIST
Replicat rfunca
#dbconnect ()
SourceDefs ./dirdef/section13.defs
DiscardFile ./dirrpt/rfunca.dsc, Purge
#bpsettings ()
#funcsmap (WEST.wshop_funcs, EAST.wshop_funcs)
Map WEST.CUST_ZIP, TARGET EAST.CUST_CITY_STATE,
SQLEXEC (ID ZIPLKUP,
         QUERY ' SELECT zip_city, zip_state FROM EAST.zip_lookup
                 WHERE zip = :vzip ',
         PARAMS (vzip = cust_zip) ),
ColMap (usedefaults,
        cust_city = @GETVAL (ZIPLKUP.zip_city),
        cust_state = @GETVAL (ZIPLKUP.zip_state)
       );

GGSCI (EURO)>
```

Note: The QUERY must have 'single, straight' quotes.

The Map statement for the wshop_funcs table uses the #funcsmap macro:

```
MACRO #funcsmap
PARAMS (#src_table, #target_table)
BEGIN
   Map #src_table, TARGET #target_table,
    ColMap (usedefaults,
            gg_commit_ts = @GETENV ( 'GGHEADER' , 'COMMITTIMESTAMP' ),
            lag_extract_ms = @TOKEN ( 'TKN-EXTLAG-MSEC' ),
            lag_replicat_ms = @GETENV ( 'LAG' , 'MSEC' ),
            src_db_name = @TOKEN ( 'TKN-SRC-DBNAME' ),
            src_db_version = @TOKEN ( 'TKN-SRC-DBVERSION' ),
            src_txn_csn = @TOKEN ( 'TKN-TXN-CSN' )
     );
END;
```

This macro uses two column conversion functions: @GETENV to get the commit timestamp from the GoldenGate header record and Replicat lag from the Oracle GoldenGate environment, and @TOKEN to get the tokens set by the source Extract. The @GETENV quotes must be 'single straight' quotes.

Also notice that for the rfunca parameters, the Map statement for EAST.CUST_ZIP uses the SQLEXEC option to query a lookup table.

For every WEST.CUST_ZIP record, the query will be executed using the incoming data for the cust_zip column in the WHERE clause. The returned data is mapped to the target zip_city and zip_state columns.

6. On the target database EURO_EAST_SQLPLUS tab, execute the queries from a SQL prompt. If necessary, restart SQL*Plus:

```
[OS_prompt labs] rlsqlplus east@euro/oracle_4U
```

a. Verify that the token data was applied to the target by executing the following query:

```
EURO_SQL> SELECT src_db_name, src_db_version, src_txn_csn
          FROM wshop_funcs;

SRC_DB_NAM SRC_DB_VERSION                                          SRC_TXN_CSN
---------- ------------------------------------------------------- -----------
AMER       Oracle Database 12c Enterprise Edition Release 12.1.0.1     2768888
AMER       Oracle Database 12c Enterprise Edition Release 12.1.0.1     2768890
AMER       Oracle Database 12c Enterprise Edition Release 12.1.0.1     2768892
AMER       Oracle Database 12c Enterprise Edition Release 12.1.0.1     2768894
AMER       Oracle Database 12c Enterprise Edition Release 12.1.0.1     2768896
AMER       Oracle Database 12c Enterprise Edition Release 12.1.0.1     2768898
AMER       Oracle Database 12c Enterprise Edition Release 12.1.0.1     2768900
AMER       Oracle Database 12c Enterprise Edition Release 12.1.0.1     2768878
AMER       Oracle Database 12c Enterprise Edition Release 12.1.0.1     2768882
AMER       Oracle Database 12c Enterprise Edition Release 12.1.0.1     2768886

10 rows selected.

SQL>
```

Your SCNs will be different.

b. Still in EURO_EAST_SQLPLUS, execute the following query:

```
EURO_SQL> SELECT lag_extract_ms, lag_replicat_ms
          FROM wshop_funcs;

LAG_EXTRACT_MS LAG_REPLICAT_MS
-------------- ---------------
          3369        7470
          3369        7472
          3369        8473
          3369        8473
          3369        8473
          3369        8473
          3369        8473
          3369        7470
          3369        7470
          3369        7470

10 rows selected.

EURO_SQL>
```

In the above example, Extract took 3.3 seconds to capture each record from Oracle redo, while Replicat lag time (Replicat current time – transaction commit timestamp) was around 8 seconds. Your lag values may vary.

c. To return the data that was recorded as the source record commit timestamp, execute the following query:

```
EURO_SQL> SELECT gg_commit_ts FROM wshop_funcs;

GG_COMMIT_TS
---------------------------------------------------------------------------
09-DEC-13 02.37.03.000396 PM
09-DEC-13 02.37.03.000396 PM
09-DEC-13 02.37.03.000396 PM
09-DEC-13 02.37.03.000396 PM
09-DEC-13 02.37.03.000396 PM
09-DEC-13 02.37.03.000396 PM
09-DEC-13 02.37.03.000396 PM
09-DEC-13 02.37.03.000396 PM
09-DEC-13 02.37.03.000396 PM
09-DEC-13 02.37.03.000396 PM

10 rows selected.

EURO_SQL>
```

d. Back on the source database AMER_WEST_SQLPLUS tab, execute the following query:

```
[OS_prompt labs] rlsqlplus west@amer/oracle_4U
```

e. Verify the ZIP codes:

```
AMER_SQL> SELECT * FROM cust_zip;

   CUST_ID   CUST_ZIP
---------- ----------
         1      80033
         2      70117
         3      94105
         4      80202
         5      70001

AMER_SQL>
```

f. Verify the SQLEXEC functionality by executing the following query on the target database EURO_EAST_SQLPLUS tab:

```
EURO_SQL> SELECT * FROM cust_city_state;

   CUST_ID CUST_CITY                CU   CUST_ZIP
---------- ------------------------- -- ----------
         1 Wheat Ridge               CO      80033
         2 New Orleans               LA      70117
         3 San Francisco             CA      94105
         4 Denver                    CO      80202
         5 Jefferson                 LA      70001

EURO_SQL>
```

7. Stop the processes for Practice 13-3 and clean up:
   a. On the source system AMER_WEST_GGSCI tab, enter the following commands:

```
GGSCI (AMER)> Stop ER *
GGSCI (AMER)> Delete ER * !
GGSCI (AMER)> sh rm dirdat/*
```

   b. On the target system EURO_EAST_GGSCI tab, enter the following commands:

```
GGSCI (EURO)> Stop ER *
GGSCI (EURO)> Delete ER * !
GGSCI (EURO)> sh rm dirdat/*
```

This completes Practice 13-3. Continue now with Practice 13-4.

Practice 13-4: Encryption Using Wallets

Overview

In previous practices, you used ENCKEYS to manually maintain encryption keys. In this practice, you will use the Wallet to maintain encryption keys. Wallet keys are the preferred method of specifying trail encryption.

Tasks

1. On the AMER_WEST_GGSCI tab, open the Wallet:

```
GGSCI (AMER)> Open Wallet
Opened wallet at location 'dirwlt'.
```

2. On the AMER_WEST_GGSCI tab, add a master key: (You need only one, but you can have more than one.)

```
GGSCI (AMER)> Add MasterKey
Master key 'OGG_DEFAULT_MASTERKEY' added to wallet at location 'dirwlt'.

GGSCI (AMER)> Info MasterKey
Masterkey Name:                 OGG_DEFAULT_MASTERKEY
Creation Date:                  Mon Dec  9 15:10:14 2013

Version:        Creation Date:                  Status:
1               Mon Dec  9 15:10:14 2013        Current
```

3. The Wallet cannot be explicitly closed, nor does it need to be. You can still copy it at this point. If this really bothers you, you can Exit GGSCI to completely release the Wallet. On the AMER_WEST_GGSCI tab, copy the Wallet single-sign-on file from source to target:

```
GGSCI (AMER)> sh cp dircrd/cwallet.sso $GG_EURO_HOME/dircrd
GGSCI (AMER)> sh cp dirwlt/cwallet.sso $GG_EURO_HOME/dirwlt
```

4. Verify that the master key made it to the target. On the EURO_EAST_GGSCI tab, execute the following commands:

```
GGSCI (EURO)> Open Wallet
GGSCI (EURO)> Info MasterKey
```

   The information should match on both the source and the target. If you had a shared disk, this extra step to copy would not be necessary.

5. You are going to use the original extwest, pwest, and reast from the earlier labs. The two files that need to be modified are the Extract extwest and data pump pwest; the replicat is okay as is.

   On the AMER_WEST_GGSCI tab, edit the extwest and make sure that the last Table line is a wildcard (as in Lab 6), not restricted (as in Lab 10).

```
GGSCI (AMER)> Edit Param extwest
```

```
Extract extwest
ExtTrail ./dirdat/ew
UserIDAlias gguamer
--Table WEST.ACCOUNT_TRANS, Where (ACCOUNT_TRANS_TYPE = "CR");
Table WEST.*;
```

6. On the AMER_WEST_GGSCI tab, edit pwest and add one line "EncryptTrail AES256":

```
GGSCI (AMER)> Edit Param pwest
```

```
Extract pwest
-- Add the following one line:
EncryptTrail AES256
RmtHost easthost, MgrPort 15001, Compress
RmtTrail ./dirdat/pe
Passthru
Table WEST.*;
```

The matching Decrypt in the Replicat is automatic; you do not need to specify it.

7. Start up the source processes. (These are the same as in earlier labs). On the AMER_WEST_GGSCI tab, enter the following commands:

```
GGSCI (AMER)> Add Extract extwest, TranLog, Begin Now
EXTRACT added.

GGSCI (AMER)> Add ExtTrail ./dirdat/ew, Extract extwest
EXTTRAIL added.

GGSCI (AMER)> Add Extract pwest, ExtTrailSource ./dirdat/ew
EXTRACT added.

GGSCI (AMER)> Add RmtTrail ./dirdat/pe, Extract pwest
RMTTRAIL added.

GGSCI (AMER)> Start *
Sending START request to MANAGER ...
EXTRACT EXTWEST starting

Sending START request to MANAGER ...
EXTRACT PWEST starting

GGSCI (AMER)> Info All
```

8. Start up the target processes. (These are the same as in the earlier labs.) On the EURO_EAST_GGSCI tab, enter the following commands:

```
GGSCI (EURO)> Add Replicat reast, ExtTrail ./dirdat/pe
REPLICAT added.

GGSCI (EURO)> Start *
Sending START request to MANAGER ...
REPLICAT REAST starting

GGSCI (EURO)> Info All
```

9. Insert a row on the source database. On the AMER_WEST_SQLPLUS tab, enter the following DML:

```
AMER_SQL> INSERT INTO west.branch VALUES ( 50, 80071 );
AMER_SQL> COMMIT;
```

10. Verify that the row was replicated to the target database. On the EURO_EAST_SQLPLUS tab, enter the following query:

```
EURO_SQL> SELECT * FROM east.branch WHERE branch_number > 45;
BRANCH_NUMBER BRANCH_ZIP
------------- ----------
           50      80071

EURO_SQL>
```

11. View the report for the pump. You should see the encryption type at the very end. On the AMER_WEST_GGSCI tab, enter the following command:

```
GGSCI (AMER)> View Report pwest
  ...many lines omitted for clarity...
2013-12-09 16:04:26  INFO    OGG-05519  Output trail file encryption: AES256.
```

    This was much simpler than the previous ENCKEYS method. The Wallet method is the preferred method *if available*. The only reason that the previous method was shown first is that not all Oracle GoldenGate platforms support the Wallet.

12. Clean up the Oracle GoldenGate environment by executing the following GGSCI commands on both the source and target servers in the OGG HOME directories:

```
GGSCI (both)> Stop ER *
GGSCI (both)> DBLogin UserID gguser, Password oracle_4U
GGSCI (both)> Delete ER *                    (confirm the delete)
GGSCI (EURO)> Delete CheckpointTable         (only for the EURO db)
GGSCI (both)> Stop mgr !
GGSCI (both)> Exit
[OS_PROMPT] rm ./dirdat/*
[OS_PROMPT] rm ./dirrpt/*
```

    Do not type the *(red comments)*. If you wanted to automate this cleanup, then stop both GGSCI sessions and run ~/labs/setup/kill-all-ogg.sh from the fifth oracle@hostname tab.

This completes Practice 13-4. This completes the practices for Lesson 13. Stop here.

Practices for Lesson 14: Bidirectional Replication

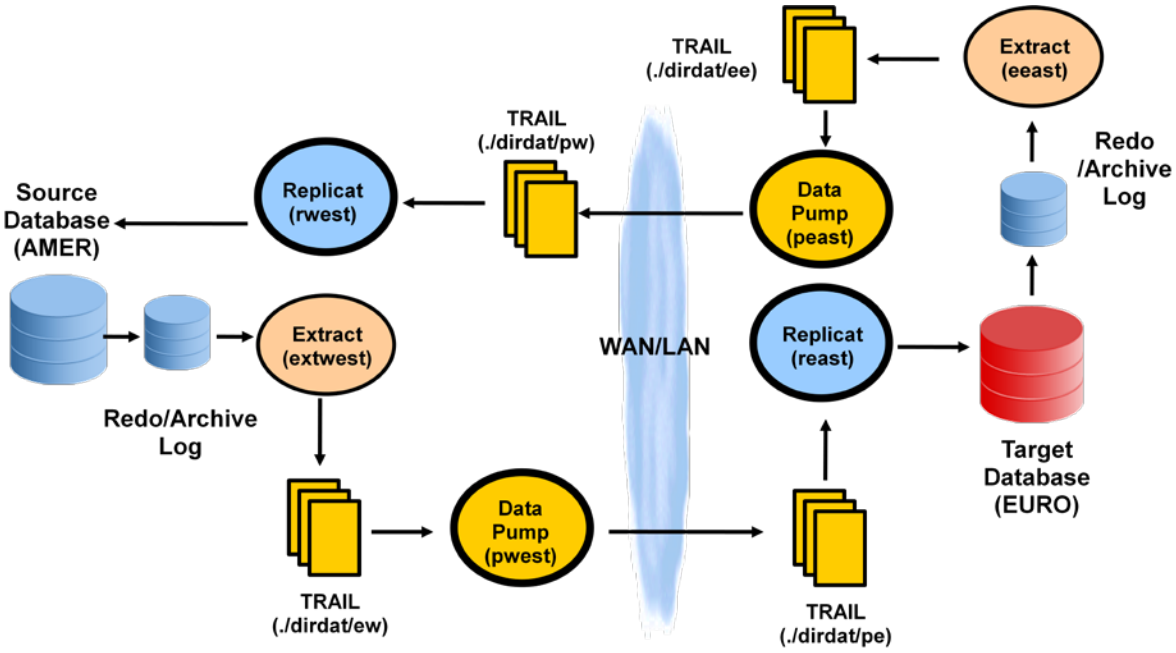
Chapter 14

Practices for Lesson 14: Overview

Practices Overview

In this practice, you configure Oracle GoldenGate to perform real-time data capture and apply on two active databases with data segregated by location.

Big Picture



In the preceding diagram, the AMER and EURO databases are exact copies of one another. Your application is load-balanced across the two data centers so that each data center services 50% of all active accounts. Each data center is also the backup to the other, so the two databases must always be in sync. This is accomplished by configuring Oracle GoldenGate in Active-Active mode, with Change Data Capture Extracts reading from each database's Redo and Archive logs, pumping the captured data over TCP/IP to a staging queue (Oracle GoldenGate Remote Trail), and applying via Replicat. Notice the potential for a loop.

Practice 14-1: Two-Way Active-Active Data Replication

Overview

In this practice, you will create the two databases (AMER and EURO) and seed them both with identical data, and then you will ensure that database logging is enabled.

In a two-way active-active (bidirectional) replication environment, you must *avoid* data looping, where data changes replicated from system A to system B are replicated back to system A. There are several loop detection and avoidance methods. You will configure each Extract process to ignore transactions originated by its local Replicat user. This is done by setting the `TranLogOptions ExcludeUser` parameter in the Extract parameter file.

Note: If at any time or for any reason you exit GGSCI, make sure you re-`DBLogin` when you restart GGSCI. You can do this easily by entering:

```
GGSCI (BOTH)> Obey /home/oracle/labs/setup/startup.oby
```

Tasks

1. Note that AMER is no longer the only source database and EURO is no longer the only target database. Pay particular attention then to which tab you are entering the commands into. On the AMER_WEST_SQLPLUS tab, navigate to the `Section14/Practice14-1/sqlscripts` directory to run the script. To create the AMER database, execute the following commands:

```
[OS_prompt labs] rlsqlplus west@amer/oracle_4U

SQL> set sqlprompt AMER_SQL>
AMER_SQL> set echo on
AMER_SQL> @Section14/Practice14-1/sqlscripts/database.sql
AMER_SQL> /*
AMER_SQL>                 Script to create the source database tables
AMER_SQL>   This will simulate a banking database used with an OLTP
application
... Many lines omitted for clarity ...
```

   Scroll down the terminal window to see what objects were created, and what SQL statements were executed.

2. To seed the West AMER source database, on the AMER_WEST_SQLPLUS tab, execute the following commands:

```
AMER_SQL> @Section14/Practice14-1/sqlscripts/seed_database.sql
AMER_SQL> /*
AMER_SQL>      Seed the branch and teller lookup tables with these values
AMER_SQL>        Number of accounts: 2000
AMER_SQL>        Starting account balance: Random up to $10000
AMER_SQL>        Number of branches: 40
AMER_SQL>        Tellers per branch: 20
AMER_SQL> */
AMER_SQL>
AMER_SQL> TRUNCATE TABLE ACCOUNT;
Table truncated.

... Many lines omitted for clarity ...
```

Scroll down the terminal window to see what objects were created, and what SQL statements were executed.

3. On the EURO_EAST_SQLPLUS tab, navigate to the same Section14/Practice14-1/sqlscripts directory to run the same scripts. To create the EURO database, execute the following commands:

```
[OS_prompt labs] rlsqlplus east@euro/oracle_4U

SQL> set sqlprompt EURO_SQL>
EURO_SQL> set echo on
EURO_SQL> @Section14/Practice14-1/sqlscripts/database.sql
EURO_SQL> /*
EURO_SQL>                Script to create the source database tables
EURO_SQL>   This will simulate a banking database used with an OLTP
application
... Many lines omitted for clarity ...
```

Scroll down the terminal window to see that the same objects were created for schema east as for west, and the same SQL statements were executed.

4. To seed the East EURO source database, on the EURO_EAST_SQLPLUS tab, execute the following commands:

```
EURO_SQL> @Section14/Practice14-1/sqlscripts/seed_database.sql
EURO_SQL> /*
EURO_SQL>      Seed the branch and teller lookup tables with these values
EURO_SQL>        Number of accounts: 2000
EURO_SQL>        Starting account balance: Random up to $10000
EURO_SQL>        Number of branches: 40
EURO_SQL>        Tellers per branch: 20
EURO_SQL> */
EURO_SQL>
EURO_SQL> TRUNCATE TABLE ACCOUNT;
Table truncated.

... Many lines omitted for clarity ...
```

Scroll down the terminal window to see what objects were created, and what SQL statements were executed.

Creating the Replicat Checkpoint Table on AMER

Replicat checkpoints to a special table created in the database. This checkpoint table ensures that the Replicat checkpoint is part of the transaction and is used to ensure data integrity. You already had configured one for EURO (but it was deleted as part of the last cleanup).

5. Make sure GGSCI is *not* running in AMER. To create the GLOBALS parameter file, on the AMER_WEST_GGSCI tab, execute the following command:

```
[OS_prompt gg_amer] vi GLOBALS
```

You can use gedit or any text editor instead of vi. Input the following parameter line:

```
CheckpointTable GGS_CHECKPOINT
```

Note: The file name must be all uppercase with no extension. Save and close the file.

6. To create the checkpoint table on the AMER database server, on the AMER_WEST_GGSCI tab, start GGSCI and execute the following commands:

```
[OS_prompt gg_amer] more GLOBALS
CheckpointTable GGS_CHECKPOINT

[OS_prompt gg_amer] rlggsci
Oracle GoldenGate Command Interpreter for Oracle

GGSCI (AMER) 1> DBLogin UserIDAlias gguamer
Successfully logged into database.

GGSCI (AMER) 2> Add CheckpointTable
No checkpoint table specified, using GLOBALS specification (GGS_CHECKPOINT)...
Successfully created checkpoint table GGS_CHECKPOINT.

GGSCI (AMER) 3> set editor gedit          ← This is optional…
```

Note: If the checkpoint table already exists, you will receive an error message. If so, issue the `Delete CheckpointTable` command, confirm the deletion, and then try adding the checkpoint table again.

Creating the Replicat Checkpoint Table on EURO

7. To create the checkpoint table on the EURO database server, start GGSCI and execute the following commands:

```
[OS_prompt gg_euro] rlggsci

Oracle GoldenGate Command Interpreter for Oracle

GGSCI (EURO) 1> DBLogin UserIDAlias ggueuro
Successfully logged into database.

GGSCI (EURO) 2> Add CheckpointTable
No checkpoint table specified, using GLOBALS specification (GGS_CHECKPOINT)...
Successfully created checkpoint table GGS_CHECKPOINT.

GGSCI (EURO) 3> set editor gedit          ← This is optional…
```

Note: If the checkpoint table already exists, you will receive an error message. If so, issue the `Delete CheckpointTable` command, confirm the deletion, and then try adding the checkpoint table again.

Getting the Database to Log Keys

As you have seen in previous practices, the Oracle database must be set to log the table key values whenever it logs a row change, so that they are available to Oracle GoldenGate in Redo.

8. On the AMER_WEST_GGSCI database server tab, execute the following commands:

   ```
   GGSCI (AMER) > Add TranData WEST.*
   ```

   Many of these were already enabled. Had you originally done SchemaTranData, the added tables would have automatically been picked up. Validate table-level supplemental logging is on by executing the following command in the GG_AMER_HOME terminal window:

   ```
   GGSCI (AMER) > Info TranData WEST.*
   ```

9. On the EURO_EAST_GGSCI database server tab, execute the following commands:

   ```
   GGSCI (EURO) > Add TranData EAST.*
   ```

   All of these were new (as compared to AMER's TranData). Validate that table-level supplemental logging is on by executing the following command in the GG_EURO_HOME terminal window:

   ```
   GGSCI (EURO) > Info TranData EAST.*
   ```

   Dropping a table and re-creating it will cause you to need to re-add the TranData, because TranData does not persist across a DDL TABLE DROP/CREATE. However, SchemaTranData does effectively persist (well, actually it gets re-added) across a TABLE DROP/CREATE.

Creating the Oracle GoldenGate Components on AMER

In this practice, you will configure the Manager, Extract, Data Pump Extract, and Replicat on the AMER instance.

From the previous practices, you should still have a number of parameter files already created for the AMER database. For this practice, you will be using:

- mgr.prm
- extwest.prm
- pwest.prm

You will modify those three parameter files for this practice and create a number of new entries for both the AMER and EURO databases.

Configuring Change Data Capture, AMER to EURO

10. On the AMER_WEST_GGSCI tab, execute the following command:

    ```
    GGSCI (AMER) > Edit Param mgr
    ```

    Verify that it includes the following:

    ```
    Port 15000
    DynamicPortList 15100-15150
    PurgeOldExtracts ./dirdat/*, UseCheckpoints
    ```

11. Start the Manager and verify that it is running. Note there is no harm in trying to start it twice:

```
GGSCI (AMER)> Start mgr
Manager started.

GGSCI (AMER)> Start mgr
MGR is already running.

GGSCI (AMER)> Info mgr
Manager is running (IP port westhost.15000, Process ID 32173).

GGSCI (AMER)>
```

12. On the AMER_WEST_GGSCI tab, modify the extract file by executing the following command:

    GGSCI (AMER) > **Edit Param extwest**

    Add the following parameter on an empty line just *before* the first Table statement:

    TranLogOptions ExcludeUser gguser

    Edit and verify that the extwest.prm file looks similar to the following:

```
Extract extwest
ExtTrail ./dirdat/ew
UserIDAlias gguamer
StatOptions ResetReportStats
Report At 00:01
ReportRollover At 00:01
ReportCount Every 1000 Records
ReportCount Every 60 Seconds, Rate
TranLogOptions ExcludeUser gguser
Table WEST.*;
```

    Note: Your file may look slightly different depending on how you completed Lab 10-3. If extwest has the line that says:

    Table WEST.ACCOUNT_TRANS, Where (ACCOUNT_TRANS_TYPE = "CR");

    then remove that one line, or comment it out, and replace it with:

    Table WEST.*;

    because the Where () clause will cause conflicts here.

    Do not forget the last trailing semi-colon. Save and close the file.

13. Add the Extract group by executing the following GGSCI commands:

    - Add Extract extwest, Tranlog, Begin Now
      This adds the extwest Extract to Oracle GoldenGate specifying that it will read from Oracle DB Redo/Archive logs, with an initial Redo checkpoint of the current timestamp.

- `Add ExtTrail ./dirdat/ew, Extract extwest, Megabytes 50`
  This adds the `dirdat/ew` Local Extract Trail, and links the trail to the `extwest` Extract.

```
GGSCI (AMER)> Add Extract extwest, TranLog, Begin Now
EXTRACT added.

GGSCI (AMER)> Add ExtTrail ./dirdat/ew, Extract extwest
EXTTRAIL added.

GGSCI (AMER)>
```

14. On the `AMER_WEST_GGSCI` tab, configure the Extract Data Pump to read from the `dirdat/ew` ("Extract West") local Extract Trail, transmit the data to the `EURO` server, and write it to the `./dirdat/pe` ("Pump East") Remote Trail:

    Execute the following command:

    ```
    GGSCI (AMER)> Edit Param pwest
    ```

    Verify that the file includes the following: (`EncryptTrail` is optional.)

```
Extract pwest
EncryptTrail AES256                    ← This is optional…
RmtHost easthost, MgrPort 15001, Compress
RmtTrail ./dirdat/pe
Passthru
Table WEST.*;
```

    Save and close the file.

15. Add the Extract group by executing the following commands:

```
GGSCI (AMER)> Add Extract pwest, ExtTrailSource ./dirdat/ew
EXTRACT added.

GGSCI (AMER)> Add RmtTrail ./dirdat/pe, Extract pwest
RMTTRAIL added.

GGSCI (AMER)>
```

16. To set up Oracle GoldenGate Delivery on the `AMER` database, you must configure the Replicat component. On the `AMER_WEST_GGSCI` tab, execute the following command:

    ```
    GGSCI (AMER)> Edit Param rwest
    ```

    The text editor of your choice opens the empty `rwest.prm` file. Add the following runtime parameters to the text file:

```
Replicat rwest
AssumeTargetDefs
DiscardFile ./dirrpt/rwest.dsc, Purge
UserIDAlias gguamer
Map EAST.*, Target WEST.*;
```

    Save and close the file.

17. Add the Replicat by executing the following GGSCI command:

```
GGSCI (AMER)> Add Replicat rwest, ExtTrail ./dirdat/pw
REPLICAT added.

GGSCI (AMER)>
```

Creating the Oracle GoldenGate Components on EURO

Now you will configure the Manager, Extract, Data Pump Extract, and Replicat on the EURO instance.

From the previous practices, you should still have a number of parameter files already created for the EURO database. For this practice, you will be using:

- `mgr.prm`
- `reast.prm`

You will modify these parameter files in this practice and create a number of new entries and a new parameter file for both the AMER and EURO databases.

Configuring the Change Data Capture, EURO to AMER

18. On the EURO_EAST_GGSCI tab, execute the following:

```
GGSCI (EURO)> Edit Param mgr
```

Verify that it includes at least the following:

```
Port 15001
PurgeOldExtracts ./dirdat/*, UseCheckpoints
```

19. Start the Manager and verify that it is running:

```
GGSCI (EURO)> Start mgr
GGSCI (EURO)> Info manager
```

20. On the EURO_EAST_GGSCI tab, create the extract file by executing the following:

```
GGSCI (EURO)> Edit Param eeast
```

The text editor of your choice opens the empty `eeast.prm` file. Add the following runtime parameters to the text file:

```
Extract eeast
ExtTrail ./dirdat/ee
UserIDAlias ggueuro
TranLogOptions ExcludeUser GGUSER
Table EAST.*;
```

Save and close the file.

21. On the EURO_EAST_GGSCI tab, add the Extract group by executing the following commands:

- `Add Extract eeast, Tranlog, Begin Now`
  This adds the `eeast` Extract to GoldenGate, specifying that it will read from Oracle Redo/Archive with an initial Redo checkpoint of the current timestamp.

- `Add ExtTrail ./dirdat/ee, Extract eeast`
  This adds the `dirdat/ee` local Extract Trail, and links the trail to the Extract `eeast`.

```
GGSCI (EURO)> Add Extract eeast, TranLog, Begin Now
EXTRACT added.

GGSCI (EURO)> Add ExtTrail ./dirdat/ee, Extract eeast
EXTTRAIL added.

GGSCI (EURO)>
```

22. On the `EURO_EAST_GGSCI` tab, configure the Extract Data Pump to read from the `dirdat/ee` local Extract Trail, transmit the data to the `AMER` server, and write it to the `dirdat/pw` Remote Trail:

    From a GGSCI prompt, execute the following command:

    `GGSCI(EURO)> Edit Param peast`

    Your text editor of choice opens the empty `peast.prm` file. Add the following runtime parameters to the text file:

```
Extract peast
RmtHost westhost, MgrPort 15000, Compress
RmtTrail ./dirdat/pw
Passthru
Table EAST.*;
```

    Save and close the file.

23. On the `EURO_EAST_GGSCI` tab, add the Extract group by executing the following commands:

```
GGSCI (EURO)> Add Extract peast, ExtTrailSource ./dirdat/ee
EXTRACT added.

GGSCI (EURO)> Add RmtTrail ./dirdat/pw, Extract peast
RMTTRAIL added.

GGSCI (EURO) >
```

24. To set up Oracle GoldenGate Delivery on the `EURO` database, you must modify the Replicat parameter file. On the `EURO_EAST_GGSCI` tab, execute the following command:

    `GGSCI (EURO)> Edit Param reast`

    Verify that it includes the following:

```
Replicat reast
AssumeTargetDefs
DiscardFile ./dirrpt/reast.dsc, Purge
UserIDAlias ggueuro
Map WEST.*, Target EAST.*;
```

    Save and close the file.

25. On the EURO_EAST_GGSCI tab, add the Replicat by executing the following command:

```
GGSCI (EURO)> Add Replicat reast, ExtTrail ./dirdat/pe
REPLICAT added.

GGSCI (EURO) >
```

Validating the Results

26. On both the AMER_WEST_GGSCI tab and the EURO_EAST_GGSCI tab, execute the following commands to start all of the configured Extracts and Replicats:

    For AMER, the processes should all end in WEST:

```
GGSCI (AMER)> Start ER *

Sending START request to MANAGER ...
EXTRACT EXTWEST starting

Sending START request to MANAGER ...
EXTRACT PWEST starting

Sending START request to MANAGER ...
REPLICAT RWEST starting

GGSCI (AMER)>
```

    For EURO, the processes should all end in EAST:

```
GGSCI (EURO)> Start ER *

Sending START request to MANAGER ...
EXTRACT EEAST starting

Sending START request to MANAGER ...
EXTRACT PEAST starting

Sending START request to MANAGER ...
REPLICAT REAST starting

GGSCI (EURO)>
```

27. Execute the following GGSCI commands to make sure that all of the components are in the RUNNING state:

On the AMER_WEST_GGSCI tab:

```
GGSCI (AMER)> Info ER *

EXTRACT    EXTWEST   Last Started 2013-12-10 14:24   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:07 ago)
Process ID           425
Log Read Checkpoint  Oracle Redo Logs
                     2013-12-10 14:25:14  Seqno 301, RBA 25918464
                     SCN 0.6226792 (6226792)

EXTRACT    PWEST     Last Started 2013-12-10 14:24   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:02 ago)
Process ID           426
Log Read Checkpoint  File ./dirdat/ew000000
                     First Record  RBA 1417

REPLICAT   RWEST     Last Started 2013-12-10 14:24   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:05 ago)
Process ID           427
Log Read Checkpoint  File ./dirdat/pw000000
                     First Record  RBA 0

GGSCI (AMER) >
```

Write down the Log Read Checkpoint RBA numbers you see on your screen:

Answer:_____

On the EURO_EAST_GGSCI tab:

```
GGSCI (EURO) > Info ER *

EXTRACT    EEAST     Last Started 2013-12-10 14:24   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:08 ago)
Process ID           460
Log Read Checkpoint  Oracle Redo Logs
                     2013-12-10 14:28:30  Seqno 234, RBA 372736
                     SCN 0.4194269 (4194269)

EXTRACT    PEAST     Last Started 2013-12-10 14:24   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:01 ago)
Process ID           461
Log Read Checkpoint  File ./dirdat/ee000000
                     First Record  RBA 1413

REPLICAT   REAST     Last Started 2013-12-10 14:24   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:06 ago)
Process ID           462
Log Read Checkpoint  File ./dirdat/pe000000
                     First Record  RBA 0

GGSCI (EURO) >
```

Write down the Log Read Checkpoint RBA numbers you see on your screen:

Answer:_____

If any Oracle GoldenGate groups show a state of STOPPED or ABEND, view the group's report file, fix the error condition, and restart the group.

28. Start database activity by running the transaction generators simultaneously in the shells.

    On the AMER_WEST_SQLPLUS tab, generate transactions for the AMER database by executing the following script:

    ```
    AMER_SQL> @Section14/Practice14-1/sqlscripts/trans_generator_west.sql
    ```

    On the EURO_EAST_SQLPLUS tab, generate transactions for the EURO database by executing the following script:

    ```
    EURO_SQL> @Section14/Practice14-1/sqlscripts/trans_generator_east.sql
    ```

29. After a minute (West takes longer than East), the transaction generator scripts complete. Then execute the GGSCI command on each system:

    ```
    GGSCI (BOTH)> Info ER *
    ```

    This will display information for the two Replicats. When the Log Read Checkpoint RBA value returned stops (slows) incrementing for the Replicats, all the captured data has been applied.

    Compare the RBA values with those you wrote down earlier, after you started the Replicats and Extracts. The RBA values should now be much higher.

30. Execute the command on the AMER_WEST_GGSCI tab to get operation counts per table for the change data capture Extract:

    ```
    GGSCI (AMER)> Stats extwest, Total
    ```

    On the EURO_EAST_GGSCI tab, execute the same command for the Replicat.

    ```
    GGSCI (EURO)> Stats reast, Total
    ```

    The counts for the Extract (on AMER) and Replicat (on EURO) numbers should match.

31. Do the same for the EEAST Extract and the RWEST Replicat. The transaction counts for these components should match as well.

    Execute the command on the EURO_EAST_GGSCI tab to get operation counts per table for the change data capture Extract:

    ```
    GGSCI (EURO)> Stats eeast, Total
    ```

    On the AMER_WEST_GGSCI tab, execute the same command for the Replicat.

    ```
    GGSCI (AMER)> Stats rwest, Total
    ```

    The counts for the Extract (on EURO) and Replicat (on AMER) numbers should match.

    Because `TranLogOptions ExcludeUser GGUSER` was specified in the Extract parameter files on both AMER and EURO, it ensured that changes made by the local Replicat user will not be extracted along with changes made by other users on that site.

32. Clean up the Oracle GoldenGate environment by executing the following GGSCI command. This one script will reset both the source and target servers:

    a. Exit both GGSCI prompts. This is necessary for the kill shell script.

    b. On the fifth tab oracle@hostname, at the OS prompt, run:

    ```
    [OS_prompt] ~/labs/setup/kill-all-ogg.sh
    ```

This completes Practice 14-1. Continue now with Practice 14-2.

Practice 14-2: Conflict Detection and Resolution

Overview

Because Oracle GoldenGate is an asynchronous solution, in a dual active environment, conflicts can occur when modifications are made to identical sets of data on separate systems at the same time.

Conflicts occur when the timing of simultaneous changes results in one of the following out-of-sync conditions:

- A replicated insert attempts to add a row that already exists in the target.
- The Before image of a replicated update does not match the current row in the target.
- A replicated delete attempts to remove a row that does not exist in the target.

For example, UserA on DatabaseA updates a row, and UserB on DatabaseB updates the same row. If UserB's transaction occurs before UserA's transaction is synchronized to DatabaseB, there will be a *conflict* on the replicated transaction.

In this practice, you will set up a very simple Oracle GoldenGate configuration to demonstrate net change value. Conflict detection and conflict resolution will be implemented to maintain accurate inventory counts when the application uses update statements to decrement inventory. The premise for this demonstration is an online retail environment where:

1. The warehouse is stocked with 20 widgets. If the quantity on hand goes to 0, the item is tagged as out of stock in our online ordering system.
2. Customer A places an order for 15 widgets that is routed through the AMER (also known as West) database.
   - The SQL statement executed is:
     ```
     UPDATE products SET qty_in_stock = qty_in_stock + 10
     WHERE products_name = 'widget' AND qty_in_stock = 20;
     ```
3. Customer B simultaneously places an order for 4 widgets that is routed through the EURO (also known as East) database.
   - The SQL statement executed is:
     ```
     UPDATE products SET qty_in_stock = qty_in_stock + 4
     WHERE products_name = 'widget' AND qty_in_stock = 20;
     ```

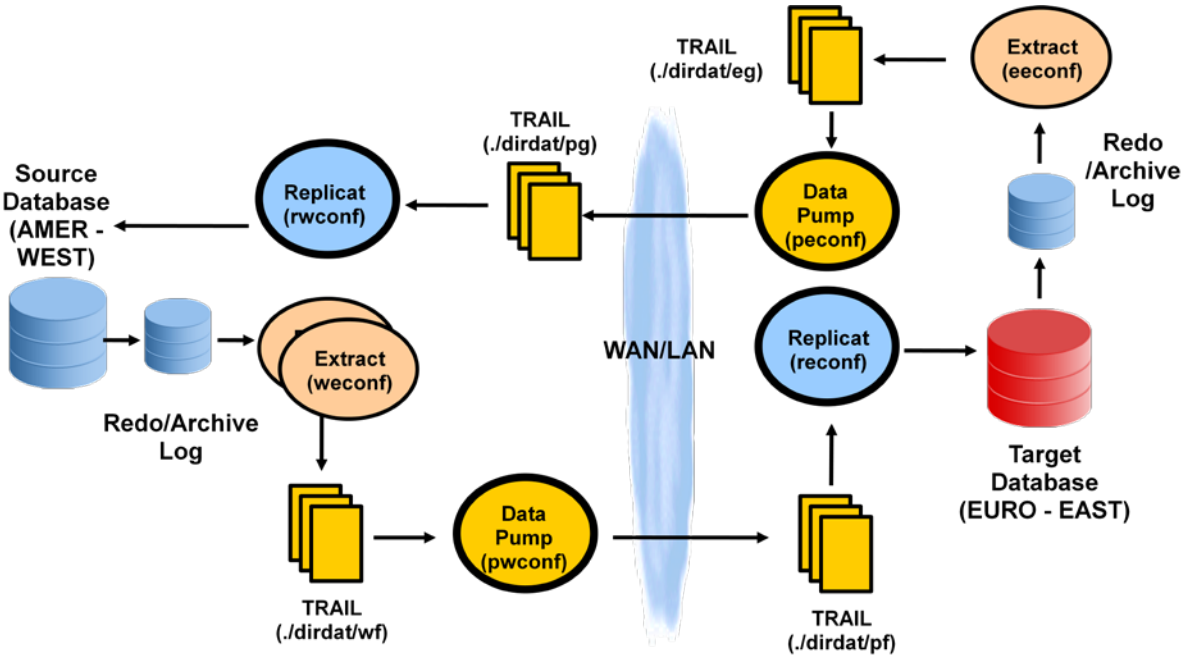
   In this example, the number of widgets in stock according to the AMER database is 30, and the number in stock according to the EURO database is 24.

   With conflict detection enabled, when Oracle GoldenGate attempts to update the EURO (East) database with the transaction from customer A, routed through AMER (West), the transaction will fail because the before image (from AMER) of `qty_in_stock` is 20, which does not match the current column value of 24 in EURO. Conversely, the transaction replicated to the AMER (West) database will also fail because the before image (from EURO) of `qty_in_stock`, 20, does not match the current column value of 30 in AMER.

   To resolve the conflict, you will add code in the Replicat Delivery configuration to add the sum total of the orders from both sites, so that the resulting `qty_in_stock` values after the replicated transactions are applied will be the same on both sites. The result is 34.

Big Picture

The following flowchart shows the components and relationships that you will create:



Tasks

Creating the Table and Data in the Database

1. On the AMER_WEST_SQLPLUS tab, create the table in the AMER database by executing the following commands. If you are still connected as `west@amer`, then just run the `.sql` script:

```
[OS_prompt labs] rlsqlplus west@amer/oracle_4U
SQL> set sqlprompt AMER_SQL>
AMER_SQL> set echo on
AMER_SQL> @Section14/Practice14-2/sqlscripts/database_west.sql
```

   Scroll down the terminal window to see what objects were created, and what SQL statements were executed.

2. On the EURO_EAST_SQLPLUS tab, create the identical table in the EURO database by executing the following commands. If you are still connected as `east@euro`, then just run the `.sql` script:

```
[OS_prompt labs] rlsqlplus east@euro/oracle_4U
SQL> set sqlprompt EURO_SQL>
EURO_SQL> set echo on
EURO_SQL> @Section14/Practice14-2/sqlscripts/database_east.sql
```

   So far the tables and data are identical for schemas east and west. Leave `sqlplus` running.

Getting the Database to Log Keys and Columns

3. On the AMER_WEST_GGSCI tab, execute the following commands to turn on supplemental logging. The Cols option adds more than the primary key to the Before image. Enter each command all on one line, with no line breaks:

```
[OS_prompt gg_amer] rlggsci
Oracle GoldenGate Command Interpreter for Oracle

GGSCI (AMER) 1> DBLogin UserIDAlias gguamer
Successfully logged into database.

GGSCI (AMER) 2> Add TranData WEST.PRODUCTS, Cols (product_name,
   vendor, qty_in_stock, modified_ts)

Logging of supplemental redo data enabled for table WEST.PRODUCTS.
TRANDATA for scheduling columns has been added on table 'WEST.PRODUCTS'.

GGSCI (AMER) 3> Info TranData WEST.PRODUCTS
Logging of supplemental redo log data is enabled for table WEST.PRODUCTS.
Columns supplementally logged for table WEST.PRODUCTS: PRODUCT_NUMBER,
PRODUCT_NAME, VENDOR, QTY_IN_STOCK, MODIFIED_TS.
```

4. On the EURO_EAST_GGSCI tab, execute the following commands to enable supplemental logging:

```
[OS_prompt gg_euro] rlggsci
Oracle GoldenGate Command Interpreter for Oracle

GGSCI (EURO) 1> DBLogin UserIDAlias ggueuro
Successfully logged into database.

GGSCI (EURO) 2> Add TranData EAST.PRODUCTS, Cols (product_name,
   vendor, qty_in_stock, modified_ts)

Logging of supplemental redo data enabled for table WEST.PRODUCTS.
TRANDATA for scheduling columns has been added on table 'EAST.PRODUCTS'.

GGSCI (EURO) 3> Info TranData EAST.PRODUCTS
Logging of supplemental redo log data is enabled for table EAST.PRODUCTS.
Columns supplementally logged for table EAST.PRODUCTS: PRODUCT_NUMBER,
PRODUCT_NAME, VENDOR, QTY_IN_STOCK, MODIFIED_TS.
```

Configuring and Starting the GoldenGate Manager on AMER and EURO

5. On the AMER_WEST_GGSCI tab, verify that the Manager parameter file is configured:

```
GGSCI (AMER)> Edit Param mgr
```

Verify that the following parameters are present:

```
Port 15000
DynamicPortList 15100-15150
PurgeOldExtracts ./dirdat/*, Usecheckpoints
```

Save and close this file.

6. Start the West Manager process:

```
GGSCI (AMER)> Start mgr
Manager started.

GGSCI (AMER)> Info manager
Manager is running (IP port westhost.15000 , Process ID 1993).

GGSCI (AMER)>
```

7. On the EURO_EAST_GGSCI tab, verify that the Manager parameter file is configured:

```
GGSCI (EURO)> Edit Param mgr
```

Verify that at least the following parameters are present:

```
Port 15001
PurgeOldExtracts ./dirdat/*, UseCheckpoints
```

Save and close this file.

8. Start the East Manager process:

```
GGSCI (EURO)> Start mgr
Manager started.

GGSCI (EURO)> Info manager
Manager is running (IP port easthost.15001, Process ID 18408).

GGSCI (EURO)>
```

Setting Up Change Data Capture on AMER

9. On the AMER_WEST_GGSCI tab, execute the following command to create and edit the parameter file for the "Extract West Conflict" group, ewconf:

```
GGSCI (AMER)> Edit Param ewconf
```

The text editor of your choice opens the empty ewconf.prm file. Add the following runtime parameters to the text file:

```
Extract ewconf
ExtTrail ./dirdat/wf
UserIDAlias gguamer
GetUpdateBefores
TranLogOptions ExcludeUser gguser
Table WEST.PRODUCTS GetBeforeCols(On Update All);
```

Check the *Reference for Oracle GoldenGate* in ~/Documents. Write down the purpose of the GetUpdateBefores parameter:

Answer:_____

Save and close the file.

10. On the AMER_WEST_GGSCI tab, add the Extract Group, ewconf to start extracting now and to write to a trail file in the dirdat directory with a name of wf by executing the following commands:

```
GGSCI (AMER)> Add Extract ewconf, TranLog, Begin Now
EXTRACT added.

GGSCI (AMER)> Add ExtTrail ./dirdat/wf, Extract ewconf
EXTTRAIL added.

GGSCI (AMER)>
```

11. On the AMER_WEST_GGSCI tab, set up the Extract Data Pump on the AMER database by executing the following command:

```
GGSCI (AMER)> Edit Param pwconf
```

The text editor of your choice opens the empty pwconf.prm file. Add the following runtime parameters to the text file:

```
Extract pwconf
RmtHost easthost, MgrPort 15001, Compress
RmtTrail ./dirdat/pf
Passthru
Table WEST.*;
```

Save and close the file.

12. On the AMER_WEST_GGSCI tab, add the Data Pump Extract group by executing the following commands. Configure the Extract Data Pump to read from the dirdat/wf local Extract Trail, transmit the data to the AMER server, and write it to the dirdat/pf Remote Trail:

```
GGSCI (AMER)> Add Extract pwconf, ExtTrailSource ./dirdat/wf
EXTRACT added.

GGSCI (AMER)> Add RmtTrail ./dirdat/pf, Extract pwconf
RMTTRAIL added.

GGSCI (AMER) >
```

Setting Up the EURO Database for Delivery

13. You need a checkpoint table for this task. It may already exist from a previous lab. After logging in to the database, try the Info command. If the checkpoint table exists, go to the next step. If the checkpoint table does not exist, run the Add command. To test or create the checkpoint table, on the EURO_EAST_GGSCI tab, execute the following commands:

```
GGSCI (EURO)> Info CheckpointTable
GGSCI (EURO)> Add  CheckpointTable
```

14. On the EURO_EAST_GGSCI tab, configure the Replicat Delivery component by executing the following command:

```
GGSCI (EURO)> Edit Param reconf
```

The text editor of your choice opens the empty `reconf.prm` file. Add the following runtime parameters to the text file:

```
Replicat reconf
AssumeTargetDefs
DiscardFile ./dirrpt/reconf.dsc, Purge
UserIDAlias ggueuro
AllowDupTargetMap
IgnoreUpdates
Map WEST.*, Target EAST.*;
GetUpdates
IgnoreInserts
IgnoreDeletes
Map WEST.products, Target EAST.products,
  CompareCols(On Update All),
  ResolveConflict (UpdateRowExists,
    (delta_resolution_method, UseDelta, Cols (qty_in_stock)),
    (max_resolution_method,   UseMax (modified_ts), Cols
(vendor, modified_ts)),
    (Default, Overwrite));
```

This `Map` statement performs the conflict detection and resolution on the `WEST.PRODUCTS` table for transactions performed on the `EAST.PRODUCTS` table. For all update operations, a query is executed against the target table and the data from the `qty_in_stock` column is returned.

If the Before (source) and current (target) values for `qty_in_stock` are not equal (that is, a conflict is detected), the source table value for `qty_in_stock` is subtracted from the source Before image `qty_in_stock` value. The result is then subtracted from the current target `qty_in_stock` value (returned from the query), and the result is applied to the target table.

Save and close the file.

15. On the EURO_EAST_GGSCI tab, add the Replicat by executing the following command:

```
GGSCI (EURO)> Add Replicat reconf, ExtTrail ./dirdat/pf
REPLICAT added.

GGSCI (EURO)>
```

Configuring Data Capture on EURO

16. On the EURO_EAST_GGSCI tab, execute the following command to configure the primary Extract:

```
GGSCI (EURO)> Edit Param eeconf
```

The text editor of your choice opens the empty eeconf.prm file. Add the following runtime parameters to the text file:

```
Extract eeconf
ExtTrail ./dirdat/eg
UserIDAlias ggueuro
GetUpdateBefores
TranLogOptions ExcludeUser gguser
Table EAST.PRODUCTS GetBeforeCols(On Update All);
```

17. On the EURO_EAST_GGSCI tab, add the Extract group by executing the following commands. Add the Extract group eeconf to start extracting now and to write to a trail file in the dirdat directory with a name of eg.

```
GGSCI (EURO)> Add Extract eeconf, TranLog, Begin Now
EXTRACT added.

GGSCI (EURO)> Add ExtTrail ./dirdat/eg, Extract eeconf
EXTTRAIL added.

GGSCI (EURO)>
```

18. On the EURO_EAST_GGSCI tab, configure the Extract Data Pump to read from the dirdat/eg local Extract Trail, transmit the data to the AMER server, and write it to the dirdat/pg Remote Trail:

```
GGSCI (EURO)> Edit Param peconf
```

The text editor of your choice opens the empty peconf.prm file. Add the following runtime parameters to the text file:

```
Extract peconf
RmtHost westhost, MgrPort 15000, Compress
RmtTrail ./dirdat/pg
Passthru
Table EAST.*;
```

Save and close the file.

19. On the EURO_EAST_GGSCI tab, add the Extract group by executing the following commands:

```
GGSCI (EURO)> Add Extract peconf, ExtTrailSource ./dirdat/eg
EXTRACT added.

GGSCI (EURO)> Add RmtTrail ./dirdat/pg, Extract peconf
RMTTRAIL added.

GGSCI (EURO)>
```

Creating the Delivery Components on the AMER Database

20. On the AMER_WEST_GGSCI tab, create the checkpoint table on the AMER database server by executing the following command:

```
GGSCI (AMER)> Add CheckpointTable
No checkpoint table specified, using GLOBALS specification (GGS_CHECKPOINT)...
Successfully created checkpoint table GGS_CHECKPOINT.

GGSCI (AMER)>
```

21. On the AMER_WEST_GGSCI tab, configure the Replicat delivery component:

```
GGSCI (AMER)> Edit Param rwconf
```

The text editor of your choice opens the empty rwconf.prm file. Add the following runtime parameters to the text file:

```
Replicat rwconf
AssumeTargetDefs
DiscardFile ./dirrpt/rwconf.dsc, Purge
UserIDAlias gguamer
AllowDupTargetMap
IgnoreUpdates
Map EAST.*, Target WEST.*;
GetUpdates
IgnoreInserts
IgnoreDeletes
Map EAST.products, Target WEST.products,
  CompareCols(On Update All),
  ResolveConflict (UpdateRowExists,
    (delta_resolution_method, UseDelta, Cols (qty_in_stock)),
    (max_resolution_method,   UseMax (modified_ts), Cols
(vendor, modified_ts)),
    (Default, Overwrite));
```

This is essentially the same conflict detection and resolution code as defined on the EURO delivery component, except that Map EAST/WEST is reversed.

Save and close the file.

22. On the AMER_WEST_GGSCI tab, add the Replicat by executing the following commands:

```
GGSCI (AMER)> Add Replicat rwconf, ExtTrail ./dirdat/pg
REPLICAT added.

GGSCI (AMER)>
```

Starting the GoldenGate Processes and Generating Data

23. On *both* the AMER_WEST_GGSCI and the EURO_EAST_GGSCI tabs, execute the following commands to start all of the configured Extracts and Replicats. Make sure that they are all in the RUNNING status.

    In AMER_WEST_GGSCI:

```
GGSCI (AMER)> Start ER *

Sending START request to MANAGER ...
EXTRACT EWCONF starting

Sending START request to MANAGER ...
EXTRACT PWCONF starting

Sending START request to MANAGER ...
REPLICAT RWCONF starting

GGSCI (AMER)> Info ER *

EXTRACT    EWCONF    Last Started 2013-12-11 09:47   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:09 ago)
Process ID           20712
Log Read Checkpoint  Oracle Redo Logs
                     2013-12-11 09:47:48  Seqno 307, RBA 8429056
                     SCN 0.6297903 (6297903)

EXTRACT    PWCONF    Last Started 2013-12-11 09:47   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:05 ago)
Process ID           20713
Log Read Checkpoint  File ./dirdat/wf000000
                     First Record  RBA 1415

REPLICAT   RWCONF    Last Started 2013-12-11 09:47   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:00 ago)
Process ID           20714
Log Read Checkpoint  File ./dirdat/pg000000
                     First Record  RBA 0

GGSCI (AMER)>
```

In EURO_EAST_GGSCI:

```
GGSCI (EURO)> Start ER *

Sending START request to MANAGER ...
EXTRACT EECONF starting

Sending START request to MANAGER ...
EXTRACT PECONF starting

Sending START request to MANAGER ...
REPLICAT RECONF starting

GGSCI (EURO)> Info ER *

EXTRACT    EECONF    Last Started 2013-12-11 09:49   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:08 ago)
Process ID           20783
Log Read Checkpoint  Oracle Redo Logs
                     2013-12-11 09:52:01  Seqno 240, RBA 25440256
                     SCN 0.4263409 (4263409)

EXTRACT    PECONF    Last Started 2013-12-11 09:52   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:04 ago)
Process ID           20846
Log Read Checkpoint  File ./dirdat/eg000000
                     First Record  RBA 1415

REPLICAT   RECONF    Last Started 2013-12-11 09:49   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:08 ago)
Process ID           20784
Log Read Checkpoint  File ./dirdat/pf000000
                     First Record  RBA 0

GGSCI (EURO)>
```

If any Oracle GoldenGate group shows a status of STOPPED or ABEND, view the report file, correct any errors, and then restart the group.

24. SQL*Plus should still be running for both instances AMER and EURO. Set the screen size to be friendlier:

    a. On the AMER_WEST_SQLPLUS tab, enter:

```
AMER_SQL> set lines 200
AMER_SQL> set pages 40
```

    b. On the EURO_EAST_SQLPLUS tab, enter:

```
EURO_SQL> set lines 200
EURO_SQL> set pages 40
```

25. Make sure that both the tables in both the instances have the same data, even down to the timestamp:

```
BOTH_SQL> SELECT * FROM products;
PRODUCT_NUMBER PRODUCT_NA VENDOR     QTY_IN_STOCK MODIFIED_TS
-------------- ---------- ---------- ------------ ----------------------------
           100 widget     Vendor ABC           20 21-MAY-12 12.00.00.000000 AM
```

26. Make sure non-conflicted updates get replicated through.
    a. On the AMER_WEST_SQLPLUS tab, enter:

```
AMER_SQL> UPDATE west.products SET product_name='foobar';
AMER_SQL> COMMIT;
```

    b. On the EURO_EAST_SQLPLUS tab, enter:

```
EURO_SQL> SELECT * FROM east.products;
PRODUCT_NUMBER PRODUCT_NA VENDOR     QTY_IN_STOCK MODIFIED_TS
-------------- ---------- ---------- ------------ ----------------------------
           100 foobar     Vendor ABC           20 21-MAY-12 12.00.00.000000 AM
EURO_SQL> UPDATE east.products SET product_name='whizbang';
EURO_SQL> COMMIT;
```

    c. On the AMER_WEST_SQLPLUS tab, enter:

```
AMER_SQL> SELECT * FROM west.products;
PRODUCT_NUMBER PRODUCT_NA VENDOR     QTY_IN_STOCK MODIFIED_TS
-------------- ---------- ---------- ------------ ----------------------------
           100 whizbang   Vendor ABC           20 21-MAY-12 12.00.00.000000 AM
```

    Notice that Oracle Database does not automatically update a timestamp unless you explicitly tell it to.

    d. Reset both databases. Enter the following script at the SQL prompt for both:
    `SQL> @Section14/Practice14-2/sqlscripts/database_reset.sql`

27. The basic testing methodology is as follows:
    a. Stop both pumps to simulate a network lag.
    b. Update and commit *different* local changes. This is the conflict.
    c. Restart both pumps to simulate end-of-lag.
    d. Examine the conflict resolution.
    e. Reset the database for the next test.

    While the Befores are checking all columns, the business rules you have chosen are:
    - For `name`, you do not care if there is a conflict (but you can tell if there is one)
    - For `vendor`, you want the latest
    - For `timestamp`, you want the biggest
    - For `qty_in_stock`, you want to add both differences together

    Re-examine `reconf.prm` to see the syntax to accomplish these rules.

28. The first test is to change the vendor and quantity on AMER/west first and EURO/east second. Perform the steps *in this order!* WEST first…
    a. On AMER, enter: `GGSCI (AMER)> Stop pwconf`
    On EURO enter: `GGSCI (EURO)> Stop peconf`

b. On WEST, enter:
SQL> **@Section14/Practice14-2/sqlscripts/trans_generator_west**
On EAST, enter:
SQL> **@Section14/Practice14-2/sqlscripts/trans_generator_east**
On both of them it will **SELECT * FROM products;** so you see what you did.

c. On AMER, enter: GGSCI (AMER)> **Start pwconf**
On EURO, enter: GGSCI (EURO)> **Start peconf**

d. Enter **SELECT * FROM products;** on both sides so you see what it did.

Applying the rules from the Replicats:

- Because vendor takes the latest based on timestamp, both sides show EURO's value of XYZ.
- Because timestamp is also the latest, both sides show EURO's value.
- Because qty_in_stock started at 20, then both additions are applied, so both sides show 20+10+4=34.

e. Reset both databases. Enter the following script at the SQL prompt for both:
SQL> **@Section14/Practice14-2/sqlscripts/database_reset.sql**

29. The second test is the same as the first but stop and start in a different order (EAST first).

a. On EURO, enter: GGSCI (EURO)> **Stop peconf**
On AMER, enter: GGSCI (AMER)> **Stop pwconf**

b. On EAST , enter (or use the up arrow to repeat history):
SQL> **@Section14/Practice14-2/sqlscripts/trans_generator_east**
On WEST enter:
SQL> **@Section14/Practice14-2/sqlscripts/trans_generator_west**
On both of them it will **SELECT * FROM products;** so you see what you did.

c. On EURO, enter: GGSCI (EURO)> **Start peconf**
On AMER, enter: GGSCI (AMER)> **Start pwconf**

d. Enter **SELECT * FROM products;** on both sides so you see what it did.

Applying the rules from the Replicats:

- Because vendor takes the latest based on timestamp, both sides show AMER's value of DEF.
- Because timestamp is also the latest, both sides show AMER's value.
- Because qty_in_stock started at 20, then both additions are applied, so both sides show 20+10+4=34. This is unchanged by the test order.

e. Reset both databases. Enter the following script at the SQL prompt for both:
SQL> **@Section14/Practice14-2/sqlscripts/database_reset.sql**

30. The last test is where there is a conflict, but you do not care.

a. On EURO, enter: GGSCI (EURO)> **Stop peconf**
On AMER, enter: GGSCI (AMER)> **Stop pwconf**

b. On EURO, enter SQL> **UPDATE products SET product_name='dingbat';**
On EURO, enter SQL> **COMMIT;**
On AMER, enter SQL> **UPDATE products SET product_name='wingnut', qty_in_stock=15;**
On AMER, enter SQL> **COMMIT;**

c. On AMER, enter: `GGSCI (AMER)> **Start pwconf**`
On EURO, enter: `GGSCI (EURO)> **Start peconf**`

d. Enter **`SELECT * FROM products;`** on both sides so you see what it did.

Applying the rules from the Replicats:

- There is a conflict on `product_name`, but you do not care, so it defaults to `Overwrite`. EURO sets local `product_name` to `dingbat`, AMER sets local `product_name` to `wingnut`. When the pumps start, AMER overwrites `product_name` on EURO with `wingnut`, then the EURO pump starts and overwrites `product_name` on AMER with `dingbat`; in other words, they switch. The "winner" depends on the order of the pumps, not the order of the transactions.
- Even though there is a conflict, it does not affect `qty_in_stock`, so that replicates normally.
- `Timestamp` was not updated and, therefore, not a conflict.

e. Reset both databases. Enter the following script at the SQL prompt for both:
`SQL> **@Section14/Practice14-2/sqlscripts/database_reset.sql**`

31. The statistics record DML and conflicts. Examine the conflict detections and resolutions, those items with "CDR" in the field:

a. On the EURO_EAST_GGSCI tab, enter: **`Stats reconf, ReportCDR`**

```
GGSCI (EURO)> Stats reconf, ReportCDR

Sending STATS request to REPLICAT RECONF ...
Start of Statistics at 2013-12-11 19:47:15.
Replicating from WEST.PRODUCTS to EAST.PRODUCTS:

...many lines omitted for clarity...

*** Total statistics since 2013-12-11 13:06:17 ***
   Total inserts                                0.00
   Total updates                                8.00
   Total deletes                                0.00
   Total discards                               0.00
   Total operations                             8.00
   Total CDR conflicts                          4.00
   CDR resolutions succeeded                    4.00
   CDR UPDATEROWEXISTS conflicts                4.00
...many lines omitted for clarity...
```

b. On the AMER_WEST_GGSCI tab, enter: **Stats rwconf, ReportCDR**

```
GGSCI (AMER)> Stats rwconf, ReportCDR

Sending STATS request to REPLICAT RWCONF ...
Start of Statistics at 2013-12-11 19:47:15.
Replicating from EAST.PRODUCTS to WEST.PRODUCTS:

...many lines omitted for clarity...

*** Total statistics since 2013-12-11 13:06:17 ***
   Total inserts                                0.00
   Total updates                                8.00
   Total deletes                                0.00
   Total discards                               0.00
   Total operations                             8.00
   Total CDR conflicts                          4.00
   CDR resolutions succeeded                    4.00
   CDR UPDATEROWEXISTS conflicts                4.00
...many lines omitted for clarity...
```

Your numbers may be slightly different depending on your experimentation.

32. Clean up the Oracle GoldenGate environment by executing the following GGSCI commands on both the source and target servers:
    a. Exit both GGSCI prompts. This is necessary for the kill shell script.
    b. On the fifth tab oracle@hostname, at the OS prompt, run:
       [OS_prompt] **~/labs/setup/kill-all-ogg.sh**
    c. Exit both SQL*Plus prompts. All tabs should be back at the OS prompts, but still in their proper directories.

This completes Practice 14-2. This completes the practices for Lesson 14. Stop here.

Practices for Lesson 15: DDL Replication

Chapter 15

Practices for Lesson 15: Overview

Practices Overview

In this practice, you will support the synchronization of DDL operations in homogeneous, unidirectional environments.

Practice 15-1: DDL Replication Database Setup

Assumptions

To enable DDL capture, several modifications must be made to the source database; specifically, the creation of five tables in a tablespace with sufficient space to allow growth of the marker and history tables, a database trigger and associated packages, and a user role.

Before attempting to set up DDL capture, all open sessions in the database must be closed (which requires a database outage in a production environment).

Tasks

To set up the source database for DDL capture, execute the following on the AMER_WEST_SQLPLUS tab:

1. Choose which schema DDL objects will be created in. (For this practice, the GGUSER schema will be used because it exists on both the source and the target.) Using the text editor of your choice, edit the GLOBALS file, located in the source Oracle GoldenGate home directory, and add the following parameter: **GGSCHEMA GGUSER**.

```
[OS_prompt labs] cd $GG_AMER_HOME
[OS_prompt gg_amer] gedit GLOBALS
```

```
CheckpointTable GGS_CHECKPOINT
GGSCHEMA GGUSER
```

   Save and close the file.

2. On the AMER_WEST_SQLPLUS tab, start SQL*Plus as sysdba:

```
[OS_prompt gg_amer] rlsqlplus / as sysdba
```

3. Execute the following DBA commands:

```
SQL> GRANT EXECUTE ON UTL_FILE TO gguser;
Grant succeeded.

SQL>
```

4. Run the `marker_setup.sql` script. When prompted for a schema, enter GGUSER:

```
SQL> @marker_setup.sql

Marker setup script

You will be prompted for the name of a schema for the Oracle GoldenGate
database objects.
NOTE: The schema must be created prior to running this script.
NOTE: Stop all DDL replication before starting this installation.

Enter Oracle GoldenGate schema name:GGUSER

Marker setup table script complete, running verification script...
Please enter the name of a schema for the GoldenGate database objects:
Setting schema name to GGUSER

MARKER TABLE
-------------------------------
OK

MARKER SEQUENCE
-------------------------------
OK

Script complete.
SQL>
```

5. Run the `ddl_setup.sql` script. When prompted for a schema, enter GGUSER.

```
SQL> @ddl_setup.sql

Oracle GoldenGate DDL Replication setup script

Verifying that current user has privileges to install DDL Replication...
You will be prompted for the name of a schema for the Oracle GoldenGate
database objects.
NOTE: For an Oracle 10g source, the system recycle bin must be disabled. For
Oracle 11g and later, it can be enabled.
NOTE: The schema must be created prior to running this script.
NOTE: Stop all DDL replication before starting this installation.

Enter Oracle GoldenGate schema name:GGUSER
Working, please wait ...
Spooling to file ddl_setup_spool.txt

Checking for sessions that are holding locks on Oracle Golden Gate metadata
tables ...
Check complete.

Using GGUSER as a Oracle GoldenGate schema name.
Working, please wait ...

DDL replication setup script complete, running verification script...
Please enter the name of a schema for the GoldenGate database objects:
Setting schema name to GGUSER

CLEAR_TRACE STATUS:

Line/pos                                Error
--------------------------------------- ----------------------------------
No errors                               No errors

... Many lines omitted for clarity ...

Analyzing installation status...

STATUS OF DDL REPLICATION
------------------------------------------------------------------------------
SUCCESSFUL installation of DDL Replication software components

Script complete.
SQL>
```

6. Run the `role_setup.sql` script. When prompted for a schema, enter GGUSER:

```
SQL> @role_setup.sql

GGS Role setup script

This script will drop and recreate the role GGS_GGSUSER_ROLE
To use a different role name, quit this script and then edit the params.sql
script to change the gg_role parameter to the preferred name.

You will be prompted for the name of a schema for GoldenGate database objects.
NOTE: The schema must be created prior to running this script.
NOTE: Stop all DDL replication before starting this installation.

Enter GoldenGate schema name:GGUSER
Wrote file role_setup_set.txt

PL/SQL procedure successfully completed.

Role setup script complete
Grant this role to each user assigned to the Extract, GGSCI, and Manager
processes, by using the following SQL command:
GRANT GGS_GGSUSER_ROLE TO <loggedUser>
where <loggedUser> is the user assigned to the GoldenGate processes.
SQL>
```

7. As specified in the `role_setup.sql` output, grant GGSUSER_ROLE to the Oracle GoldenGate database user:

```
SQL> GRANT ggs_ggsuser_role TO gguser;
Grant succeeded.
SQL>
```

8. Enable DDL triggers:

```
SQL> @ddl_enable.sql
Trigger altered.
SQL>
```

9. Make the `ddl_pin` script part of the database startup. It must be invoked with the Oracle GoldenGate DDL username. (This script improves the performance of the DDL trigger and requires the Oracle `dbms_shared_pool` system package.)

```
SQL> @ddl_pin GGUSER


PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.
PL/SQL procedure successfully completed.

SQL> exit
[OS_prompt]
```

Exit SQL*Plus. The database is now set up and enabled for DDL capture. Had you been using Integrated Extract, none of these *@scripts* for triggers would have been necessary because Integrated Extract supports DDL replication by default.

This completes Practice 15-1. Continue now with Practice 15-2.

Practice 15-2: DDL Replication Oracle GoldenGate Setup

Tasks

Housekeeping Considerations

For continuing operations, the DDL marker, history, and trace tables must be monitored and routinely cleared of old or nonessential data.

1. On the AMER_WEST_GGSCI tab, start GGSCI:

```
[OS_prompt gg_amer] rlggsci
GGSCI (AMER) 1> set editor gedit
```

2. On the AMER_WEST_GGSCI tab, edit the mgr.prm parameter file. Add the following three lines to the end of the file:

```
UserIDAlias gguamer
PurgeMarkerHistory MinKeepDays 3, MaxKeepDays 5
PurgeDDLHistory MinKeepDays 3, MaxKeepDays 5
```

```
Port 15000
PurgeOldExtracts ./dirdat/*, UseCheckpoints, MinKeepDays 1
DynamicPortList 15100-15150
UserIDAlias gguamer
PurgeMarkerHistory MinKeepDays 3, MaxKeepDays 5
PurgeDDLHistory MinKeepDays 3, MaxKeepDays 5
```

Do not be concerned if the existing lines in your file slightly differ from the book. The purpose of this task is to add the last three lines that are for PurgeMarkerHistory and PurgeDDLHistory. Save and close the file.

3. On the AMER_WEST_GGSCI tab, stop and restart the Oracle GoldenGate Manager:

```
GGSCI (AMER)> Stop mgr !
GGSCI (AMER)> Start mgr
```

Note: To prevent the DDL trace file from consuming excessive disk space, run the @ddl_cleartrace.sql script on a regular basis. This script deletes the file, but Oracle GoldenGate will create it again. This script must be run manually or via cron.

Setting Up the Workshop Database and Configuration

Before continuing further, make sure that you start with clean schema areas for your lab.

4. On the AMER_WEST_SQLPLUS tab, navigate to ~/labs, and then execute the following SQL script to clean out the source schema:

```
[OS_prompt] cd ~/labs
[OS_prompt labs] rlsqlplus west@amer/oracle_4U
@Section15/sqlscripts/drop_tables.sql
PL/SQL procedure successfully completed.

AMER_SQL> SELECT * FROM user_tables;
no rows selected
```

5. On the EURO_EAST_SQLPLUS tab, navigate to the same ~/labs, and execute the same SQL script to clean out the target schema:

```
[OS_prompt] cd ~/labs
[OS_prompt labs] rlsqlplus east@euro/oracle_4U
@Section15/sqlscripts/drop_tables.sql
PL/SQL procedure successfully completed.

EURO_SQL> SELECT * FROM user_tables;
no rows selected
```

   Leave SQL*Plus running.

6. Dropping the tables in the previous steps should have also cleared the TranData (transaction data). Verify this on the AMER_WEST_GGSCI tab:

```
GGSCI (AMER)> DBLogin UserIDAlias gguamer
Successfully logged into database.

GGSCI (AMER)> Info TranData WEST.*
ERROR: No viable tables matched specification.

GGSCI (AMER)>
```

   Note: The last step of the practice verifies that the TranData comes back automatically.

7. To set up Oracle GoldenGate Change Data Capture on the source server AMER_WEST_SQLPLUS tab, configure the change data capture extract (eddl.prm) with the following parameters:

   ```
   GGSCI (AMER)> Edit Param eddl
   ```

   The text editor of your choice opens the empty eddl.prm file. Enter the following parameters:

```
Extract eddl
ExtTrail ./dirdat/ed
UserIDAlias gguamer
-- DDL Include Mapped ObjName "WEST.*"  <-- This won't work!
DDL Include Mapped ObjName WEST.*
DDLOptions AddTranData, Report
Table WEST.*;
```

   Save and close the file.

8. On the AMER_WEST_GGSCI tab, add the Extract group and two local Extract Trails by executing the following commands:

```
GGSCI (AMER)> Add Extract eddl, TranLog, Begin Now
EXTRACT added.

GGSCI (AMER)> Add ExtTrail ./dirdat/ed, Extract eddl
EXTTRAIL added.

GGSCI (AMER)>
```

9. On the source AMER_WEST_GGSCI tab, configure the Extract Data Pump (pddl.prm) to read from the dirdat/ed local Extract Trail, transmit the data to the target server, and write it to the dirdat/pd Remote Trail:

```
GGSCI (AMER)> Edit Param pddl
```

The text editor of your choice opens the empty pddl.prm file. Add the following runtime parameters to the text file:

```
Extract pddl
RmtHost easthost, MgrPort 15001, Compress
RmtTrail ./dirdat/pd
Passthru
Table WEST.*;
```

Save and close the file.

10. Add the Extract group by executing the following GGSCI commands:

```
GGSCI (AMER)> Add Extract pddl, ExtTrailSource ./dirdat/ed
GGSCI (AMER)> Add RmtTrail ./dirdat/pd, Extract pddl
```

11. On the target EURO_EAST_GGSCI tab, Replicat checkpoints to a special table created in the database. This checkpoint table ensures that the Replicat checkpoint is part of the transaction and is used to ensure data integrity. If it does not already exist, create the Checkpoint Table.

On the target server EURO_EAST_GGSCI tab, start GGSCI and execute the following commands:

```
[OS_prompt gg_euro] rlggsci
GGSCI (EURO) 1> set editor gedit
GGSCI (EURO) 2> DBLogin UserIDAlias ggueuro
GGSCI (EURO) 3> Info CheckpointTable
GGSCI (EURO) 4> Add  CheckpointTable
```

Note: Because this is a frequent-enough sequence, it is useful to create an Obey file with these commands that could be invoked by entering:

```
GGSCI> Obey /home/oracle/labs/setup/startup.oby
```

12. On the target server EURO_EAST_GGSCI tab, configure the Replicat delivery components by executing the following command:

```
GGSCI (EURO)> Edit Param rddl
```

The text editor of your choice opens the empty rddl.prm file. Add the following runtime parameters to the text file:

```
Replicat rddl
DiscardFile ./dirrpt/rddl.dsc, Purge
AssumeTargetDefs
UserIDAlias ggueuro
DDL Include Mapped
Map WEST.*, Target EAST.*;
```

Save and close the file.

13. Add the Replicat by executing the following GGSCI command:

```
GGSCI (EURO)> Add Replicat rddl, ExtTrail ./dirdat/pd
```

14. Verify that the mgr process is running on ***both*** the AMER_WEST_GGSCI tab and the EURO_EAST_GGSCI tab.

```
GGSCI (BOTH)> Info mgr
```

If the manager process is not already running, start it, and then verify that it is running:

```
GGSCI (BOTH)> Start mgr
GGSCI (BOTH)> Info mgr
```

To start the Extract and Replicat processes, issue the following command on ***both*** the AMER and EURO instances.

On AMER:

```
GGSCI (AMER)> Start ER *

Sending START request to MANAGER ...
EXTRACT EDDL starting

Sending START request to MANAGER ...
EXTRACT PDDL starting

GGSCI (AMER)> Info ER *

EXTRACT    EDDL      Last Started 2013-12-11 17:42   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:04 ago)
Process ID           29972
Log Read Checkpoint  Oracle Redo Logs
                     2013-12-11 17:42:51  Seqno 308, RBA 22850048
                     SCN 0.6319581 (6319581)

EXTRACT    PDDL      Last Started 2013-12-11 17:42   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:01 ago)
Process ID           29973
Log Read Checkpoint  File ./dirdat/ed000000
                     First Record  RBA 1411

GGSCI (AMER)>
```

On EURO:

```
GGSCI (EURO)> Start ER *

Sending START request to MANAGER ...
REPLICAT RDDL starting

GGSCI (EURO)> Info ER *

REPLICAT   RDDL      Last Started 2013-12-11 17:42   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:08 ago)
Process ID           29995
Log Read Checkpoint  File ./dirdat/pd000000
                     First Record  RBA 0

GGSCI (EURO)>
```

Generate Source Database Activity

15. On the AMER_WEST_SQLPLUS tab, execute the following SQL script:

```
AMER_SQL> @Section15/sqlscripts/source_database.sql

Table created.
Table created.
Table created.
Table created.
Table created.
Table created.

AMER_SQL>
```

16. On the AMER_WEST_SQLPLUS tab, execute the following SQL script:

```
AMER_SQL> @Section15/sqlscripts/seed_database.sql

PL/SQL procedure successfully completed.

AMER_SQL>
```

17. On the AMER_WEST_SQLPLUS tab, execute the following SQL script:

```
AMER_SQL> @Section15/sqlscripts/trans_generator.sql

PL/SQL procedure successfully completed.

AMER_SQL>
```

Wait a few seconds for the transaction generator to complete.

Validating the Results

18. On the EURO_EAST_SQLPLUS tab, execute the following query:

```
EURO_SQL> SELECT table_name FROM user_tables;

TABLE_NAME
------------------------------
ACCOUNT
ACCOUNT_TRANS
BRANCH
TELLER
TELLER_TRANS
BRANCH_ATM

6 rows selected.

SQL>
```

Six tables should have been created/replicated in the EAST schema.

19. From the AMER_WEST_GGSCI tab, execute the following command:

```
GGSCI (AMER)> Stats eddl, Total

Sending STATS request to EXTRACT EDDL ...
Start of Statistics at 2013-12-12 07:29:28.

DDL replication statistics (for all trails):

*** Total statistics since extract started     ***
        Operations                            6.00
        Mapped operations                     6.00
        Unmapped operations                   0.00
        Other operations                      0.00
        Excluded operations                   0.00

Output to ./dirdat/ed:

Extracting from GGUSER.GGS_MARKER to GGUSER.GGS_MARKER:

*** Total statistics since 2013-12-12 07:22:59 ***
        No database operations have been performed.

Extracting from WEST.ACCOUNT to WEST.ACCOUNT:

*** Total statistics since 2013-12-12 07:22:59 ***
        Total inserts                          1060.00
        Total updates                          1500.00
        Total deletes                             0.00
        Total discards                            0.00
        Total operations                       2560.00

   ... Many lines omitted for clarity ...

End of Statistics.

GGSCI (AMER)>
```

The Stats output should show mapped DDL operations followed by transaction counts for all tables where data was captured.

20. Execute the same command for the Replicat on the EURO_EAST_GGSCI tab:

```
GGSCI (EURO)> Stats rddl, Total

Sending STATS request to REPLICAT RDDL ...
Start of Statistics at 2013-12-12 07:33:18.

DDL replication statistics:

*** Total statistics since replicat started     ***
        Operations                          6.00
        Mapped operations                   6.00
        Unmapped operations                 0.00
        Other operations                    0.00
        Excluded operations                 0.00
        Errors                              0.00
        Retried errors                      0.00
        Discarded errors                    0.00
        Ignored errors                      0.00

Replicating from WEST.ACCOUNT to EAST.ACCOUNT:

*** Total statistics since 2013-12-12 07:23:43 ***
        Total inserts                       1060.00
        Total updates                       1500.00
        Total deletes                          0.00
        Total discards                         0.00
        Total operations                    2560.00

   ... Many lines omitted for clarity ...

End of Statistics.

GGSCI (EURO)>
```

All counts should match. Notice that there is more DDL information on the Replicat than on the Extract.

21. Verify that the transaction data was added to the AMER source because of parameter `DDLOptions AddTranData, Report`. You know it had to be added because the Replicat worked, but check for it anyway.

```
GGSCI (AMER)> Info TranData WEST.*

Logging of supplemental redo log data is enabled for table WEST.ACCOUNT.
Columns supplementally logged for table WEST.ACCOUNT: ACCOUNT_NUMBER.

Logging of supplemental redo log data is enabled for table WEST.ACCOUNT_TRANS.
Columns supplementally logged for table WEST.ACCOUNT_TRANS: ACCOUNT_NUMBER,
TRANS_NUMBER, ACCOUNT_TRANS_TS.

Logging of supplemental redo log data is enabled for table WEST.BRANCH.
Columns supplementally logged for table WEST.BRANCH: BRANCH_NUMBER.

Logging of supplemental redo log data is enabled for table WEST.BRANCH_ATM.
Columns supplementally logged for table WEST.BRANCH_ATM: BRANCH_NUMBER,
ATM_NUMBER, TRANS_NUMBER, ATM_TRANS_TS.

Logging of supplemental redo log data is enabled for table WEST.TELLER.
Columns supplementally logged for table WEST.TELLER: TELLER_NUMBER.

Logging of supplemental redo log data is enabled for table WEST.TELLER_TRANS.
Columns supplementally logged for table WEST.TELLER_TRANS: TELLER_NUMBER,
TRANS_NUMBER, TELLER_TRANS_TS.

GGSCI (AMER)>
```

This completes Practice 15-2. This completes the practices for Lesson 15. Stop here.

If you are in a classroom, ask the instructor whether you should leave the PCs running or not, signed in, or logged off.