



Hardware and Software
Engineered to Work Together

Oracle GoldenGate 12c: Advanced Configuration for Oracle

Activity Guide
D89113GC10
Edition 1.0 | July 2015 | D91917

Learn more from Oracle University at oracle.com/education/

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

Disclaimer

This document contains proprietary information and is protected by copyright and other intellectual property laws. You may copy and print this document solely for your own use in an Oracle training course. The document may not be modified or altered in any way. Except where your use constitutes "fair use" under copyright law, you may not use, share, download, upload, copy, print, display, perform, reproduce, publish, license, post, transmit, or distribute this document in whole or in part without the express authorization of Oracle.

The information contained in this document is subject to change without notice. If you find any problems in the document, please report them in writing to: Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA. This document is not warranted to be error-free.

Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Trademark Notice

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Author

Elio Bonazzi

Technical Contributors and Reviewers

Joe Debuzna, Randall Richeson

This book was published using: [Oracle Tutor](#)

Table of Contents

Course Practice Environment: Security Credentials	I-1
Practices for Lesson 1: Introduction.....	1-1
Practices for Lesson 1: Overview.....	1-2
Practices for Lesson 2: Oracle GoldenGate Integrated Capture/Apply.....	2-1
Practices for Lesson 2: Overview.....	2-2
Practice 2-1: Checking the Oracle Instance Parameters That Affect Replication.....	2-6
Practice 2-2: Preparing the Oracle Source Database	2-8
Practice 2-3: Preparing the Oracle Target Database	2-14
Practice 2-4: Preparing the Oracle GoldenGate Environment.....	2-18
Practice 2-5: Set Up Replication	2-20
Practices for Lesson 3: Integrated Capture Deployment and Required Components.....	3-1
Practices for Lesson 3: Overview.....	3-2
Practice 3-1: Using the Oracle GoldenGate Wallet Facility	3-3
Practice 3-2: Preparing Source and Downstream Database for Integrated Capture.....	3-6
Practice 3-3: Deploying Integrated Capture by Using Downstream Real-Time Mode.....	3-16
Practices for Lesson 4: Oracle GoldenGate with Oracle Real Application Clusters.....	4-1
Practices for Lesson 4: Overview.....	4-2
Practice 4-1: Prepare the Environment	4-3
Practice 4-2: Configuring DBFS	4-13
Practice 4-3: Configuring DBFS as a CRS service.....	4-21
Practices for Lesson 5: Oracle GoldenGate Event Marker System	5-1
Practices for Lesson 5: Overview.....	5-2
Practice 5-1: Triggering an End-of-Day Job (Database Backup).....	5-3
Practice 5-2: Stopping Processing on Detecting Data Anomalies	5-11
Practice 5-3: Tracing the Insertion of Rows Where Specific Columns Carry Abnormally High Values.....	5-17
Practice 5-4: Executing Shell Scripts to Validate Data and Sending an Email when Validation Fails	5-21
Practices for Lesson 6: Data Mapping, Data Selection/Filtering, and Data Transformation	6-1
Practices for Lesson 6: Overview.....	6-2
Practice 6-1: Column Mapping Between Dissimilar Source and Target Tables	6-7
Solution 6-1: Column Mapping Between Dissimilar Source and Target Tables	6-8
Practice 6-2: Data Selection in Extract and Replicat Streams.....	6-14
Solution 6-2: Data Selection in Extract and Replicat Streams.....	6-15
Practice 6-3: Applying Data Transformation Functions	6-18
Solution 6-3: Applying Data Transformation Functions	6-19
Practice 6-4: Using SQLEXEC to Invoke a Stored Procedure	6-22
Solution 6-4: Using SQLEXEC to Invoke a Stored Procedure	6-23
Practices for Lesson 7: Custom Behavior Through User Exits	7-1
Practices for Lesson 7: Overview.....	7-2
Practice 7-1: Verifying the Java Environment and Installing Apache ActiveMQ and Oracle GoldenGate for Java	7-3
Practice 7-2: Configuring Data Capture	7-7
Practice 7-3: Configuring Data Delivery to ActiveMQ.....	7-11
Practice 7-4: Generating Data and Verifying Replication	7-16
Practices for Lesson 8: Configuring Zero-Downtime Migration Replication	8-1
Practices for Lesson 8: Overview.....	8-2

Practice 8-1: Configuring a Zero–Down Time Migration.....	8-3
Practice 8-2: Configuring the Fallback	8-14
Practice 8-3: Performing Migration Cutover	8-17
Practice 8-4: Aborting the Migration	8-22
Practices for Lesson 9: Bidirectional Replication: Two-Node Configuration.....	9-1
Practices for Lesson 9: Overview.....	9-2
Practice 9-1: Configuring an Active-Active Replication Topology.....	9-3
Practice 9-2: Setting Up DDL Replication and Verifying Correct Propagation of DML and DDL Statements.	9-17
Practice 9-3: Enabling CDR-Based Conflict Detection and Resolution	9-22
Practices for Lesson 10: Conflict Detection and Resolution: Custom Techniques.....	10-1
Practices for Lesson 10: Overview.....	10-2
Practice 10-1: Data Conflict Avoidance Techniques	10-3
Practice 10-2: Resolving Data Conflicts by Using Quantitative Resolution Methods.....	10-9
Practices for Lesson 11: Multi-Master Replication Topology: Three-Node Configuration	11-1
Practices for Lesson 11: Overview.....	11-2
Practice 11-1: Creating a Three-Way Active-Active Replication Environment	11-4
Practice 11-2: Configuring CDR for a Multi-Master Replication Environment	11-25
Practices for Lesson 12: Active Data Guard and Oracle GoldenGate: How to Achieve Maximum Availability	12-1
Practices for Lesson 12: Overview.....	12-2
Practice 12-1: Prepare the Environment	12-3
Practice 12-2: Configuring Oracle GoldenGate Replication	12-12
Practice 12-3: Configuring the Oracle Bundled Agent (XAG)	12-23
Practice 12-4: Failing a RAC Node to Test Oracle GoldenGate Survival.....	12-30

Course Practice Environment: Security Credentials

Chapter I

Course Practice Environment: Security Credentials

For OS usernames and passwords, see the following:

- If you are attending a classroom-based or live virtual class, ask your instructor or LVC producer for OS credential information.
- If you are using a self-study format, refer to the communication that you received from Oracle University for this course.

For product-specific credentials used in this course, see the following table:

Oracle RDBMS and Oracle GoldenGate-Specific Credentials		
Product/Application	Username	Password
System User	oracle	oracle
Database	SYS	oracle
Database	SYSTEM	oracle
Database	SRC_USER	oracle
Database	TRG_USER	oracle
Database	C##OGG_DWNADM	oracle
Database	C##OGG_ADMIN	oracle
Database	C##OGG_ADMIN@ogg1	oracle
Database	C##OGG_DWNADM@oggdwn1	oracle
Database	WEST	Welcome1
Database	EAST	Welcome1
Database	SIMULAPP	simulapp
Database	NODE1	oracle
Database	NODE2	oracle
Database	NODE3	oracle
ActiveMQ Web Admin Console	ADMIN	admin

Practices for Lesson 1: Introduction

Chapter 1

Practices for Lesson 1: Overview

Practices Overview

There are no practices for Lesson 1. Stop here.

However, it is a good idea in customized classes (such as non-Oracle classrooms) to verify that the sign-on accounts are working before you reach the practices when you actually need them. Thus, if something is not working, you or the instructor has time to initiate a repair.

Practices for Lesson 2: Oracle GoldenGate Integrated Capture/Apply

Chapter 2

Practices for Lesson 2: Overview

Practices Overview

In these practices, you will prepare your source and target environments for replication, and perform replication from a source pluggable database to a target (different) pluggable database. Several steps are required for Oracle GoldenGate to properly connect to the source database and replicate to the target. These steps include verifying that certain database parameters have been set, and that all required privileges have been granted to the owners or schemas performing replication.

These practices will guide you through the correct sequence of procedures that are required to set up a replication environment by using Oracle databases. In addition, you will configure the Oracle GoldenGate environments (`oggsrc` for the replication source and `oggtrg` for the replication target).

Exploring your environment

The workstation that you will use for the practices runs Oracle Linux, configured with a Gnome graphical environment. You will mainly connect as the "oracle" user. Occasionally you will connect as "root" to perform tasks that require root privileges. Ask your instructor for the passwords for the "oracle" and "root" users.

Connect to the workstation assigned to you by using the "oracle" user. Open a terminal window to start interacting with your environment.

The Linux environment where you perform your practices has been customized with a few utilities that you will find useful to either minimize keystrokes or operate in graphical mode.

The `sqlplus` utility is the preferred environment to interact with the Oracle RDBMS. However, the SqlDeveloper GUI is also available, in case you prefer to browse the Oracle database by using a graphical environment.

In addition, the Linux `rlwrap` utility has been installed and is available by default when you invoke either `sqlplus` or `GGSCI`. `rlwrap` allows you to recover previously entered lines of text by using the up and down arrow keys. You can also position the cursor under any letter in the line by using the left and right arrow keys.

The “oracle” user defines eight shell “aliases,” which you will use throughout the practices:

```
alias ggsci='rlwrap ./ggsci'  
alias sqlplus='rlwrap sqlplus'  
alias node1='cd /u03/ogg/ogg_node1'  
alias node2='cd /u03/ogg/ogg_node2'  
alias node3='cd /u03/ogg/ogg_node3'  
alias oggsrc='cd /u03/ogg/ogg_src'  
alias oggtrg='cd /u03/ogg/ogg_trg'  
alias  
sqldeveloper='/u01/app/oracle/product/12.1.0/dbhome_1/sqldeveloper/sqldeveloper.sh &'
```

When you enter “sqlplus,” you are in fact wrapping the `sqlplus` utility with the `rlwrap` command. While you are in `sqlplus`, if you mistype a letter while entering a command, you can simply use the arrow keys to recover the line that contains the syntactical mistake. Then position the cursor under or immediately to the right of the typo and correct it to re-enter the corrected line.

The same applies to the `GGSCI` utility. You can recover previously entered lines by using the arrow keys, rather than retyping them. Finally, you can launch the SQL Developer GUI by entering “`sqldeveloper`” (all lowercase) at the shell prompt:



Although all steps in all practices use only `sqlplus` to interact with the Oracle RDBMS, nothing prevents you from using the GUI, either instead of `sqlplus` or as an additional tool.

Similarly, regarding the preferred editor, all practices assume “`vi`” as the default editor, but the Linux GUI of choice, Gnome, also provides a graphical editor called “`gedit`.”

You can opt to use `gedit` instead of `vi`. If you decide to do so, remember to enter this command every time you launch `GGSCI`:

```
GGSCI ([HOST]) Set Editor gedit
```

The `EDITOR` shell variable determines the default editor for several applications (unfortunately not `GGSCI`)! By default, in your environment, `EDITOR` points to “`vi`.” If you invoke `sqlplus` and you want to edit the last command entered, by default you would invoke `vi`. However, if you prefer using `gedit`, you can change the `EDITOR` variable to point to `gedit`:

```
[OS prompt]$ export EDITOR=gedit
```

Helper scripts provided in /home/oracle/labs/Practice02/configs:

Catch-up/Clean-up script	Purpose
catch-up_2_1_to_2_4.sh	Executes Practices 2-1 through 2-4
catch-up_2_5.sh	Executes steps for Practice 2-5
clean-up_2_1_to_2_4.sh	Undoes all steps performed during Practices 2-1 through 2-4
clean-up_2_5.sh	Undoes all steps performed during Practice 2-5

Note: Do not execute the `clean-up_2_5.sh` script at the end of the practice. Practice 3 expects the steps performed during Practice 2-5 to be present. The `clean-up_2_1_to_2_4.sh` and `clean-up_2_5.sh` scripts are provided in case a destructive mistake is made during the practice and you want to start the practice from scratch on a clean state.

Oracle GoldenGate operating in an Oracle 12c multitenant environment

Nearly all practices of this course use the new Oracle RDBMS 12c multitenant container database architecture. The multitenant architecture enables an Oracle database to function as a multitenant container database (CDB) that includes zero, one, or many customer-created pluggable databases (PDBs). A PDB is a portable collection of schemas, schema objects, and non-schema objects that appears to an Oracle Net client as a non-CDB.

A container is either a PDB or the root container (also called *the root*). The root is a collection of schemas, schema objects, and non-schema objects to which all PDBs belong.

Every CDB has the following containers:

1) Exactly one root

The root stores Oracle-supplied metadata and common users. An example of metadata is the source code for Oracle-supplied PL/SQL packages. A common user is a database user known in every container. The root container is named `CDB$ROOT`.

2) Exactly one seed PDB

The seed PDB is a system-supplied template that the CDB can use to create new PDBs.

The seed PDB is named `PDB$SEED`. Objects in `PDB$SEED` cannot be added or modified.

3) Zero or more user-created PDBs

A PDB is a user-created entity that contains the data and code required for a specific set of features (typically an application).

In a multitenant database, each object is uniquely identified by a three-part name: `container.schema.object`. As an alternative to specifying three-part names, in Oracle GoldenGate parameter files, you can specify a default pluggable database with the `SOURCECATALOG` parameter, and then specify only the `schema.object` in subsequent `TABLE` or `SEQUENCE` parameters.

One Oracle GoldenGate Extract group can capture data from multiple pluggable databases to a single trail. However, Replicat can only connect and apply to one pluggable database. To specify the correct one, users must issue a SQL*Net connect string for the database user that is connecting to the specific pluggable database specified with the `USERID` or `USERIDALIAS` parameter.

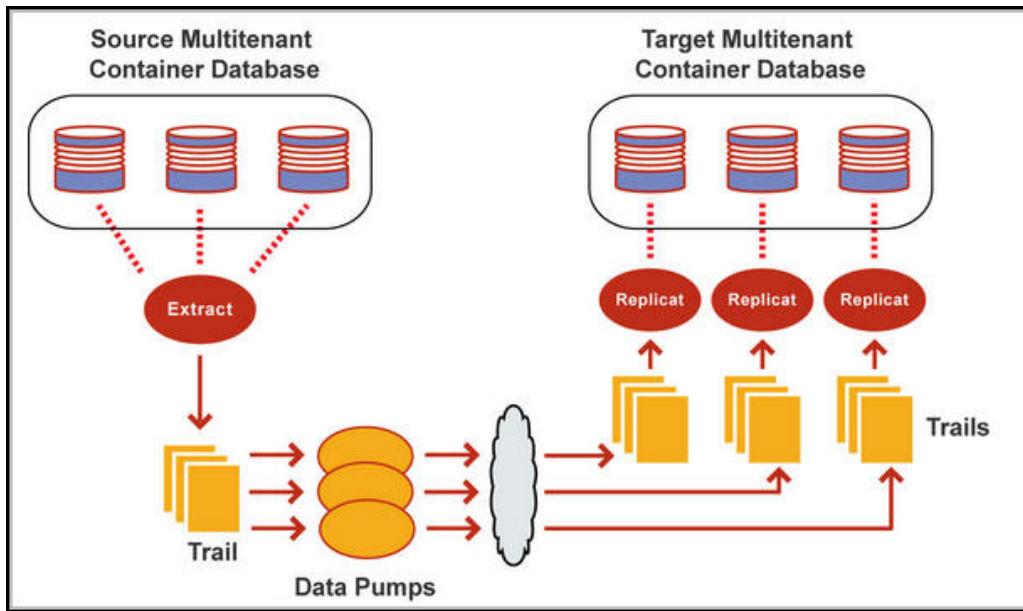


Figure 1: One Extract collects data from multiple PDBs in one single trail, but separate Replicat groups are needed on the data delivery side.

Multitenancy is only supported by Integrated Extract on the data capture side. Classic Replicat or Integrated Replicat can however be used for data delivery.

The practices for this course typically define Extract groups that connect to the root container and use the SOURCECATALOG parameter to narrow down the PDB required for data extraction. On the data delivery side, Integrated Replicat groups connect directly to the required PDB and MAP statements use the three-part names to link source objects to target objects.

Extract example:

```
Extract EUEVT
UseridAlias ogg_admin
...many lines omitted for clarity...
SOURCECATALOG ogg1
Table src_user.*;
```

The `ogg_admin` alias points to the `C##OGG_ADMIN@CDB12c` user (`CDB12c` is the root container, and `C##OGG_ADMIN` is a user defined in all containers). So `ogg_admin` connects to the root container, but `SOURCECATALOG ogg1` specifies that extraction occurs on the `ogg1` PDB.

Replicat example:

```
Replicat revtb
...many lines omitted for clarity...
UserIdAlias ogg_repl
map ogg1.src_user.gdp_by_year, target ogg2.trg_user.gdp_by_year;
```

The `ogg_repl` alias points to the `C##OGG_ADMIN@ogg2` user (`ogg2` is a pluggable database or PDB). Therefore, `ogg_repl` connects directly to the `ogg2` PDB hosted on the `CDB12c` container. The MAP statement uses three-part names to identify tables in the `ogg1` and `ogg2` PDBs.

Practice 2-1: Checking the Oracle Instance Parameters That Affect Replication

Overview

In this practice, you verify that your Oracle environment is appropriately set up for replication. Several database-level requirements must be met before you attempt to start the Oracle GoldenGate processes (Manager, Extract, and Replicat). The Oracle RDBMS used in this practice is CDB12c, which is a multitenant container database. Two pluggable databases, ogg1 and ogg2, which are created in the CDB12c container database store the data that must be replicated.

Assumptions

If instance-wide parameters and settings must be changed, you must use the `SYS` user.

Tasks

1. Open a terminal window as the `oracle` user.
2. Verify that the Oracle instance runs in `ARCHIVELOG` mode. If `LOG_MODE` is `ARCHIVELOG`, then skip step 3:

```
[OS prompt] export ORACLE_SID=CDB12c
[OS prompt] sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Mon Mar 9 13:46:04
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> select log_mode from v$database;

LOG_MODE
-----
ARCHIVELOG
```

3. If the Oracle instance is running in `NOARCHIVELOG` mode, you must change its running mode.

```
[OS prompt] sqlplus / as sysdba;
SQL*Plus: Release 12.1.0.2.0 Production on Mon Mar 9 13:46:04
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
```

```
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -  
64bit Production  
With the Partitioning, OLAP, Advanced Analytics and Real  
Application Testing options  
SQL> shutdown immediate  
Database closed.  
Database dismounted.  
ORACLE instance shut down.  
SQL> startup mount  
ORACLE instance started.  
  
Total System Global Area 2181038080 bytes  
Fixed Size          3713000 bytes  
Variable Size       1275070488 bytes  
Database Buffers    889192448 bytes  
Redo Buffers        13062144 bytes  
Database mounted.  
  
SQL> alter database archivelog;  
Database altered.  
  
SQL> alter database open;  
Database altered.
```

4. You must check that supplemental logging has been enabled. If SUPPLEMENTAL_LOG_DATA_MIN is YES, then skip step 5:

```
SQL> SELECT SUPPLEMENTAL_LOG_DATA_MIN, FORCE_LOGGING FROM  
V$DATABASE;  
  
SUPPLEMENTAL FORCE_LOGGING  
-----  
YES      YES  
SQL>
```

5. If the system displays NO, you must enable supplemental logging and force logging:

```
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;  
Database altered.  
SQL> ALTER DATABASE FORCE LOGGING;  
Database altered.  
SQL> ALTER SYSTEM SWITCH LOGFILE;  
System altered.
```

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

Practice 2-2: Preparing the Oracle Source Database

Overview

In this practice, you perform the preliminary steps that are necessary to prepare the Oracle source schema for replication. You must create the database objects that are being replicated on the source pluggable database.

Your system provides a container database (`CDB12c`) and two pluggable databases (`ogg1` and `ogg2`). You must create a common Oracle GoldenGate user (`c##ogg_admin`) that is able to access all CDB databases. Oracle GoldenGate will connect to `c##ogg_admin` to perform the integrated Extract, accessing data residing in the `ogg1` database, `src_user` schema.

Assumptions

The `ogg_user` user has access to the Oracle `sqlplus` utility and belongs to the `dba` group. The container database (`CDB12c`) and the `ogg1` and `ogg2` pluggable databases are running.

Tasks

1. Open a bash shell by using the Gnome terminal while connected to the workstation as "oracle." Set the `ORACLE_SID` variable to point to the `CDB12c` database and access `CDB12c` as `sysdba`.

```
[OS prompt] export ORACLE_SID=CDB12c
[OS prompt] sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Mon Mar 9 13:46:04
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL>
```

2. Create the common user `c##ogg_admin` and grant the `dba` privilege to it. Additionally, run the `dbms_goldengate_auth.grant_admin_privilege` stored procedure to grant that user admin privileges for the Oracle GoldenGate subsystem. If uncertain, ask your instructor what the password for the `c##ogg_admin` user should be:

```

SQL> create user c##ogg_admin identified by <password> default
tablespace users container=all;
User created.

SQL> grant dba, cdb_dba, pdb_dba to C##OGG_ADMIN container=all;
Grant succeeded.

SQL> exec
dbms_goldengate_auth.grant_admin_privilege('C##OGG_ADMIN', CONTAINER=>'all');

PL/SQL procedure successfully completed.

SQL>

```

3. Exit sqlplus. Change the directory to ~/labs/Practice02/configs and invoke sqlplus to connect to the replication source database (ogg1). Create the oggdata tablespace at a PDB level, and then create the src_user user. You must give the newly created src_user the necessary privileges. Also, you must grant local privileges on the ogg1 PDB database to the C##OGG_ADMIN common user. Finally, connect to the pluggable database ogg1 as src_user and run the script that creates the source replication objects (oracle_source_table_creation.sql).

```

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
[OS prompt] cd ~/labs/Practice02/configs
[OS prompt] sqlplus sys/<password>@ogg1 as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Mon Mar 9 13:46:04
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> create tablespace oggdata datafile
' /u02/oradata/CDB12c/ogg1/oggdata01.dbf' size 1G extent
management local uniform size 512K;
Tablespace created.
SQL> alter pluggable database default tablespace oggdata;
Pluggable database altered.
SQL> create user src_user identified by <password> default
tablespace oggdata;
User created.
SQL> grant connect,resource,unlimited tablespace to src_user;
Grant succeeded.
SQL> connect src_user/<password>@ogg1
Connected.

```

```
SQL> set echo on
SQL> @oracle_source_table_creation.sql
SQL> create table economic_entity
  2 (
  3     entity_id integer not null,
  4     economic_entity varchar2(128) not null,
  5     continent varchar2(20),
  6     constraint ecentpk PRIMARY KEY (entity_id)
  7 ) tablespace oggdata;
Table created.

SQL>
SQL> create table gdp_by_year
  2 (
  3     entity_id      integer not null,
  4     gdp_year       integer not null,
  5     gdp_value      number(15,5) not null,
  6     constraint gdpyr PRIMARY KEY (entity_id,gdp_year)
  7 ) tablespace oggdata;;
Table created.

SQL>
SQL> create table gdp_growth_by_year
  2 (
  3     entity_id      integer not null,
  4     gdp_year       integer not null,
  5     gdp_value      number(15,5) not null,
  6     constraint gdpgyr PRIMARY KEY (entity_id,gdp_year)
  7 ) tablespace oggdata;;
Table created.

SQL> select table_name from user_tables;
TABLE_NAME
-----
ECONOMIC_ENTITY
GDP_BY_YEAR
GDP_GROWTH_BY_YEAR
```

4. The `~/labs/Practice02/configs` subdirectory contains the following files:

File Name	Contents
<code>oracle_source_table_creation.sql</code> <code>oracle_target_table_creation.sql</code>	DDL scripts that create all the required tables for the replication exercise for source schemas
<code>economic_entity.sql</code>	DML script that populates the <code>ECONOMIC_ENTITY</code> table
<code>gdp_by_year.sql</code>	DML script that populates the World GDP data by year (in absolute terms) from 2005 to 2007 (included)
<code>gdp_by_year_2008.sql</code>	DML script that populates the World GDP (in absolute terms) for the year 2008
<code>gdp_by_year_2009.sql</code>	DML script that populates the World GDP (in absolute terms) for the year 2009
<code>gdp_by_year_2010.sql</code>	DML script that populates the World GDP (in absolute terms) for the year 2010
<code>gdp_growth_by_year.sql</code>	DML script that populates the World GDP data by year (in terms of growth over the previous year) from 2005 to 2007 (included)
<code>gdp_growth_by_year_2008.sql</code>	DML script that populates the World GDP (in terms of growth over the previous year) for the year 2008
<code>gdp_growth_by_year_2009.sql</code>	DML script that populates the World GDP (in terms of growth over the previous year) for the year 2009
<code>gdp_growth_by_year_2010.sql</code>	DML script that populates the World GDP (in terms of growth over the previous year) for the year 2010
<code>oracle_drop_src_tables.sql</code> <code>oracle_drop_trg_tables.sql</code>	DDL scripts that drop the tables used in all the practices in the source or the target schema

Run three scripts to populate the `ECONOMIC_ENTITY`, `GDP_BY_YEAR`, and `GDP_GROWTH_BY_YEAR` tables that were created in step 3:

```
SQL> set echo off
SQL> @economic_entity.sql
 1 row created.
[many rows omitted...]
 1 row created.

SQL> @gdp_by_year.sql
 1 row created.
[many rows omitted...]
```

```

1 row created.

SQL> @gdp_growth_by_year.sql
1 row created.
[many rows omitted...]
1 row created.
SQL>

```

Note that because you are connected as `src_user` and the DML scripts did not specify a schema name, all source replication objects are created as `src_user.<object_name>`.

- The last step that is required to prepare the source database is to enable additional logging at the table level to allow the necessary information to be written in the Oracle Redo logs. Open a new terminal shell, invoke the `oggsrc` alias to navigate to the `/u03/ogg/ogg_src` directory, and launch the `GGSCI` utility:

```

[OS prompt] oggsrc
[OS prompt] ggsci
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
10:21:34
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
rights reserved.

GGSCI ([HOST]) >

```

To enable supplemental logging, you must first log in to the `OGG1` database as the privileged `C##OGG_ADMIN` user. Then, you can enter the `ADD SCHEMATRANSACTION` commands for the `src_user` schema:

```

GGSCI ([HOST]) > dblogin userid C##OGG_ADMIN@ogg1 password
<password>
Successfully logged into database OGG1.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG1) > ADD SCHEMATRANSACTION
src_user ALLCOLS
INFO      OGG-01788 SCHEMATRANSACTION has been added on schema
src_user.
INFO      OGG-01976 SCHEMATRANSACTION for scheduling columns has
been added on schema src_user.
INFO      OGG-01977 SCHEMATRANSACTION for all columns has been added
on schema src_user.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG1) > info trandata
src_user.*  

(useful recap of what you just did)

```

```
INFO    OGG-06480 Schema level supplemental logging, excluding
non-validated keys, is enabled on schema SRC_USER.

INFO    OGG-01981 Schema level supplemental logging is enabled
on schema SRC_USER for all columns.

Logging of supplemental redo log data is enabled for table
OGG1.SRC_USER.ECONOMIC_ENTITY.

Columns supplementally logged for table
OGG1.SRC_USER.ECONOMIC_ENTITY: CONTINENT, ECONOMIC_ENTITY,
ENTITY_ID.

Logging of supplemental redo log data is enabled for table
OGG1.SRC_USER.GDP_BY_YEAR.

Columns supplementally logged for table
OGG1.SRC_USER.GDP_BY_YEAR: ENTITY_ID, GDP_VALUE, GDP_YEAR.

Logging of supplemental redo log data is enabled for table
OGG1.SRC_USER.GDP_GROWTH_BY_YEAR.

Columns supplementally logged for table
OGG1.SRC_USER.GDP_GROWTH_BY_YEAR: ENTITY_ID, GDP_VALUE,
GDP_YEAR.

GGSCI (orahost) 4> quit
```

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

Practice 2-3: Preparing the Oracle Target Database

Overview

In this practice, you perform the preliminary steps that are necessary to prepare the Oracle target schema for replication. You must create the database objects being replicated on the target.

Assumptions

The `ogg_user` user has access to the Oracle `sqlplus` utility and belongs to the `dba` group, so that it can connect as a privileged user to the Oracle RDBMS `CDB12c`.

Tasks

1. Open a bash shell and change the directory to `~/labs/Practice02/configs`; invoke `sqlplus` to connect to the `ogg2` PDB database (the replication target). Create a PDB-level tablespace called `oggdata` and create the `trg_user` Oracle RDBMS user for the `ogg2` PDB database:

```
[OS prompt] cd ~/labs/Practice02/configs
[OS prompt] sqlplus sys/<password>@ogg2 as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Mon Mar 9 13:46:04
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> create tablespace oggdata datafile
'./u02/oradata/CDB12c/ogg2/oggdata01.dbf' size 500M extent
management local uniform size 512K;
Tablespace created.
SQL> alter pluggable database default tablespace oggdata;
Pluggable database altered.
SQL> create user trg_user identified by <password> default
tablespace oggdata;
User created.
SQL> grant connect,resource,unlimited tablespace to trg_user;
Grant succeeded.
SQL>
```

2. You need to create the objects being replicated in the replication target schema. Connect as `trg_user` and run the DDL script that creates the replication objects:

```
SQL> connect trg_user/<password>@ogg2
Connected.

SQL> set echo on
SQL> @oracle_target_table_creation.sql
```

```
SQL> create table economic_entity
  2 (
  3     entity_id integer not null,
  4     economic_entity varchar2(128) not null,
  5     continent varchar2(20),
  6     constraint ecentpk PRIMARY KEY (entity_id)
  7 ) tablespace oggdata;
Table created.

SQL> create table gdp_by_year
  2 ( entity_id      integer not null,
  3   gdp_year      integer not null,
  4   gdp_value    number(15,5) not null,
  5   constraint gdpyr PRIMARY KEY (entity_id,gdp_year)
  6 );
Table created.

SQL> create table gdp_growth_by_year
  2 ( entity_id      integer not null,
  3   gdp_year      integer not null,
  4   gdp_value    number(15,5) not null,
  5   constraint gdpgyr PRIMARY KEY (entity_id,gdp_year)
  6 );
Table created.

SQL> SELECT * FROM cat;

TABLE_NAME          TABLE_TYPE
-----
ECONOMIC_ENTITY      TABLE
GDP_BY_YEAR         TABLE
GDP_GROWTH_BY_YEAR  TABLE

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
[OS prompt]
```

Note that because you are connected as `trg_user` and the DDL scripts did not specify a schema name, all target replication objects are created as `trg_user.<object name>`.

Checking the Database Parameters

3. It is important to verify that the underlying Oracle instance has been adequately sized, from a memory standpoint. The Integrated Capture mechanism uses memory carved out from the `streams_pool_size` area of the SGA, and you need to make sure that `streams_pool_size` is large enough to accommodate the memory request that the Extract process will submit in order to allocate an instance for itself. Open yet another shell window and use `sqlplus` to connect to the CDB12c RDBMS instance by using a privileged account (`sys as sysdba`).

```
[OS prompt]$ export ORACLE_SID=CDB12c
[OS prompt]$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Thu Mar 12 22:03:18
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL>
```

4. The information available from the V\$ view can help determine if the instance has been appropriately configured for Integrated Capture:

NAME	BYTES	RESIZABLE	CON_ID
Fixed SGA Size	2925504	No	0
Redo Buffers	13848576	No	0
Buffer Cache Size	754974720	Yes	0
In-Memory Area Size	0	No	0
Shared Pool Size	251658240	Yes	0
Large Pool Size	33554432	Yes	0
Java Pool Size	16777216	Yes	0
Streams Pool Size	838860800	Yes	0
Shared IO Pool Size	50331648	Yes	0
Data Transfer Cache Size	0	Yes	0
Granule Size	16777216	No	0
Maximum SGA Size	1811939328	No	0
Startup overhead in Shared Pool	143599736	No	0
Free SGA Memory Available	0		0
12 rows selected.			

```
SQL>
```

Because the SGA areas are allocated dynamically, your values may be different from the ones shown above. However, the output shows that the `streams_pool_size` parameter has been allocated approximately 800 megabytes of memory, which is sufficient for an

Integrated Capture configuration. Your value for streams_pool_size should also be approximately 800 megabytes. If your environment shows less than 800 megabytes allocated to the stream pool, you can always rectify the situation by issuing the following command while connected as sys to the CDB12c instance:

```
SQL> alter system set streams_pool_size = '800M' scope = both;  
System altered.
```

Also check that the ENABLE_GOLDENGATE_REPLICATION parameter is set to TRUE.

```
SQL> show parameter enable_goldengate_replication
```

NAME	TYPE	VALUE
enable_goldengate_replication	Boolean	TRUE

If not, set that parameter to TRUE:

```
SQL> alter system set enable_goldengate_replication = TRUE scope  
= both;  
System altered.
```

This completes Practice 2-3. Continue with Practice 2-4.

Practice 2-4: Preparing the Oracle GoldenGate Environment

Overview

In this practice, you configure the manager process for the replication source (`ogg_src`) and the replication target (`ogg_trg`) environments.

Assumptions

The two directories `/u03/ogg/ogg_src` and `/u03/ogg/ogg_trg` exist, and the Oracle GoldenGate software has been installed in those directories.

Tasks

1. Start a new shell window. Use the `oggsr` alias to navigate to the `/u03/ogg/ogg_src` directory and launch `GGSCI`. Stop the manager and edit the parameter file for the manager. If the Oracle GoldenGate process manager is not running, you receive the message "Manager already stopped" which you can safely disregard. Make sure that you are using port 7809, and set other relevant parameters:

```
[OS prompt]$ oggsr
[OS prompt]$ pwd
/u03/ogg/ogg_src
[OS prompt]$ ggsci
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
10:21:34
Operating system character set identified as UTF-8.
Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
rights reserved.

GGSCI ([SRCHOST]) 1> stop mgr!
Sending STOP request to MANAGER ...
Request processed.

Manager stopped.

GGSCI ([SRCHOST]) 2> edit param mgr
```

```
Port 7809
DynamicPortList 20000-20099
PurgeOldExtracts ./dirdat/*, UseCheckPoints, MinKeepHours 2
Autostart Extract E*
AUTORESTART Extract *, WaitMinutes 1, Retries 3
```

2. Save the parameter file for the manager and start the Oracle GoldenGate manager process for the `ogg_src` environment:

```
GGSCI ([SRCHOST]) 3> start mgr
Manager started.

GGSCI ([SRCHOST]) 3> info mgr detail
Manager is running (IP port [SRCHOST].7809).
```

3. Exit GGSCI and use the `oggtrg` alias to navigate to the `/u03/ogg/ogg_trg` directory. Launch GGSCI. Stop the manager and edit the parameter file for the manager. If the Oracle GoldenGate process manager is not running, you receive the message “Manager already stopped” which you can safely disregard. Make sure that you are using port 7909, and set other relevant parameters:

```
GGSCI ([TRGHOST]) 2> exit  
[OS prompt]$ oggtrg  
[OS prompt]$ pwd  
/u03/ogg/ogg_trg  
[OS prompt]$ ggsci  
Oracle GoldenGate Command Interpreter for Oracle  
Version 12.1.2.1.0  
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO  
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014  
10:21:34  
Operating system character set identified as UTF-8.
```

Copyright (C) 1995, 2014, Oracle and/or its affiliates. All rights reserved.

```
GGSCI ([TRGHOST]) 1> stop mgr!  
Sending STOP request to MANAGER ...  
Request processed.  
Manager stopped.
```

```
GGSCI ([TRGHOST]) 1> edit param mgr
```

```
Port 7909  
DynamicPortList 20100-20199  
PurgeOldExtracts ./dirdat/pe*, UseCheckPoints, MinKeepHours 2  
Autostart Replicat R*  
AUTORESTART Replicat *, WaitMinutes 1, Retries 3
```

4. Save the parameter file for the manager and start the manager, verifying that the parameter file was correctly interpreted by GGSCI:

```
GGSCI ([TRGHOST]) 2> start mgr  
Manager started.  
GGSCI ([TRGHOST]) 3> info mgr detail  
Manager is running (IP port [TRGHOST].7909).
```

This completes Practice 2-4. Continue with Practice 2-5.

Practice 2-5: Set Up Replication

Overview

In this practice, you register and create the Integrated Extract group `einta`, the Data Pump `pinta`, and the Integrated Replicat `pinta`. You then insert data in the source PDB replication database (`ogg1.src_user`) and make sure that the data is replicated to the target PDB (`ogg2.trg_user`).

Assumptions

Practices 2-1 to 2-4 have been successfully completed, and both replication source and replication target environments are correctly set. The `CDB12c` database instance has been checked to assess if the necessary memory has been allocated to the stream pool.

Tasks

Configuring the Integrated Extract

1. Open a new terminal window, or reuse one that is already open, as long as it is connected as the "oracle" user. Use the `oggsrc` alias to navigate to the `/u03/ogg/ogg_src` directory, set the `ORACLE_SID` environment variable to `CDB12c`, and launch `GGSCI`. Create the parameter file for the `einta` Integrated Extract:

```
[OS prompt]$ oggsrc
[OS prompt]$ pwd
/u03/ogg/ogg_src
[OS prompt]$ export ORACLE_SID=CDB12c
[OS prompt]$ ggsci
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
10:21:34
Operating system character set identified as UTF-8.
Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
rights reserved.
GGSCI (<hostname>) 1> edit param einta
```

```
EXTRACT einta
SETENV (ORACLE_SID='CDB12c')
USERID C##OGG_ADMIN password <password>
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
EXTTRAIL ./dirdat/in
SOURCECATALOG ogg1
TABLE src_user.*;
```

- Save the file and, when you are back in GGSCI, connect to the database by using DBLOGIN and enter the register command, which denotes an Integrated Extract:

```
GGSCI (<hostname>) 2> DBLOGIN USERID C##OGG_ADMIN password  
<password>  
Successfully logged into database CDB$ROOT.  
GGSCI (<hostname>) 3> REGISTER EXTRACT einta DATABASE CONTAINER  
(ogg1)  
Extract EINTA successfully registered with database at SCN  
2160172.
```

- Your SCN will vary, but the output should show you the SCN valid for your environment. You can now add the Integrated Extract and connect it to its exttrail file, sizing it to 10 megabytes:

```
GGSCI ([host]) 4> add extract einta, integrated tranlog, begin  
now  
EXTRACT added.  
GGSCI ([host]) 5> add exttrail ./dirdat/in, extract einta,  
megabytes 10  
EXTTRAIL added.
```

- Define a Data Pump Extract:

```
GGSCI ([host]) 7> edit param pinta
```

```
EXTRACT pinta  
SETENV (ORACLE_SID='CDB12c')  
USERID C##OGG_ADMIN password <password>  
RMTHOST ogg_target, MGRPORT 7909  
RMTTRAIL ./dirdat/pn  
SOURCECATALOG ogg1  
TABLE src_user.*;
```

- Save the Data Pump parameter file for PINTA and, when you are back at the GGSCI prompt, add PINTA together with its local exttrail file. Then add the remote trail, linking it to the PINTA Data Pump:

```
GGSCI ([host]) 8> Add extract pinta, exttrailsource ./dirdat/in  
EXTRACT added.  
GGSCI ([host]) 9> Add rmttrail ./dirdat/pn, extract pinta,  
megabytes 10  
RMTTRAIL added.
```

6. Start both Extract processes, and verify their status by entering INFO ALL:

```
GGSCI (<host> as C##OGG_ADMIN@CDB12c/CDB$ROOT) 10> start extract EINTA
Sending START request to MANAGER ...
EXTRACT EINTA starting
GGSCI ((<host> as C##OGG_ADMIN@CDB12c/CDB$ROOT) 10> start extract PINTA
Sending START request to MANAGER ...
EXTRACT PINTA starting
GGSCI (<host> as C##OGG_ADMIN@CDB12c/CDB$ROOT) 11> info all

Program      Status      Group      Lag at Chkpt  Time Since
Chkpt

MANAGER      RUNNING
EXTRACT      RUNNING    EINTA      00:00:03    00:00:08
EXTRACT      RUNNING    PINTA      00:00:00    00:04:50
```

7. The Integrated Capture was started successfully. Investigate further, requesting GGSCI to produce a report on the EINTA Extract:

```
GGSCI (<host> as C##OGG_ADMIN@CDB12c/CDB$ROOT) 15> view report einta
*****
*
          Oracle GoldenGate Capture for Oracle
          Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
          Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
          10:40:21
          Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
          rights reserved.

          Starting at 2015-03-12 23:31:11
*****
*
Operating System Version:
Linux
Version #2 SMP Wed Jan 28 17:03:28 PST 2015, Release 3.8.13-
55.1.5.el6uek.x86_64
Node: host01.localdomain
Machine: x86_64
--More-- (20%)
```

8. In the einta report, look for the message:

```
Integrated capture successfully attached to logmining server
OGG$CAP_EINTA using OGGCapture API.
```

It informs you that the Integrated Extract successfully connected to the logmining server.

Configuring the Replicat

9. Open a new terminal window, or reuse one that is already open, as long as it is connected as the "oracle" user. Use the `oggtrg` alias to navigate to the `/u03/ogg/ogg_trg` directory, set the `ORACLE_SID` environment variable to `CDB12c`, and launch `GGSCI`. Create the parameter file for the `rinta` Integrated Replicat:

```
[OS prompt]$ oggtrg
[OS prompt]$ pwd
/u03/ogg/ogg_trg
[OS prompt]$ export ORACLE_SID=CDB12c
[OS prompt]$ ggsci
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
10:21:34
Operating system character set identified as UTF-8.
Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
rights reserved.
GGSCI (<hostname>) 1> edit param rinta
```

```
REPLICAT rinta
SETENV (ORACLE_SID='CDB12c')
DBOPTIONS INTEGRATEDPARAMS(parallelism 4)
USERID C##OGG_ADMIN@ogg2, PASSWORD <password>
ASSUMETARGETDEFS
MAP ogg1.src_user.* , TARGET ogg2.trg_user.*;
```

10. Save the `RINTA` parameter file and, when you are back in `GGSCI`, connect to the `ogg2` RDBMS and add `RINTA`, linking it to its `extrail` file. Then start the Replicat and assess the situation by issuing the `INFO ALL` command. Note the `INTEGRATED` keyword before `EXTTRAIL`, which denotes this as an Integrated Replicat:

```
GGSCI (<hostname>) 2> dblogin USERID C##OGG_ADMIN@ogg2, PASSWORD
<password>
Successfully logged into database OGG2.
GGSCI (<host> as C##OGG_ADMIN@CDB12c/OGG2) 3> add replicat
rinta, INTEGRATED EXTTRAIL ./dirdat/pn
REPLICAT (Integrated) added.
GGSCI (<host> as C##OGG_ADMIN@CDB12c/OGG2) 4> start replicat
rinta
Sending START request to MANAGER ...
REPLICAT RINTA starting
```

GGSCI (<host> as C##OGG_ADMIN@CDB12c/OGG2) 5> info all					
Program Chkpt	Status	Group	Lag at Chkpt	Time Since	
MANAGER	RUNNING				
REPLICAT	RUNNING	RINTA	00:00:00	00:13:23	

Generating database activity to test replication

11. Open a Gnome terminal window, or reuse the shell that you previously accessed to verify the database parameters (in this case, exit sqlplus) and navigate to the ~/labs/Practice02/configs directory. Launch sqlplus to connect to the ogg1 (src_user) replication source database and generate some activity on the GDP_BY_YEAR table:

```
[OS prompt]$ export ORACLE_SID=CDB12c
[OS prompt]$ cd ~/labs/Practice02/configs
[OS prompt]$ sqlplus src_user/<password>@ogg1
SQL*Plus: Release 12.1.0.2.0 Production on Fri Mar 13 00:23:24
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Mon Mar 09 2015 23:35:21 +11:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> @gdp_by_year_2008.sql
1 row created.

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

12. Connect to the target database/schema and verify that the rows have been replicated for the GDP_BY_YEAR table:

```
SQL> connect trg_user/<password>@ogg2
Connected.
SQL> select count(*) from gdp_by_year where gdp_year=2008;

COUNT(*)
-----
235
```

The count for the rows (235) confirms that the Integrated Capture worked, and the Data Pump and Integrated Replicat received the traffic generated on the source schema.

13. Select the window where GGSCI is connected to the OGG_SRC Oracle GoldenGate instance. Enter the stats einta command to display the statistics for the EINTA Extract group. Then enter the stats pinta command to display the statistics for the PINTA Extract group:

```
GGSCI (<host> as C##OGG_ADMIN@CDB12c/CDB$ROOT) 16> stats einta
Sending STATS request to EXTRACT EINTA ...
Start of Statistics at 2015-03-13 00:34:04.
Output to ./dirdat/in:
Extracting from OGG1.SRC_USER.GDP_BY_YEAR to
OGG1.SRC_USER.GDP_BY_YEAR:
*** Total statistics since 2015-03-13 00:26:47 ***
    Total inserts                      235.00
    Total updates                       0.00
    Total deletes                       0.00
    Total discards                      0.00
    Total operations                   235.00
*** Daily statistics since 2015-03-13 00:26:47 ***
    Total inserts                      235.00
    Total updates                       0.00
    Total deletes                       0.00
    Total discards                      0.00
    Total operations                   235.00
*** Hourly statistics since 2015-03-13 00:26:47 ***
    Total inserts                      235.00
    Total updates                       0.00
    Total deletes                       0.00
    Total discards                      0.00
    Total operations                   235.00
*** Latest statistics since 2015-03-13 00:26:47 ***
    Total inserts                      235.00
    Total updates                       0.00
    Total deletes                       0.00
    Total discards                      0.00
    Total operations                   235.00
End of Statistics.

GGSCI (<host> as C##OGG_ADMIN@CDB12c/CDB$ROOT) 17> stats pinta
Sending STATS request to EXTRACT PINTA ...
Start of Statistics at 2015-03-13 00:35:53.
Output to ./dirdat/pn:
Extracting from OGG1.SRC_USER.GDP_BY_YEAR to
OGG1.SRC_USER.GDP_BY_YEAR:
*** Total statistics since 2015-03-13 00:26:49 ***
    Total inserts                      235.00
```

Total updates	0.00
Total deletes	0.00
Total discards	0.00
Total operations	235.00
*** Daily statistics since 2015-03-13 00:26:49 ***	
Total inserts	235.00
Total updates	0.00
Total deletes	0.00
Total discards	0.00
Total operations	235.00
*** Hourly statistics since 2015-03-13 00:26:49 ***	
Total inserts	235.00
Total updates	0.00
Total deletes	0.00
Total discards	0.00
Total operations	235.00
*** Latest statistics since 2015-03-13 00:26:49 ***	
Total inserts	235.00
Total updates	0.00
Total deletes	0.00
Total discards	0.00
Total operations	235.00

End of Statistics.

14. Select the window where the GGSCI session is connected to the OGG_TRG Oracle GoldenGate instance. Enter the stats rinta command to display the statistics for the RINTA Replicat group:

```
GGSCI (<host> as C##OGG_ADMIN@CDB12c/OGG2) 11> stats rinta
Sending STATS request to REPLICAT RINTA ...
Start of Statistics at 2015-03-13 00:36:02.

Integrated Replicat Statistics:
  Total transactions           1.00
  Redirected                   0.00
  DDL operations               0.00
  Stored procedures             0.00
  Datatype functionality       0.00
  Event actions                0.00
  Direct transactions ratio    0.00%
Replicating from OGG1.SRC_USER.GDP_BY_YEAR to
OGG2.TRG_USER.GDP_BY_YEAR:
*** Total statistics since 2015-03-13 00:26:52 ***
  Total inserts                 235.00
  Total updates                 0.00
  Total deletes                 0.00
  Total discards                0.00
  Total operations              235.00
*** Daily statistics since 2015-03-13 00:26:52 ***
  Total inserts                 235.00
  Total updates                 0.00
  Total deletes                 0.00
  Total discards                0.00
  Total operations              235.00
*** Hourly statistics since 2015-03-13 00:26:52 ***
  Total inserts                 235.00
  Total updates                 0.00
  Total deletes                 0.00
  Total discards                0.00
  Total operations              235.00
*** Latest statistics since 2015-03-13 00:26:52 ***
  Total inserts                 235.00
  Total updates                 0.00
  Total deletes                 0.00
  Total discards                0.00
  Total operations              235.00
End of Statistics.
```

This last step completes Practice 2-5.

This completes Practice 2 for *Oracle GoldenGate 12c Advanced Configuration for Oracle*.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

Practices for Lesson 3: Integrated Capture Deployment and Required Components

Chapter 3

Practices for Lesson 3: Overview

Practices Overview

In these practices, you will explore the advanced features of the Integrated Capture facility, which is provided by Oracle GoldenGate 12.1.0.2. Integrated Capture was first available in Oracle RDBMS version 11.2.0.3.

In the previous practice, you created an Integrated Extract group running straight from the Oracle RDBMS instance CDB12c. In these practices, you will use the Oracle GoldenGate Wallet facility to avoid providing passwords in clear text in parameter files and GGSCI sessions. You will also configure log shipping and create an Extract group that uses the real-time downstream deployment option.

Helper scripts provided in /home/oracle/labs/Practice03/configs:

Catch-up/clean-up script	Purpose
catch-up-prac-3_1.sh	Performs all steps for Practice 3-1
catch-up-prac-3_2.sh	Performs all steps for Practice 3-2
catch-up-prac-3_3.sh	Performs all steps for Practice 3-3
clean-up-prac-3_1.sh	Undoes all steps performed by Practice 3-1
clean-up-prac-3_2.sh	Undoes all steps performed by Practice 3-2
clean-up-prac-3_3.sh	Undoes all steps performed by Practice 3-3
clean-up-database.sh	Removes Oracle Data Guard configuration and restores the configuration of the CDB12c and DWNSTR databases as they were at the beginning of Practice 3

Note: You should either run the `clean-up-database.sh` script or perform the reconfiguration manually, as instructed in the "Resetting the database instances to their default parameters" section (step 24 in Practice 3-3) before continuing with Practice 4. The `clean-up-database.sh` script should be run only once, just before moving on to Practice 4.

Practice 3-1: Using the Oracle GoldenGate Wallet Facility

Overview

To avoid storing passwords in clear text in the various Extract and Replicat parameter files, Oracle GoldenGate 12c offers the new Wallet facility. You can store encrypted credentials in the Wallet credential store and refer to them through an alias, rather than by using the username/password combination.

Tasks

1. Open a new terminal window or reuse one that is already open if you are connected as the `oracle` user. Use the `oggsrc` alias to navigate to the `/u03/ogg/ogg_src` directory, set the `ORACLE_SID` environment variable to `CDB12c`, and launch `GGSCI`.

```
[OS prompt]$ oggsrc
[OS prompt]$ pwd
/u03/ogg/ogg_src
[OS prompt]$ export ORACLE_SID=CDB12c
[OS prompt]$ ggsci
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
10:21:34
Operating system character set identified as UTF-8.
Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
rights reserved.
GGSCI (<hostname>) 1>
```

2. Create the Wallet, add a credential store to it, and add the user credentials for the `C##OGG_ADMIN` user in the `CDB12c` database and the `C##OGG_DWNADM` user in the `DWNSTR` database. You will create the `C##OGG_DWNADM` user in a subsequent step. Additionally, create user credentials for `C##OGG_ADMIN@ogg2`, which is the user that is used by the Integrated Replicat to connect to the replication target schema.

```
GGSCI (<host>) > Create Wallet
Created wallet at location 'dirwlt'.
Opened wallet at location 'dirwlt'.
GGSCI (<host>) > Add CredentialStore
Credential store created in ./dircrd/.
GGSCI (<host>) > Alter CredentialStore Add User
C##OGG_ADMIN@CDB12c Password <password> Alias ogg_admin
Credential store in ./dircrd/ altered.

GGSCI (<host>) > Alter CredentialStore Add User
C##OGG_DWNADM@DWNSTR Password <password> Alias ogg_dwnadm
Credential store in ./dircrd/ altered.
GGSCI (<host>) > Alter CredentialStore Add User
C##OGG_ADMIN@ogg2 Password <password> Alias ogg_repl
```

```
GGSCI (<host>) > Info CredentialStore
Reading from ./dircrd/:
Default domain: OracleGoldenGate
  Alias: ogg_admin
  Userid: C##OGG_ADMIN@CDB12c
  Alias: ogg_dwnadm
  Userid: C##OGG_DWNADM@DWNSTR
  Alias: ogg_repl
  Userid: C##OGG_ADMIN@ogg2
GGSCI (<host>) >
```

3. Make sure the C##OGG_ADMIN Oracle GoldenGate user can connect to the Oracle database by using a credential alias and exit the GGSCI utility.

```
GGSCI (<host>) > DBLogin UserIdAlias ogg_admin
Successfully logged into database CDB$ROOT.
GGSCI (<host> as C##OGG_ADMIN@CDB12c/CDB$ROOT) 7> exit
[OS prompt]$
```

4. At the OS prompt, you must copy the Wallet and the credential files from the ogg_src directory to the ogg_trg directory. Use the oggtrg alias to navigate to the target replication environment and copy the files:

```
[OS prompt]$ oggtrg
[OS prompt]$ pwd
/u03/ogg/ogg_trg
[OS prompt]$ cp /u03/ogg/ogg_src/dircrd/* ./dircrd
[OS prompt]$ ls ./dircrd
cwallet.sso
[OS prompt]$ cp /u03/ogg/ogg_src/dirwlt/* ./dirwlt
[OS prompt]$ ls ./dirwlt
cwallet.sso
[OS prompt]$
```

5. Launch GGSCI in the replication target environment, display the entries in the credential store to make sure that they have been successfully copied, and then connect to the database by using the ogg_admin alias:

```
[OS prompt]$ ggsci
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
10:21:34
Operating system character set identified as UTF-8.
Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
rights reserved.
GGSCI (<host>) 1> Info CredentialStore
Reading from ./dircrd/:
```

```
Default domain: OracleGoldenGate
  Alias: ogg_admin
  Userid: C##OGG_ADMIN@CDB12c
  Alias: ogg_dwnadm
  Userid: C##OGG_DWNADM@DWNSTR
  Alias: ogg_repl
  Userid: C##OGG_ADMIN@ogg2
GGSCI (<host>) 2> DBLogin UserIdAlias ogg_admin
Successfully logged into database CDB$ROOT.
GGSCI (<host> as C##OGG_ADMIN@CDB12c/CDB$ROOT) 3>
```

6. Exit GGSCI.

```
GGSCI (<host> as C##OGG_ADMIN@CDB12c/CDB$ROOT) 3> exit
[OS prompt] $
```

7. Exit sqlplus and GGSCI and close all terminal windows. You need a fresh start in the next practice, which requires an uncluttered screen to avoid mistakes.

This completes Practice 3-1. Continue with Practice 3-2.

Practice 3-2: Preparing Source and Downstream Database for Integrated Capture

Overview

In this practice, you prepare the source and the downstream mining database for an Integrated Capture deployment. There are several DBA-level steps that must be followed in order to have the source database ship the Redo log files to the downstream mining database. It is crucial to follow the sequence shown in the following steps in order to achieve a working environment.

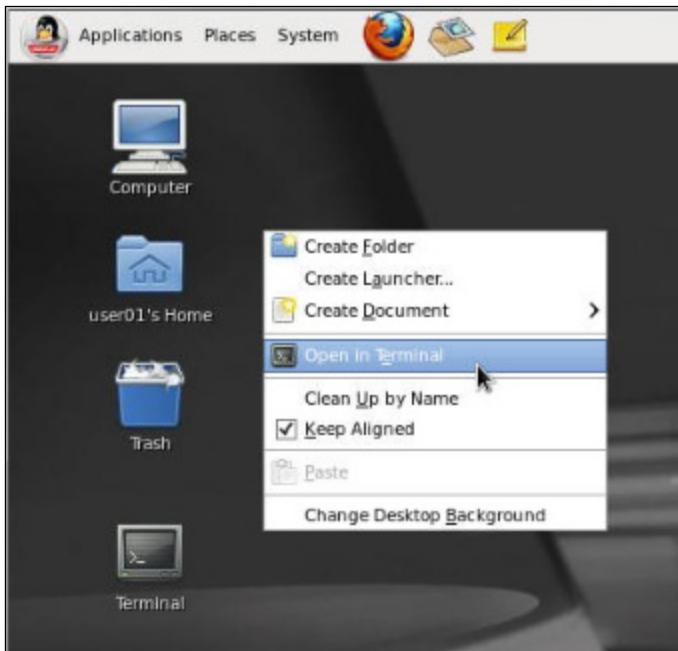
Assumptions

The CDB12C and DWNSTR databases are available in the Oracle environment. You have access to the `oracle` software owner account. By now, you should know the password for that account. If in doubt, ask your instructor before starting the practice.

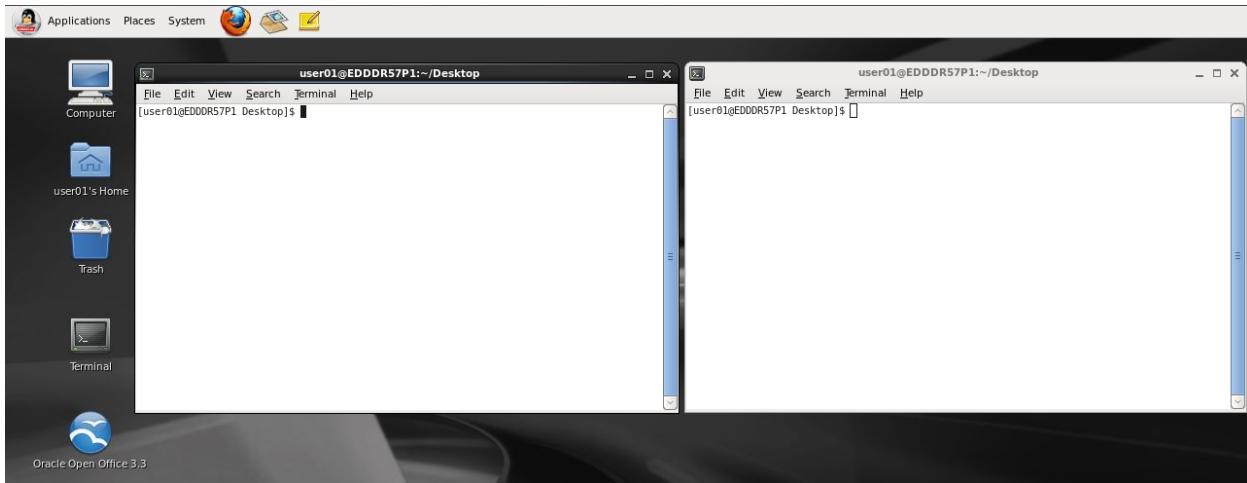
Tasks

Creating Gnome windows for Oracle GoldenGate environments

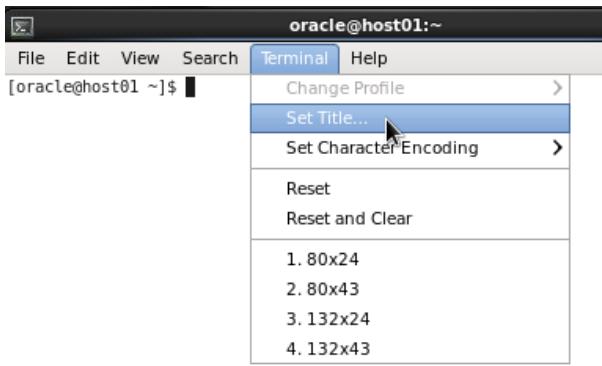
1. You should be connected to your workstation as "oracle." The main Gnome background should show only a few icons on the leftmost part of your screen and no other terminal window should be open. Position your mouse pointer somewhere in the main Gnome background and right-click. In the shortcut menu, click Open in Terminal to open a terminal window. Repeat the process to open another terminal window:



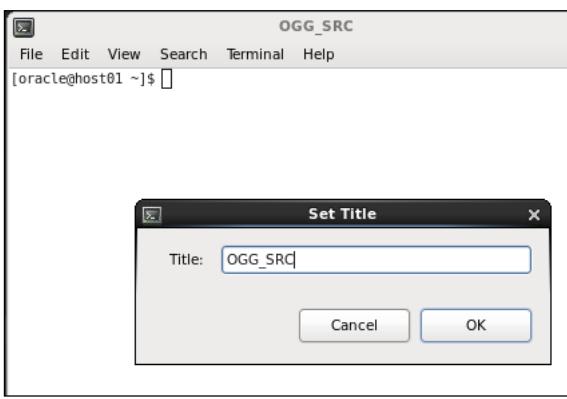
2. Drag the second window to align it next to the first one:



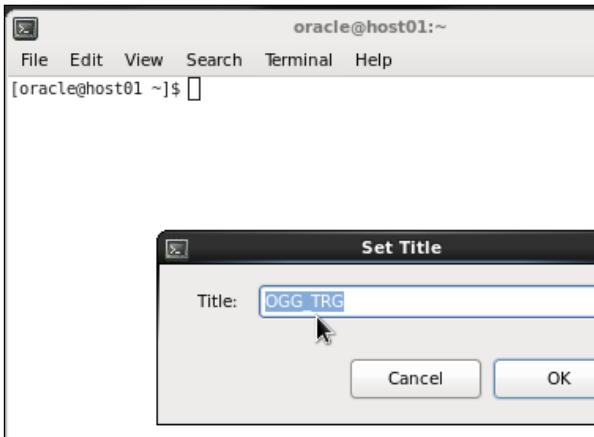
3. Click the border of the window on the left to make it active, click Terminal, and then select the Set Title option from the menu:



4. Name the window OGG_SRC. You will run the Oracle GoldenGate GGSCI environment that is connected to the source database in this window:



5. Select the window on the right, click Terminal, select Set Title, and enter OGG_TRG as the title for the window. Here, you will run the Oracle GoldenGate GGSCI environment that is connected to the target database.



Verifying source and target Oracle GoldenGate manager configuration

6. Click the OGG_SRC window, navigate to the /u03/ogg/ogg_src directory, and launch GGSCI. Edit the parameter file for the manager and make sure that you are using port 7809:

```
[OS prompt]$ oggsrc
[OS prompt]$ pwd
/u03/ogg/ogg_src
[OS prompt]$ ggsci
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
10:21:34
Operating system character set identified as UTF-8.
Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
rights reserved.

GGSCI ([Host]) 1> view param mgr
Port 7809
DynamicPortList 20000-20099
PurgeOldExtracts ./dirdat/*, UseCheckPoints, MinKeepHours 2
Autostart Extract E*
AUTORESTART Extract *, WaitMinutes 1, Retries 3
```

7. Start the manager process if it is not already running. Note that there is no problem if you try to start the manager and it is already running. A simple message will inform you that the manager is already running:

```
GGSCI ([Host]) 2> start mgr
Manager started.
```

8. Select the OGG_TRG window and navigate to the /u03/ogg/ogg_trg directory. Launch GGSCI and make sure that the target instance uses port 7909:

```
[OS prompt]$ oggtrg
[OS prompt]$ pwd
/u03/ogg/ogg_trg
[OS prompt]$ ggsci
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
10:21:34
Operating system character set identified as UTF-8.
Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
rights reserved.
GGSCI ([Host]) 1> view param mgr
Port 7909
DynamicPortList 20100-20199
PurgeOldExtracts ./dirdat/pe*, UseCheckPoints, MinKeepHours 2
Autostart Replicat R*
AUTORESTART Replicat *, WaitMinutes 1, Retries 3
GGSCI ([Host]) 2> start mgr
Manager started.
```

Stopping and deleting Extract and Replicat groups

9. Select the OGG_SRC window, and stop and delete the two Extract groups PINTA and EINTA. Then log in to the CDB12c database by using the Wallet alias and unregister the EINTA Integrated Extract, but leave the manager running:

```
GGSCI ([host]) > stop er *
Sending STOP request to EXTRACT PINTA ...
Request processed.

Sending STOP request to EXTRACT EINTA ...
Request processed.

GGSCI ([host]) > delete er *
Are you sure you want to delete all groups? y
Deleted EXTRACT EINTA.
Deleted EXTRACT PINTA.

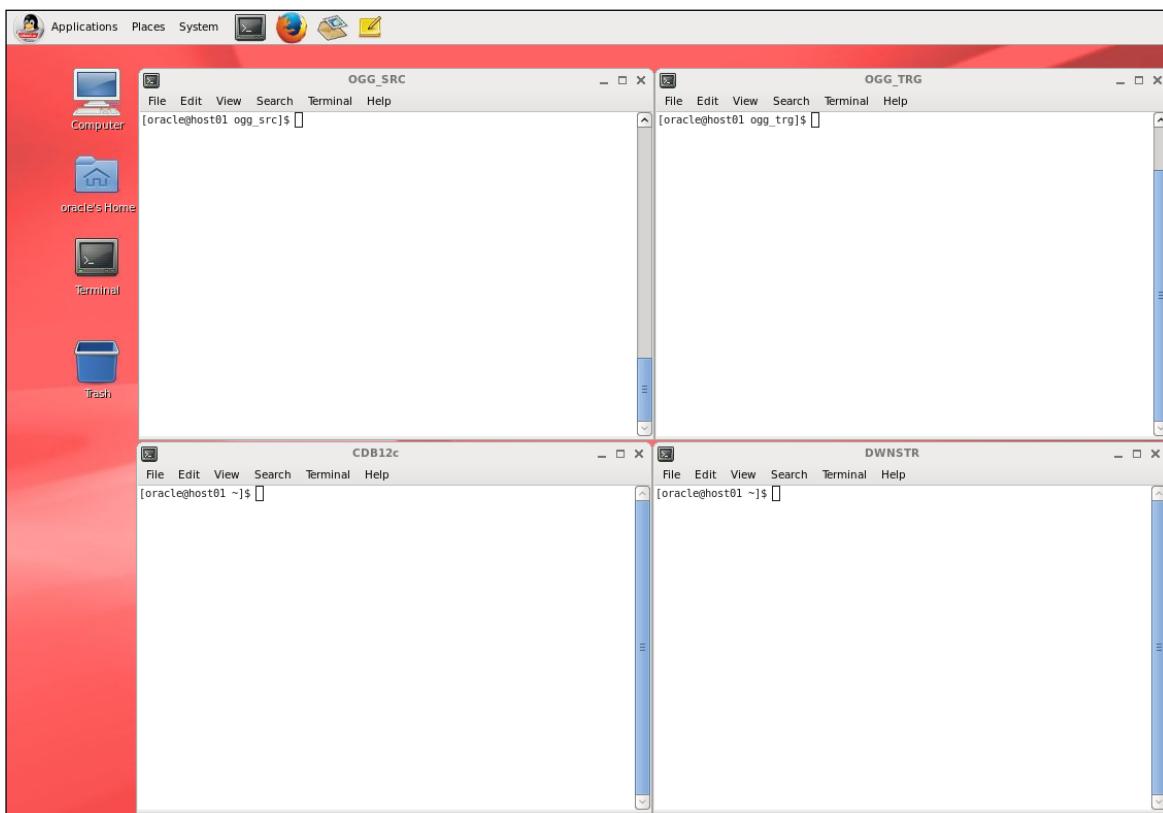
GGSCI ([host]) > DBLogin UserIdAlias ogg_admin
Successfully logged into database CDB$ROOT.
GGSCI ([host] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > unregister
extract einta database
Successfully unregistered EXTRACT EINTA from database.
```

10. Select the OGG_TRG window and log in to the CDB12c database by using the Wallet alias. Stop and delete the RINTA Replicat group:

```
GGSCI ([host]) > DBLogin UserIdAlias ogg_admin
Successfully logged into database CDB$ROOT.
GGSCI ([host] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop rep rinta
Sending STOP request to REPLICAT RINTA ...
Request processed.
GGSCI ([host] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > delete rep
rinta
Deleted REPLICAT RINTA.
```

Creating password files for both instances

11. Open two Gnome terminal windows, and set the title for the first window to CDB12c and the second window to DWNSTR. Align the two newly created windows below the other two windows OGG_SRC and OGG_TRG:



The screenshot above shows how your Gnome environment should look. The four windows should be aligned with each other. At the top, there should be the GGSCI sessions connected to the source and target replication environments, and at the bottom, there should be the OS sessions connected as oracle, where you will perform most of the DBA operations to set up the downstream data capture environment.

12. Select the CDB12c window and change the directory to \$ORACLE_HOME/dbs. You must use the orapwd utility to generate a password file for both instances. It is crucial that you provide the IGNORECASE option when creating the password file. Failing to do so will prevent proper log shipping from the source database to its target:

```
[OS prompt]$ cd $ORACLE_HOME/dbs  
[OS prompt]$ orapwd file=orapwCDB12c password=<password>  
ignorecase=Y entries=30 force=Y  
[OS prompt]$ orapwd file=orapwDWNSTR password=<password>  
ignorecase=Y entries=30 force=Y
```

13. Unset the ORACLE_SID variable and make sure that you can connect to both CDB12c and DWNSTR by using the sys account. Additionally, use the CREATE PFILE command to save the parameter file for each instance in the /tmp directory. You will use the saved parameter files to reset both instances to their default parameters at the end of this practice:

```
[OS prompt]$ export ORACLE_SID=""  
[OS prompt]$ sqlplus sys/oracle@CDB12c as sysdba  
SQL*Plus: Release 12.1.0.2.0 Production on Sat Mar 14 12:07:13  
2015  
Copyright (c) 1982, 2014, Oracle. All rights reserved.  
Connected to:  
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -  
64bit Production  
With the Partitioning, OLAP, Advanced Analytics and Real  
Application Testing options  
SQL> create pfile='/tmp/CDB12c_init.ora' from spfile;  
File created.  
SQL> exit  
Disconnected from Oracle Database 12c Enterprise Edition Release  
12.1.0.2.0 - 64bit Production  
With the Partitioning, OLAP, Advanced Analytics and Real  
Application Testing options  
[OS prompt]$ sqlplus sys/oracle@DWNSTR as sysdba  
SQL*Plus: Release 12.1.0.2.0 Production on Sat Mar 14 12:07:13  
2015  
Copyright (c) 1982, 2014, Oracle. All rights reserved.  
Connected to:  
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -  
64bit Production  
With the Partitioning, OLAP, Advanced Analytics and Real  
Application Testing options  
SQL> create pfile='/tmp/DWNSTR_init.ora' from spfile;  
File created.  
SQL> exit  
Disconnected from Oracle Database 12c Enterprise Edition Release  
12.1.0.2.0 - 64bit Production
```

With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

14. The two RDBMS instances CDB12c and DWNSTR must now be bounced (restarted) so that the new IGNORECASE option is read by the two instances. Click the CDB12c window and set the ORACLE_SID variable to CDB12c. Then, connect to the instance as sysdba and use the shutdown and startup commands to bounce the instance:

```
[OS prompt]$ export ORACLE_SID=CDB12c
[OS prompt]$ sqlplus / as sysdba
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup open
ORACLE instance started.

Total System Global Area 1811939328 bytes
Fixed Size          2925504 bytes
Variable Size       1140853824 bytes
Database Buffers    654311424 bytes
Redo Buffers        13848576 bytes
Database mounted.
Database opened.
SQL>
```

15. Click the DWNSTR window and set the ORACLE_SID variable to DWNSTR. Then, connect to the instance as sysdba and use the shutdown and startup commands to bounce the instance:

```
[OS prompt]$ export ORACLE_SID=DWNSTR
[OS prompt]$ sqlplus / as sysdba
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup open
ORACLE instance started.

Total System Global Area 1660944384 bytes
Fixed Size          2925072 bytes
Variable Size       1107299824 bytes
Database Buffers    536870912 bytes
Redo Buffers        13848576 bytes
Database mounted.
Database opened.
SQL>
```

Configuring the source database for log shipping

16. The CDB12c database is your source database, and DWNSTR is the downstream database. Configure the source database to transmit the Redo log files to the downstream mining database. Select the CDB12c window, where sqlplus is connected to the CDB12c instance as sysdba, and set the LOG_ARCHIVE_DEST_2 parameter for log shipping:

```
SQL> alter system set log_archive_dest_2='SERVICE=DWNSTR SYNC
NOREGISTER VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) REOPEN=10
DB_UNIQUE_NAME=DWNSTR';
System altered.
```

17. Enable the parameter that you just set:

```
SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2=ENABLE;
System altered.
```

18. Set the DG_CONFIG attribute of the LOG_ARCHIVE_CONFIG initialization parameter to include the DB_UNIQUE_NAME of the source database and the downstream database:

```
SQL> ALTER SYSTEM SET
LOG_ARCHIVE_CONFIG='DG_CONFIG=(CDB12c,DWNSTR)' ;
```

19. While you are connected as sys to the source database, determine the size of the source log file. Make note of the results:

```
SQL> select bytes/1024/1024 MB from gv$log;
          MB
-----
      50
      50
      50
```

20. Determine the number of online log file groups that are configured on the source database. Make note of the results:

```
SQL> select count(group#) from gv$log;
COUNT(GROUP#)
-----
            3
```

21. Alter the CDB12c database and enable FORCE LOGGING. If the database is already in force logging mode, you receive the message “Database is already in force logging stage” which you can safely disregard:

```
SQL> alter database force logging;
```

Creating additional directories to host Redo logs and standby Redo logs

22. Exit sqlplus. At the OS prompt, create three directories that will host the Redo logs, the archived logs, and the standby Redo logs for the DWNSTR downstream mining instance:

```
[OS prompt]$ mkdir /u02/oradata/DWNSTR/arch
[OS prompt]$ mkdir /u02/oradata/DWNSTR/redo
[OS prompt]$ mkdir /u02/oradata/DWNSTR/standby
[OS prompt]$ ls -l /u02/oradata/DWNSTR | grep drw
drwxr-xr-x 2 oracle oinstall        4096 Jan 22 08:00 arch
drwxr-xr-x 2 oracle oinstall        4096 Jan 21 17:54 redo
```

```
drwxr-xr-x 2 oracle oinstall      4096 Jan 21 17:45 standby
drwxr-xr-x 2 oracle oinstall      4096 Jan 21 22:19 oggdwn1
drwxr-xr-x 2 oracle oinstall      4096 Jan 21 14:26 oggdwn2
drwxr-x--- 2 oracle oinstall      4096 Jan 21 14:19 pdbseed
```

Configuring the downstream mining database

23. Select the DWNSTR window, where sqlplus is connected to the DWNSTR instance as sysdba, and configure the database to archive the standby Redo data, to receive and write the shipped Redo data locally, and to create the standby Redo log files:

```
SQL> alter system set
log_archive_dest_1='LOCATION=/u02/oradata/DWNSTR/arch
VALID_FOR=(ONLINE_LOGFILE,PRIMARY_ROLE)';
System altered.

SQL> alter system set
log_archive_dest_2='LOCATION=/u02/oradata/DWNSTR/standby
VALID_FOR=(STANDBY_LOGFILE,ALL_ROLES)';
System altered.

SQL> alter system set log_archive_config='DG_CONFIG=(CDB12c,
DWNSTR)';
System altered.
```

24. You configured the destination directory for the local archive log files (/u02/oradata/DWNSTR/arch) and the destination directory for the shipped standby log groups (/u02/oradata/DWNSTR/standby). Additionally, the LOG_ARCHIVE_CONFIG parameter configures the Data Guard component for log shipping between the CDB12c and DWNSTR instances. Enable the parameters that you have just defined:

```
SQL> alter system set log_archive_dest_state_1='ENABLE';
System altered.

SQL> alter system set log_archive_dest_state_2='ENABLE';
System altered.
```

25. The last step for the downstream mining database configuration is the creation of the standby Redo log files, which must be at least of the same size as the original Redo log files from the source database. In addition, the number of log file groups must be equal to the number of Redo log groups in the source database plus one. Previously, you determined that the size of the Redo log files in the source is 50 megabytes, and that there are three Redo log file groups. Therefore, create four standby Redo log groups, with each standby log file 60 megabytes in size:

```
SQL> alter database add standby logfile group 4
('/u02/oradata/DWNSTR/redo/slog4a.rdo','/u02/oradata/DWNSTR/redo
/slog4b.rdo') size 60M;
Database altered.

SQL> alter database add standby logfile group 5
('/u02/oradata/DWNSTR/redo/slog5a.rdo','/u02/oradata/DWNSTR/redo
/slog5b.rdo') size 60M;
Database altered.

SQL> alter database add standby logfile group 6
('/u02/oradata/DWNSTR/redo/slog6a.rdo','/u02/oradata/DWNSTR/redo
```

```
/slog6b.rdo') size 60M;  
Database altered.
```

```
SQL> alter database add standby logfile group 7  
('/u02/oradata/DWNSTR redo/slog7a.rdo','/u02/oradata/DWNSTR redo  
/slog7b.rdo') size 60M;  
Database altered.
```

26. Make sure the standby Redo log groups were added successfully, and then exit sqlplus:

```
SQL> select group#,thread#,sequence#,archived,status  
from V$STANDBY_LOG;
```

GROUP#	THREAD#	SEQUENCE#	ARC	STATUS
4	0	0	YES	UNASSIGNED
5	0	0	YES	UNASSIGNED
6	0	0	YES	UNASSIGNED
7	0	0	YES	UNASSIGNED

```
SQL>
```

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

Practice 3-3: Deploying Integrated Capture by Using Downstream Real-Time Mode

Overview

In this practice, you deploy an Integrated Capture Extract, together with a Data Pump and the corresponding Replicat group on the target schema. The Integrated Capture configuration uses a real-time downstream mining database.

Assumptions

Practice 3-2 has been successfully completed, and both source and downstream mining databases are correctly configured for Redo log shipping (source DB CDB12C) and real-time capture (downstream mining DB DWNSTR).

Tasks

Creating Oracle GoldenGate users with required privileges

1. In Practice 2-2, you created an Oracle GoldenGate administrative user (C##OGG_ADMIN) for all pluggable databases in CDB12C. You must now create a similar user on the downstream mining database (username C##OGG_DWNADM). In the Wallet, you have already defined an alias for it in Practice 3-1 (ogg_dwnadm). You must use the DBMS_GOLDENGATE_AUTH package to grant the C##OGG_DWNADM user the necessary privileges to operate as an Oracle GoldenGate administrative user.
2. Select the Gnome terminal marked DWNSTR, where sqlplus is connected to the DWNSTR instance as sysdba, and create the C##OGG_DWNADM user. Grant the dba, cdb_dba, and pdb_dba roles to C##OGG_DWNADM, and then use the DBMS_GOLDENGATE_AUTH package to grant admin privileges to the OGGADMWN user:

```
SQL> create user C##OGG_DWNADM identified by <password> default
      tablespace users container=all;
User created.

SQL> grant dba, cdb_dba, pdb_dba to C##OGG_DWNADM container=all;
Grant succeeded.
```

3. Invoke the GRANT_ADMIN_PRIVILEGE() procedure for the OGGADMWN user, specifying the following parameters (make sure you enter the entire command on one line, even though it is shown on two lines here):

```
SQL> exec dbms_goldengate_auth.grant_admin_privilege(
      'C##OGG_DWNADM',CONTAINER=>'all');
PL/SQL procedure successfully completed.
```

Connecting to both databases and registering the Integrated Extract

4. Select the Gnome window marked OGG_SRC. GGSCI is running in the Oracle GoldenGate source instance (/u03/ogg/ogg_src).
 - a) Exit GGSCI and, at the OS prompt, set the ORACLE_SID environment variable to point to the mining database (DWNSTR).
 - b) Launch GGSCI again and use DBLOGIN to log in to the CDB12C database.
 - c) Use the MININGDBLOGIN command to connect to the downstream database.

- d) Finally, register the Integrated Extract with the log mining database by using the `register extract` command:

```
GGSCI ([host]) 18> exit
[OS prompt]$ export ORACLE_SID=DWNSTR
[OS prompt]$ ggsci
GGSCI ([host]) 1> DBLogin UserIdAlias ogg_admin
Successfully logged into database CDB$ROOT.
GGSCI ([host]) as C##OGG_ADMIN@CDB12c/CDB$ROOT) 2> MiningDBLogin
UserIdAlias ogg_dwnadm
Successfully logged into mining database.
GGSCI ([host]) 4> register extract extdw database container
(ogg1)
INFO    OGG-02003 Extract EXTDW successfully registered with
database at SCN 2850346.
```

In your environment, the SCN number may differ, but you can safely disregard the discrepancy.

Creating the Integrated Extract

5. Edit the parameter file for the EXTDW Integrated Extract group:

```
GGSCI ([host]) as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit param
extdw
Extract extdw
UserIdAlias ogg_admin
TranLogOptions MiningUserAlias ogg_dwnadm
tranlogoptions integratedparams (max_sga_size 128,
      downstream_real_time_mine Y)
exttrail ./dirdat/dw
SOURCECATALOG ogg1
Table src_user.*;
```

Note: Make sure that the `TranLogOptions` options are entered on one line, even though in the text above, they are shown across two lines.

The Integrated Extract uses two sets of credentials to connect to both the source and the logmining databases. In addition, more parameters are specified through the `TRANLOGOPTIONS INTEGRATEDPARAMS` syntax. In this case, you are requesting the allocation of 128 megabytes of `streams_pool_size` memory, and you are specifying that you are using a real-time downstream mining database.

6. Save the parameter file. Add the Integrated Extract to GGSCI, specifying that it is an Integrated Extract:

```
GGSCI ([host] as C##OGG_ADMIN@CDB12c/CDB$ROOT) 7> add extract
extdw, integrated tranlog, begin now
EXTRACT added.
```

- Add the exttrail file, linking it to the EXTDW Extract:

```
GGSCI ([host] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add exttrail
./dirdat/dw, extract extdw, megabytes 10
EXTTRAIL added.
```

- Start the Integrated Extract, if it is not already started automatically by the Manager. There is no problem in trying to start the Extract and it is already running. GGSCI will just report that the Extract is already running:

```
GGSCI ([host]) 9> start extract extdw
Sending START request to MANAGER ...
EXTRACT EXTDW starting
GGSCI ([host]) 10> info all
Program      Status      Group      Lag at Chkpt Time Since Chkpt
MANAGER      RUNNING
EXTRACT      STOPPED     EXTDW      00:04:55      00:00:00
```

- The EXTDW Extract group could be initially set to STOPPED or STARTING rather than RUNNING. If no Redo log files have been shipped to the downstream database yet, the Integrated Extract fails to attach to the logmining server. You can force the CDB12c instance to switch its log file to facilitate the shipping of the Redo log file to the downstream instance. Click the CDB12c window and force a log file switch:

```
SQL> alter system switch logfile;
System altered.
```

- Click the OGG_SRC window and enter the INFO ALL command repeatedly, until you verify that the status for the EXTDW Extract group is RUNNING.
- Enter view report EXTDW to display extended information on the Integrated Extract. Look for the string INFO OGG-02036 Integrated capture successfully attached to logmining server OGG\$CAP_EXTDW, which confirms that EXTDW was able to connect to the logmining database.

Creating the Data Pump Extract

- Create the PUDW Data Pump Extract. First edit its parameter file:

```
GGSCI ([host] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit param pudw
```

```
Extract pudw
SETENV (ORACLE_SID='CDB12c')
UserIdAlias ogg_admin
RMTHOST ogg_target, MGRPORT 7909
RMTTRAIL ./dirdat/pw
SOURCECATALOG ogg1
TABLE src_user.*;
```

- Save the file. When you are back in GGSCI, add the Data Pump, specifying that it reads committed transactions from the ./dirdat/dw trail file:

```
GGSCI ([host] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
pudw, exttrailsource ./dirdat/dw
EXTRACT added.
```

14. Add the remote trail file, linking it to the Data Pump:

```
GGSCI ([host] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add rmttrail
./dirdat/pw, extract pudw, megabytes 10
RMTTRAIL added.
```

15. Start the Data Pump Extract, and show the status of all Extract groups:

```
GGSCI ([host] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start pudw
Sending START request to MANAGER ...
EXTRACT PUDW starting
GGSCI ([host]) 12> info all

Program Status Group Lag at Chkpt Time Since Chkpt
MANAGER RUNNING
EXTRACT RUNNING EXTDW 00:00:00 00:00:07
EXTRACT RUNNING PUDW 00:00:00 13:14:51
```

Creating the REDW Replicat to ensure change delivery

16. Select the OGG_TRG window and edit the parameter file for the REDW Replicat:

```
GGSCI ([host]) 1> edit param redw
REPLICAT redw
SETENV (ORACLE_SID='CDB12c')
DBOPTIONS INTEGRATEDPARAMS (parallelism 4)
UserIdAlias ogg_repl
ASSUMETARGETDEFS
MAP ogg1.src_user.* , TARGET ogg2.trg_user.*;
```

17. Save the parameter file. Log in using the ogg_repl credentials; add the Replicat group, specifying that it reads its data from the ./dirdat/pw remote trail:

```
GGSCI ([host]) > DBLogin UserIdAlias ogg_repl
Successfully logged into database OGG2.
GGSCI ([host] as C##OGG_ADMIN@CDB12c/OGG2) > Add Replicat redw,
Integrated ExtTrail ./dirdat/pw
REPLICAT (Integrated) added.
GGSCI ([host] as C##OGG_ADMIN@CDB12c/OGG2) > start rep redw

Sending START request to MANAGER ...
REPLICAT REDW starting
GGSCI ([host] as C##OGG_ADMIN@CDB12c/OGG2) > info all
Program Status Group Lag at Chkpt Time Since Chkpt
MANAGER RUNNING
REPLICAT RUNNING REDW 00:00:00 00:00:01
```

Generating database activity to test replication

18. Create an additional terminal shell. At the OS prompt, change the directory to `~/labs/Practice03/configs` and launch `sqlplus`, connecting to the CDB12c database as the `SRC_USER` user. Invoke the `gdp_by_year_2009.sql` script to insert GDP values for the year 2009:

```
[OS prompt]$ cd ~/labs/Practice03/configs
[OS prompt]$ sqlplus src_user/<password>@ogg1
SQL*Plus: Release 12.1.0.2.0 Production on Sat Mar 14 17:05:14
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> @gdp_by_year_2009.sql
1 row created.
...many lines omitted for clarity...
SQL>
```

19. Without exiting `sqlplus`, connect to the target schema `TRG_USER` and verify that the rows have been replicated. Then exit `sqlplus`:

```
SQL> connect trg_user/<password>@ogg2
Connected.
SQL> select count(*) from gdp_by_year where gdp_year=2009;

COUNT(*)
-----
235
SQL> exit
```

20. Close the terminal window opened in step 18 (either enter `exit` or click the x icon on the top right of the window). Select the `OGG_TRG` window and display the statistics for the REDW Integrated Replicat:

```
GGSCI ([host] as C##OGG_ADMIN@CDB12c/OGG2) > stats redw
Sending STATS request to REPLICAT REDW ...
Start of Statistics at 2015-03-14 17:02:45.
Integrated Replicat Statistics:

  Total transactions           1.00
  Redirected                  0.00
  DDL operations              0.00
  Stored procedures            0.00
  Datatype functionality      0.00
  Event actions               0.00
  Direct transactions ratio   0.00%

Replicating from OGG1.SRC_USER.GDP_BY_YEAR to
OGG2.TRG_USER.GDP_BY_YEAR:

*** Total statistics since 2015-03-14 17:02:39 ***
  Total inserts                235.00
  Total updates                 0.00
  Total deletes                 0.00
  Total discards                0.00
  Total operations              235.00

*** Daily statistics since 2015-03-14 17:02:39 ***
  Total inserts                235.00
  Total updates                 0.00
  Total deletes                 0.00
  Total discards                0.00
  Total operations              235.00

*** Hourly statistics since 2015-03-14 17:02:39 ***
  Total inserts                235.00
  Total updates                 0.00
  Total deletes                 0.00
  Total discards                0.00
  Total operations              235.00

*** Latest statistics since 2015-03-14 17:02:39 ***
  Total inserts                235.00
  Total updates                 0.00
  Total deletes                 0.00
  Total discards                0.00
  Total operations              235.00

End of Statistics.
```

The rows inserted into the source database have been correctly replicated to the target schema. The Extract statistics show that 235 rows were captured from the replication source. You have configured a working real-time downstream mining database by deploying an Integrated Extract with its Data Pump and a Replicat on the target environment.

21. Select the OGG_SRC window, where GGSCI is connected to the source Oracle GoldenGate instance, and enter the DBLOGIN command to make sure GGSCI is connected to the source database. Then stop and delete all Extract groups:

```
GGSCI ([host]) > DBLogin UserIdAlias ogg_admin
Successfully logged into database CDB$ROOT.
GGSCI ([host] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop er *
Sending STOP request to EXTRACT EXTDW ...
Request processed.

Sending STOP request to EXTRACT PUDW ...
Request processed.

GGSCI ([host] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > delete er *
Are you sure you want to delete all groups? y
Deleted EXTRACT EXTDW.
Deleted EXTRACT PUDW.
GGSCI ([host] as C##OGG_ADMIN@CDB12c/CDB$ROOT) >
```

22. Unregister the EXTDW Integrated Extract:

```
GGSCI ([host]) > unregister extract EXTDW database
Successfully unregistered EXTRACT EXTDW from database.
```

23. Stop and delete the Replicat group. Select the window where GGSCI is running, connected to the target Oracle GoldenGate instance (OGG_TRG), and then stop and delete the REDW Replicat group:

```
GGSCI ([host] as C##OGG_ADMIN@CDB12c/OGG2) > stop rep redw

Sending STOP request to REPLICAT REDW ...
Request processed.

GGSCI ([host] as C##OGG_ADMIN@CDB12c/OGG2) > delete rep redw
2015-03-14 17:15:14 INFO      OGG-02529 Successfully unregistered
REPLICAT REDW inbound server OGG$REDW from database.
Deleted REPLICAT REDW.
GGSCI ([host] as C##OGG_ADMIN@CDB12c/OGG2) >
```

Resetting the database instances to their default parameters

24. The log-shipping feature provided by Data Guard is not needed for the remaining practices. The CDB12c and DWNSTR RDBMS instances must be reset to their original parameters. You saved the default parameters in the /tmp directory (the /tmp/CDB12c_init.ora file stores the parameters for the CDB12c instance, and the /tmp/DWNSTR_init.ora file stores the parameters for the DWNSTR instance). Select the CDB12c window; make sure you are connected as the oracle user and, at the shell prompt, set the ORACLE_SID environment variable to CDB12c. Connect to oracle as sys and shut down the CDB12c instance. Alternatively, you can run the clean-up-database.sh script found in /home/oracle/labs/Practice03/configs. Do not run the clean-up-database.sh script more than once.

```
[OS prompt]$ export ORACLE_SID=CDB12c
[OS prompt]$ sqlplus sys/oracle as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Sat Mar 14 17:26:18
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Sat Mar 14 2015 17:01:59 +11:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

25. Start the CDB12c instance by using the pfile you previously created (/tmp/CDB12c_init.ora). When the instance has started, create the SPFILE from the parameter file:

```
SQL> startup pfile='/tmp/CDB12c_init.ora'
ORACLE instance started.

Total System Global Area 1811939328 bytes
Fixed Size          2925504 bytes
Variable Size       1140853824 bytes
Database Buffers    654311424 bytes
Redo Buffers        13848576 bytes
Database mounted.
Database opened.
SQL> create spfile from pfile='/tmp/CDB12c_init.ora';
File created.

SQL>
```

26. Repeat the same operation for the DWNSTR instance. Select the DWNSTR window; make sure you are connected as the oracle user and, at the shell prompt, set the ORACLE_SID environment variable to DWNSTR. Connect to oracle as sys and shut down the DWNSTR instance:

```
[OS prompt]$ export ORACLE_SID=DWNSTR
[OS prompt]$ sqlplus sys/oracle as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Sat Mar 14 17:26:18
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Sat Mar 14 2015 17:01:59 +11:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL>
```

27. Start the DWNSTR instance by using the pfile you previously created (/tmp/DWNSTR_init.ora). When the instance has started, create the SPFILE from the parameter file; shut down the DWNSTR database to free up some memory. You will not use DWNSTR until Practice 8:

```
SQL> startup pfile='/tmp/DWNSTR_init.ora'
ORACLE instance started.
Total System Global Area 1778384896 bytes
Fixed Size          2925408 bytes
Variable Size       1140853920 bytes
Database Buffers    620756992 bytes
Redo Buffers        13848576 bytes
Database mounted.
Database opened.
SQL> create spfile from pfile='/tmp/DWNSTR_init.ora';
File created.
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
```

28. Select the CDB12c window and exit sqlplus; then, leave the CDB12c window open and remain connected as oracle for the following practices.

29. Select the DWNSTR window and exit sqlplus. At the OS prompt, enter exit and press Enter to terminate the DWNSTR window.

This completes Practice 3-3.

This completes Practice 3. Stop here.

THESE eKIT MATERIALS ARE FOR YOUR USE IN THIS CLASSROOM ONLY. COPYING eKIT MATERIALS FROM THIS COMPUTER IS STRICTLY PROHIBITED

Oracle University and Error : You are not a Valid Partner use only

Practices for Lesson 4: Oracle GoldenGate with Oracle Real Application Clusters

Chapter 4

Practices for Lesson 4: Overview

Practices Overview

In these practices, you will verify the two-node Oracle RAC configuration available on your workstation and you will set up Oracle GoldenGate to run on the Database File System (DBFS). Your workstation uses Oracle VirtualBox to run a two-node Oracle RAC. The two nodes are:

- 1) ol6-121-rac1
- 2) ol6-121-rac2

Their network configuration is as follows:

Hostname	Public IP	Private IP	Virtual IP
ol6-121-rac1	192.168.56.101	ol6-121-rac1-priv 192.168.57.101	ol6-121-rac1-vip 192.168.56.103
ol6-121-rac2	192.168.56.102	ol6-121-rac2-priv 192.168.57.102	ol6-121-rac2-vip 192.168.56.104

Oracle Single Client Access Name (SCAN) is configured for the RAC as follows:

IP Address	Lookup Name
192.168.56.105	ol6-121-scan
192.168.56.106	ol6-121-scan
192.168.56.107	ol6-121-scan

There are no catch-up or clean-up scripts for this practice.

Practice 4-1: Prepare the Environment

Overview

In this practice, you make sure that no wastage of resources is occurring on the system so that the maximum available memory can be allocated to the Virtual Machines running the Oracle cluster.

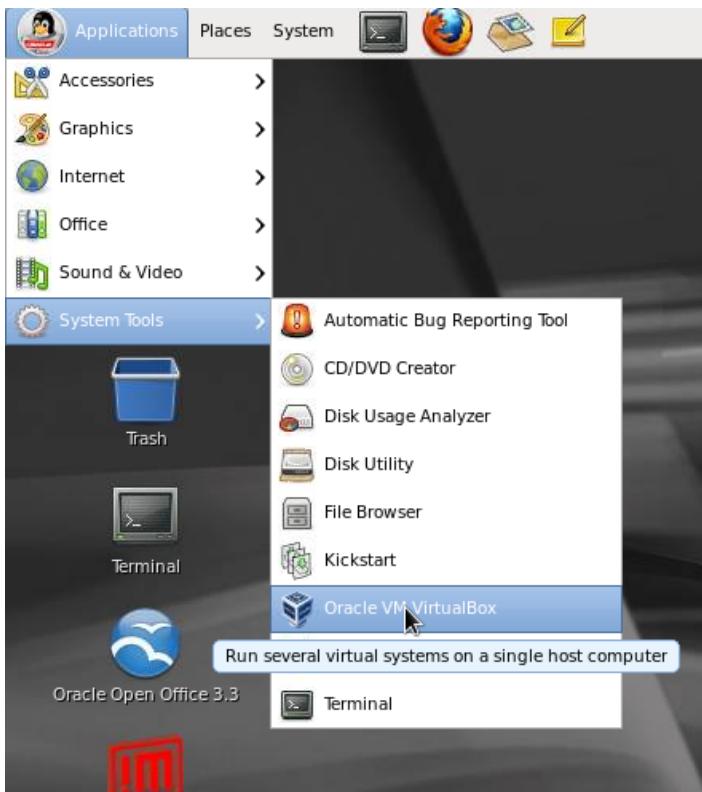
Tasks

1. Open a terminal shell and check if the DWNSTR Oracle RDBMS instance is running. Use the UNIX/Linux `ps -ef` command to check if the `pmon` processes are running. You should see a `pmon` process for the CDB12c instance. If you also determine that the DWNSTR instance is running, you must shut it down because that instance is not used and it takes up valuable memory space, which reduces the memory available for VirtualBox. If you must stop DWNSTR, set the Oracle environment variable to point to DWNSTR and use `sqlplus` connecting as `sysdba` to shut down the instance. After the instance has been shut down, exit `sqlplus` and terminate the window:

```
[OS prompt]$ ps -ef | grep pmon
oracle    4223      1  0 12:54 ?          00:00:00 ora_pmon_CDB12c
oracle    4575      1  0 12:56 ?          00:00:00 ora_pmon_DWNSTR
oracle    4908  4194  0 12:58 pts/0        00:00:00 grep pmon
[OS prompt]$ export ORACLE_SID=DWNSTR
[OS prompt]$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Sun Mar 29 13:10:51
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
[OS prompt]$ exit
```

2. Launch Oracle VirtualBox by selecting Applications->System Tools->Oracle VM Virtual Box.



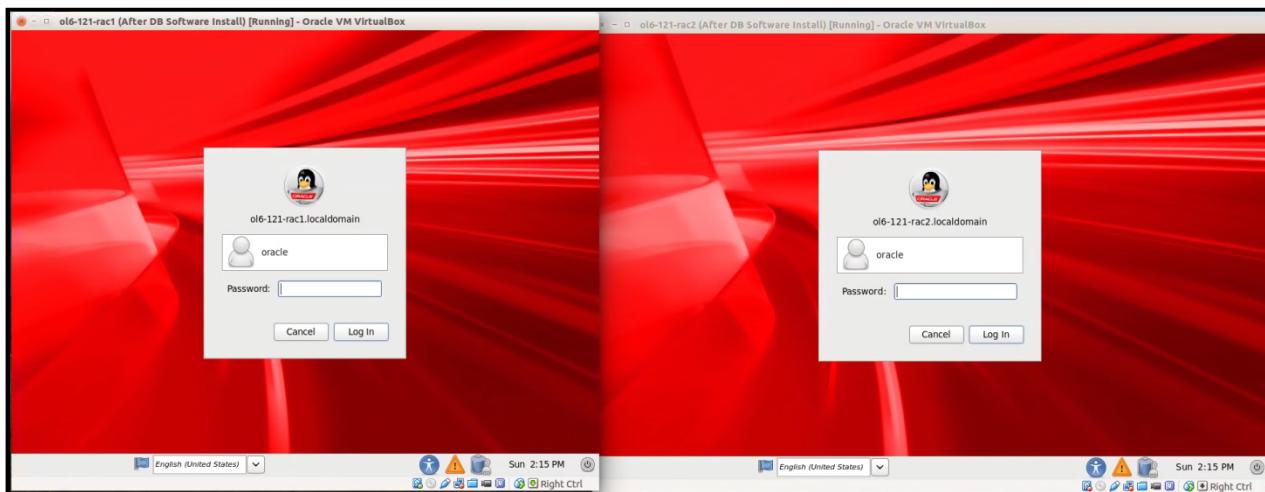
3. When the Oracle VirtualBox window appears on the screen, select the first virtual machine (ol6-121-rac1) and click the "Start" green arrow icon:



4. Repeat the same process for the second virtual machine (`ol6-121-rac2`) as soon as VirtualBox returns control to your mouse:



5. Align the two virtual machines on your desktop:



6. Connect as the "oracle" user on both nodes. **Note:** If you are running this practice in a physical classroom and have access to a real workstation, you can use the terminal shells inside the virtual machines. If, on the other hand, you are in an LVC environment where you are connected to the workstation over an Internet connection, it is better to open an `ssh` session from your client computer to the remote machine, and another `ssh` session to the virtual machine managed by VirtualBox. The amount of flickering during screen refreshes soon renders the environment unworkable. An `ssh` terminal window is under those circumstances more reliable.

Verifying the RAC configuration

7. Open a terminal on the host computer (remember, host computer, not any of the virtual machines) and perform some ping and nslookup operations to verify that networking is operating optimally. You want to make sure that the SCAN address is properly resolved through nslookup, and that you can ping the virtual machines:

```
[OS prompt]$ nslookup ol6-121-scan
Server:      192.168.56.1
Address:   192.168.56.1#53

Name:        ol6-121-scan
Address:  192.168.56.105
Name:        ol6-121-scan
Address:  192.168.56.106
Name:        ol6-121-scan
Address:  192.168.56.107
[OS prompt]$ ping ol6-121-rac1
PING ol6-121-rac1.localdomain (192.168.56.101) 56(84) bytes of
data.
64 bytes from ol6-121-rac1.localdomain (192.168.56.101):
icmp_seq=1 ttl=64 time=0.519 ms
64 bytes from ol6-121-rac1.localdomain (192.168.56.101):
icmp_seq=2 ttl=64 time=0.120 ms
64 bytes from ol6-121-rac1.localdomain (192.168.56.101):
icmp_seq=3 ttl=64 time=0.123 ms
^C
--- ol6-121-rac1.localdomain ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2725ms
rtt min/avg/max/mdev = 0.120/0.254/0.519/0.187 ms
[OS prompt]$ ping ol6-121-rac2
PING ol6-121-rac2.localdomain (192.168.56.102) 56(84) bytes of
data.
64 bytes from ol6-121-rac2.localdomain (192.168.56.102):
icmp_seq=1 ttl=64 time=1.27 ms
64 bytes from ol6-121-rac2.localdomain (192.168.56.102):
icmp_seq=2 ttl=64 time=0.131 ms
^C
--- ol6-121-rac2.localdomain ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1650ms
rtt min/avg/max/mdev = 0.131/0.700/1.270/0.570 ms
[OS prompt]$
```

8. If the SCAN was properly resolved via DNS and the public IP address of the virtual machine responded to the ping commands, it is now time to check the private and the virtual IP addresses:

```
[OS prompt]$ ping ol6-121-rac1-priv
PING ol6-121-rac1-priv.localdomain (192.168.57.101) 56(84) bytes
of data.
64 bytes from ol6-121-rac1-priv.localdomain (192.168.57.101):
icmp_seq=1 ttl=64 time=0.442 ms
64 bytes from ol6-121-rac1-priv.localdomain (192.168.57.101):
icmp_seq=2 ttl=64 time=0.178 ms
^C
--- ol6-121-rac1-priv.localdomain ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1580ms
rtt min/avg/max/mdev = 0.178/0.310/0.442/0.132 ms
[OS prompt]$ ping ol6-121-rac2-priv
PING ol6-121-rac2-priv.localdomain (192.168.57.102) 56(84) bytes
of data.
64 bytes from ol6-121-rac2-priv.localdomain (192.168.57.102):
icmp_seq=1 ttl=64 time=0.381 ms
64 bytes from ol6-121-rac2-priv.localdomain (192.168.57.102):
icmp_seq=2 ttl=64 time=0.149 ms
^C
--- ol6-121-rac2-priv.localdomain ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1418ms
rtt min/avg/max/mdev = 0.149/0.265/0.381/0.116 ms
[OS prompt]$ ping ol6-121-rac1-vip
PING ol6-121-rac1-vip.localdomain (192.168.56.103) 56(84) bytes
of data.
64 bytes from ol6-121-rac1-vip.localdomain (192.168.56.103):
icmp_seq=1 ttl=64 time=0.130 ms
64 bytes from ol6-121-rac1-vip.localdomain (192.168.56.103):
icmp_seq=2 ttl=64 time=0.094 ms
^C
--- ol6-121-rac1-vip.localdomain ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1853ms
rtt min/avg/max/mdev = 0.094/0.112/0.130/0.018 ms
[OS prompt]$ ping ol6-121-rac2-vip
PING ol6-121-rac2-vip.localdomain (192.168.56.104) 56(84) bytes
of data.
64 bytes from ol6-121-rac2-vip.localdomain (192.168.56.104):
icmp_seq=1 ttl=64 time=0.261 ms
64 bytes from ol6-121-rac2-vip.localdomain (192.168.56.104):
icmp_seq=2 ttl=64 time=0.146 ms
^C
--- ol6-121-rac2-vip.localdomain ping statistics ---
```

```
2 packets transmitted, 2 received, 0% packet loss, time 1396ms
rtt min/avg/max/mdev = 0.146/0.203/0.261/0.059 ms
```

9. Click the VirtualBox window that contains the first virtual machine (ol6-121-rac1) and open a shell terminal. Repeat all the tests that you performed on the host computer, specifically:

```
nslookup ol6-121-scan
ping ol6-121-rac1
ping ol6-121-rac2
ping ol6-121-rac1-priv
ping ol6-121-rac2-priv
ping ol6-121-rac1-vip
ping ol6-121-rac2-vip
```

10. Click the VirtualBox window that contains the second virtual machine (ol6-121-rac2) and open a terminal shell. Repeat all the tests that you performed on the host computer and in ol6-121-rac1.

IMPORTANT: If any of the tests fails, either on the host or any of the VMs, contact your instructor and seek help. If any of the network addresses is not reachable, the likelihood of the cluster being operational is very slim.

11. After you have verified that the network that is supporting the cluster is operational, you can assess the state of the cluster from an Oracle Clusterware point of view. Click the VirtualBox window that contains the first virtual machine (ol6-121-rac1) and open or reuse a terminal shell. In the /home/oracle directory, you find two shell files that set the environment for either the database (db_env) or the grid environment (grid_env). Your shell defines two aliases for grid_env and db_env, which take care of sourcing the shell files, so at the prompt, you just invoke either db_env (if you want to set up the variables to use the database) or grid_env (if you want to have access to the Clusterware environment). Source the grid_env shell file by entering grid_env and enter the crsctl command, requesting for cluster statistics:

Cluster Statistics				
Name	Target	State	Server	State
details				
Local Resources				
ora.DATA.dg	ONLINE	ONLINE	ol6-121-rac1	STABLE
	ONLINE	ONLINE	ol6-121-rac2	STABLE
ora.LISTENER.lsnr				
	ONLINE	ONLINE	ol6-121-rac1	STABLE
	ONLINE	ONLINE	ol6-121-rac2	STABLE
ora.asm				

	ONLINE	ONLINE	ol6-121-rac1	Started, STABLE
	ONLINE	ONLINE	ol6-121-rac2	Started, STABLE
ora.net1.network				
	ONLINE	ONLINE	ol6-121-rac1	STABLE
	ONLINE	ONLINE	ol6-121-rac2	STABLE
ora.ons				
	ONLINE	ONLINE	ol6-121-rac1	STABLE
	ONLINE	ONLINE	ol6-121-rac2	STABLE
<hr/>				
<hr/>				
Cluster Resources				
<hr/>				
<hr/>				
	ora.LISTENER_SCAN1.lsnr			
	1	ONLINE	ONLINE	ol6-121-rac2
	ora.LISTENER_SCAN2.lsnr			
	1	ONLINE	ONLINE	ol6-121-rac1
	ora.LISTENER_SCAN3.lsnr			
	1	ONLINE	ONLINE	ol6-121-rac1
	ora.MGMTLSNR			
	1	ONLINE	ONLINE	ol6-121-rac1
	169.254.109.232	192.		
	168.57.101	, STABLE		
	ora.cvu			
	1	ONLINE	ONLINE	ol6-121-rac1
	ora.mgmtdb			
	1	ONLINE	ONLINE	ol6-121-rac1
	Open, STABLE			
	ora.oc4j			
	1	ONLINE	ONLINE	ol6-121-rac1
	ora.oggrac.db			
	1	OFFLINE	OFFLINE	Instance Shutdown, STABLE
	2	OFFLINE	OFFLINE	Instance Shutdown, STABLE
	ora.ol6-121-rac1.vip			
	1	ONLINE	ONLINE	ol6-121-rac1
	ora.ol6-121-rac2.vip			
	1	ONLINE	ONLINE	ol6-121-rac2
	ora.scan1.vip			
	1	ONLINE	ONLINE	ol6-121-rac2
	ora.scan2.vip			
	1	ONLINE	ONLINE	ol6-121-rac1
	ora.scan3.vip			
	1	ONLINE	ONLINE	ol6-121-rac1
	STABLE			
<hr/>				

The output shows that all the components of the cluster are fine, except for the database (`ora.oggrac.db`), which is currently shut down. **IMPORTANT:** If any of the components of the cluster is in a state different from this output, inform your instructor and seek help, as you will not be able to continue the practice.

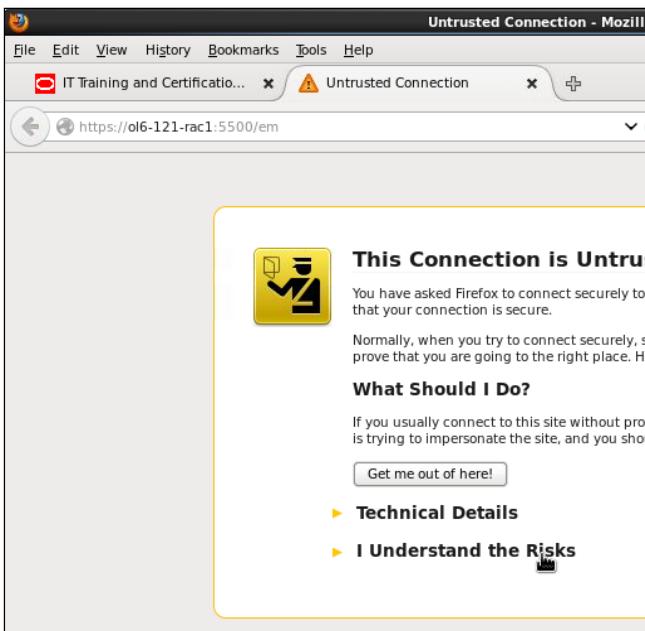
12. Click the VirtualBox window that contains the second virtual machine (`ol6-121-rac2`) and open or reuse a terminal shell. Source the `grid_env` shell file and submit the `crsctl` command `crsctl stat res -t`. Verify the output. All components should show a STABLE status, with the exception of `ora.oggrac.db`, which has yet to be started.
13. Start the RAC instance OGGRAC. Click the VirtualBox window that contains the first virtual machine (`ol6-121-rac1`) and open or reuse a terminal shell. Source the `db_env` shell file and use the `srvctl` command utility to start the database:

```
[OS prompt] $ db_env  
[OS prompt] $ srvctl start database -d OGGRAC
```

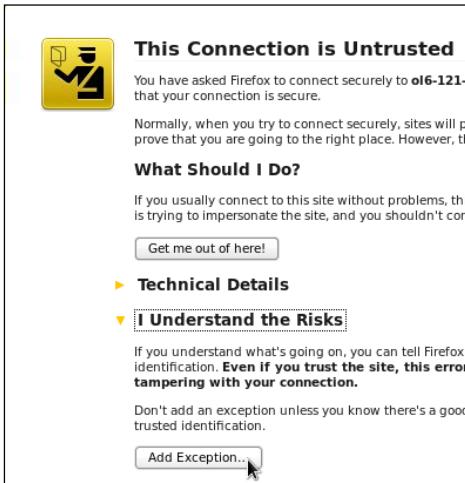
14. When the control returns to the shell after `srvctl` executes, you should have a RAC instance up and running. On the host computer, launch Firefox and enter the URL:

<https://ol6-121-rac1:5500/em>

15. Firefox displays a popup window with the alert that you are requesting an untrusted connection. Select the "I Understand the Risks" option:



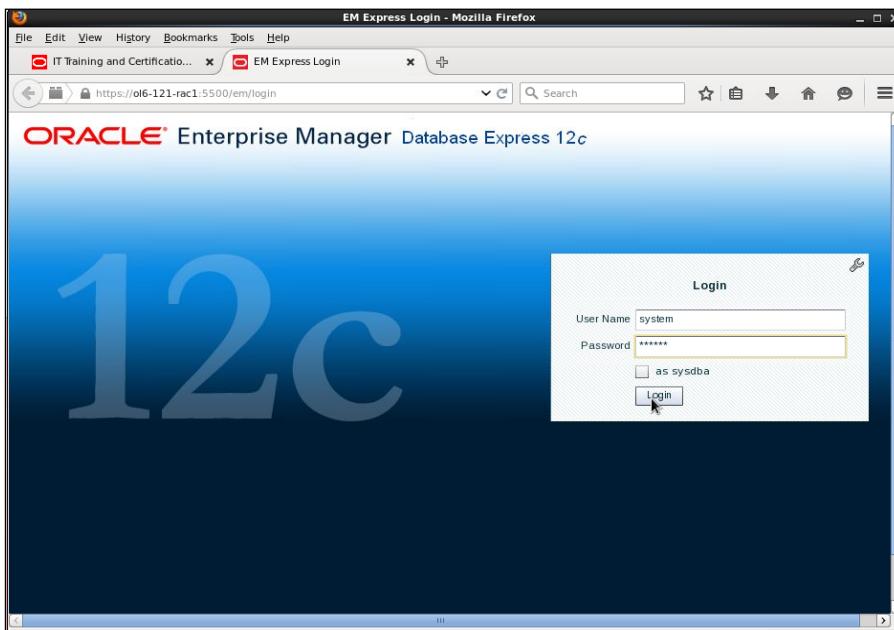
16. The window expands and an "Add Exception" button appears:



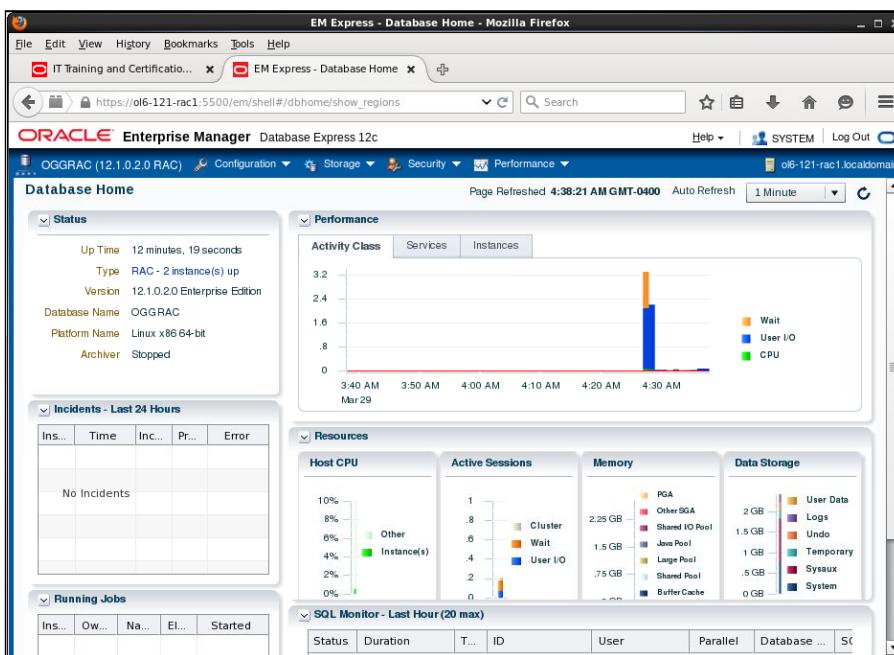
17. Click the "Add Exception" button. The "Add Security Exception" window appears:



18. Click the "Confirm Security Exception" button. The Oracle Enterprise Manager finally appears. Log in as the "system" user (ask your instructor for the password if you do not know it).



19. The OEM main page is displayed:



This last step concludes Practice 4-1. Continue with Practice 4-2.

Practice 4-2: Configuring DBFS

Overview

In this practice, you perform all the steps required to install DBFS on the RAC instance.

Assumptions

Practice 4-1 has been successfully completed and the RAC instance OGGRAC is started and healthy.

Tasks

1. If you are in an LVC environment or if you experience sluggishness while interacting with the virtual machines, you can shut down your Firefox browser. You will not need it for the rest of this practice.
2. Click the VirtualBox window that contains the ol6-121-rac1 virtual machine and open or reuse a terminal shell. You must connect as root to change a few system files. Use the su command and log in as root:

```
[OS prompt]$ su -  
Password: *****  
[OS prompt]#
```

3. Edit the /etc/fuse.conf file. Create the file if it does not exist. Add the user_allow_other directive, save the file, and exit the editor:

```
[OS prompt]# vi /etc/fuse.conf  
user_allow_other
```

4. Change the file permissions for /etc/fuse.conf to 644 (root can change the file, everyone else can read it):

```
[OS prompt]# chmod 644 /etc/fuse.conf
```
5. Repeat the same settings for the second node (ol6-121-rac2). **IMPORTANT:** Make sure that you modify the /etc/fuse.conf configuration file on the rac2 node also. Forgetting to do so is a common cause of problems down the track. Click the VirtualBox window that contains the ol6-121-rac2 virtual machine and open or reuse a terminal shell. Use su to log in as root, add user_allow_other to the /etc/fuse.conf file, save the file, and change its permissions to 644.
6. Create the DBFS mount point on both nodes (identical directory structure). Select the rac1 window and reuse the terminal shell that is connected as root. Create the /oracle/dbfs_direct directory and change its ownership to the oracle user. Repeat the same commands for the rac2 node:

ol6-121-rac1:

```
[OS prompt]# mkdir -p /oracle/dbfs_direct  
[OS prompt]# chown oracle.dba /oracle/dbfs_direct  
[OS prompt]# ls -ld /oracle/dbfs_direct  
drwxr-xr-x 2 oracle dba 4096 Mar 29 23:37 /oracle/dbfs_direct/
```

ol6-121-rac2:

```
[OS prompt]# mkdir -p /oracle/dbfs_direct
```

```
[OS prompt]# chown oracle.dba /oracle/dbfs_direct  
[OS prompt]# ls -ld /oracle/dbfs_direct  
drwxr-xr-x 2 oracle dba 4096 Mar 29 23:40 /oracle/dbfs_direct/
```

7. This step is only to be performed on rac1, when you are connected as the oracle user. Create the dbfs_ts tablespace connected as sysdba. Use the db_env alias to set the appropriate environment variables. First exit from the su environment and go back to the oracle user in the rac1 window, where you left a terminal shell connected as root. Click the VirtualBox window that contains the o16-121-rac1 virtual machine and follow these steps:

```
[OS prompt]# exit  
[OS prompt]$ db_env  
[OS prompt]$ echo $ORACLE_SID  
OGGRAC1  
[OS prompt]$ sqlplus / as sysdba  
SQL*Plus: Release 12.1.0.2.0 Production on Sun Mar 29 23:42:32  
2015  
Copyright (c) 1982, 2014, Oracle. All rights reserved.  
Connected to:  
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -  
64bit Production  
With the Partitioning, Real Application Clusters, Automatic  
Storage Management, OLAP,  
Advanced Analytics and Real Application Testing options  
SQL> create bigfile tablespace dbfs_ts datafile '+DATA' size  
1024M autoextend on next 100M maxsize 3G NOLOGGING EXTENT  
MANAGEMENT LOCAL AUTOALLOCATE SEGMENT SPACE MANAGEMENT AUTO;  
Tablespace created.
```

8. While still connected to the Oracle RDBMS as sysdba, create the DBFS_USER user and grant the required privileges to run DBFS:

```
SQL> create user dbfs_user identified by oracle default  
tablespace dbfs_ts quota unlimited on dbfs_ts;  
User created.  
SQL> grant create session, create table, create view, create  
procedure, dbfs_role to dbfs_user;  
Grant succeeded.
```

9. Exit `sqlplus` and relaunch it, this time connecting to the Oracle RDBMS as `dbfs_user`. Use the `dbfs_create_filesystem` SQL script to create the file system that will be subsumed into Oracle. The `dbfs_create_filesystem` script resides in the `$ORACLE_HOME/rdbms/admin` directory, so position your shell into that directory before launching `sqlplus` and connecting to the Oracle RDBMS as the `dbfs_user`:

```
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
[OS prompt]$ cd $ORACLE_HOME/rdbms/admin
[OS prompt]$ pwd
/u01/app/oracle/product/12.1.0.2/db_1/rdbms/admin/
[OS prompt]$ sqlplus dbfs_user/oracle
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Mon Mar 30 2015 00:01:04 +11:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
SQL> start dbfs_create_filesystem dbfs_ts FS1
No errors.

-----
CREATE STORE:
begin dbms_dbfs_sfs.createFilesystem(store_name => 'FS1',
tbl_name => 'FS1',
tbl_tbs => 'dbfs_ts', lob_tbs => 'dbfs_ts', do_partition =>
false, partition_key
=> 1, do_compress => false, compression => '', do_dedup =>
false, do_encrypt =>
false); end;
-----
REGISTER STORE:
begin dbms_dbfs_content.registerStore(store_name=> 'FS1',
provider_name =>
'sample1', provider_package => 'dbms_dbfs_sfs'); end;
-----
MOUNT STORE:
begin dbms_dbfs_content.mountStore(store_name=>'FS1',
store_mount=>'FS1'); end;
-----
```

```
CHMOD STORE:  
declare m integer; begin m := dbms_fuse.fs_chmod('/FS1', 16895);  
end;  
No errors.  
SQL> exit  
[OS prompt]$
```

10. You must set the NOCACHE storage option to the LOB segment that is used by DBFS. Reconnect to the local RAC database node (OGGRAC1) as sysdba and alter the FS1 table, which is owned by DBFS_USER, to modify the FILEDATA LOB:

```
[OS prompt]$ echo $ORACLE_SID  
OGGRAC1  
[OS prompt]$ sqlplus / as sysdba  
SQL*Plus: Release 12.1.0.2.0 Production on Mon Mar 30 22:20:19  
2015  
  
Copyright (c) 1982, 2014, Oracle. All rights reserved.  
Connected to:  
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -  
64bit Production  
With the Partitioning, Real Application Clusters, Automatic  
Storage Management, OLAP,  
Advanced Analytics and Real Application Testing options  
  
SQL> ALTER TABLE DBFS_USER.FS1 MODIFY LOB (FILEDATA) (NOCACHE  
LOGGING);  
Table altered.  
SQL> column owner format a10  
SQL> column table_name format a15  
SQL> column segment_name format a15  
SQL> SELECT owner,table_name,segment_name,logging,cache FROM  
dba_lobs WHERE tablespace_name='DBFS_TS';  
  
OWNER        TABLE_NAME          SEGMENT_NAME      LOGGING CACHE  
-----  
DBFS_USER    FS1                LOB_SFSS_FST_1    YES      NO  
SQL> exit  
Disconnected from Oracle Database 12c Enterprise Edition Release  
12.1.0.2.0 - 64bit Production  
With the Partitioning, Real Application Clusters, Automatic  
Storage Management, OLAP,  
Advanced Analytics and Real Application Testing options  
[OS prompt]$
```

11. On both RAC nodes, you must configure the fuse libraries (the libraries that implement the file system in user space, an essential component of DBFS). You must connect as the root user. Start with ol6-121-rac1:

```
[OS prompt] $ hostname
ol6-121-rac1.localdomain
[OS prompt] $ su -
Password: *****
[OS prompt] # echo "/usr/local/lib" >>
/etc/ld.so.conf.d/usr_local_lib.conf
[OS prompt] # cat /etc/ld.so.conf.d/usr_local_lib.conf
/usr/local/lib
[OS prompt] # cd /usr/local/lib
[OS prompt] # pwd
/usr/local/lib
[OS prompt] #
```

The echo command above must be entered on one line, even if it is shown over two lines because of formatting. After you position your default directory to /usr/local/lib, you create several logical links to Oracle RDBMS shared objects and fuse shared objects:

```
[OS prompt] # ln -s /u01/app/12.1.0.2/grid/lib/libclntsh.so.12.1
[OS prompt] # ln -s /u01/app/12.1.0.2/grid/lib/libnnz12.so
[OS prompt] # ln -s /lib64/libfuse.so.2 libfuse.so
[OS prompt] # ls -l
total 0
lrwxrwxrwx 1 root root 44 May 12 22:22 libclntsh.so.12.1 ->
/u01/app/12.1.0.2/grid/lib/libclntsh.so.12.1
lrwxrwxrwx 1 root root 19 May 12 22:16 libfuse.so ->
/lib64/libfuse.so.2
lrwxrwxrwx 1 root root 38 May 12 22:23 libnnz12.so ->
/u01/app/12.1.0.2/grid/lib/libnnz12.so
[OS prompt] #
```

You now run the ldconfig command to make the shared objects that you just configured available to the system:

```
[OS prompt] # ldconfig
[OS prompt] #
```

The next step is to create a logical link for the DBFS_CLIENT executable in the /sbin directory:

```
[OS prompt] # ln -s /u01/app/12.1.0.2/grid/bin/dbfs_client
/sbin/mount.dbfs
[OS prompt] # ls -l /sbin/mount.dbfs
lrwxrwxrwx 1 root root 33 Mar 30 22:59 /sbin/mount.dbfs ->
/u01/app/12.1.0.2/grid/bin/dbfs_client
```

12. The last step is to grant execute privilege on the /bin/fusermount file to all users. You will connect as "oracle" to mount DBFS, so Oracle must be able to run fusermount:

```
[OS prompt] # chmod +x /bin/fusermount
```

13. You must now repeat all the steps required to configure the fuse libraries for the second node, ol6-121-rac2. Click the VirtualBox window that contains the ol6-121-rac2 virtual machine and open or reuse a terminal shell. You must connect as root:

```
[OS prompt]$ hostname
ol6-121-rac2.localdomain
[OS prompt]$ su -
Password: *****
[OS prompt]# echo "/usr/local/lib" >>
/etc/ld.so.conf.d/usr_local_lib.conf
[OS prompt]# cat /etc/ld.so.conf.d/usr_local_lib.conf
/usr/local/lib
[OS prompt]# cd /usr/local/lib
[OS prompt]# pwd
/usr/local/lib
[OS prompt]# ln -s /u01/app/12.1.0.2/grid/lib/libclntsh.so.12.1
[OS prompt]# ln -s /u01/app/12.1.0.2/grid/lib/libnnz12.so
[OS prompt]# ln -s /lib64/libfuse.so.2 libfuse.so
[OS prompt]# ls -l
total 0
lrwxrwxrwx 1 root root 39 Mar 30 23:31 libclntsh.so.12.1 ->
/u01/app/12.1.0.2/grid/lib/libclntsh.so.12.1
lrwxrwxrwx 1 root root 19 Mar 30 23:32 libfuse.so ->
/lib64/libfuse.so.2
lrwxrwxrwx 1 root root 33 Mar 30 23:31 libnnz12.so ->
/u01/app/12.1.0.2/grid/lib/libnnz12.so
[OS prompt]# ldconfig
[OS prompt]# ln -s /u01/app/12.1.0.2/grid/bin/dbfs_client
/sbin/mount.dbfs
[OS prompt]# ls -l /sbin/mount.dbfs
lrwxrwxrwx 1 root root 33 Mar 30 23:33 /sbin/mount.dbfs ->
/u01/app/12.1.0.2/grid/bin/dbfs_client
[OS prompt]# chmod +x /bin/fusermount
[OS prompt]#
```

14. Copy the mount-dbfs.sh script from /media/sf_install to /u01/app/12.1.0.2/grid/crs/script. Edit the script and make sure that the following variables are set correctly:

Variable	Value
DBNAME	OGGRAC
MOUNT_POINT	/oracle/dbfs_direct
DBFS_USER	dbfs_user
ORACLE_HOME	/u01/app/oracle/product/12.1.0.2/db_1
DBFS_PASSWD	oracle

The `mount-dbfs.sh` script must be copied to both nodes, `ol6-121-rac1` and `ol6-121-rac2`:

```
[OS prompt]# cd /u01/app/12.1.0.2/grid/crs/script  
[OS prompt]# cp /media/sf_install/mount-dbfs.sh .  
[OS prompt]# ls -l  
-rwxr-x--- 1 root root 11610 Apr 1 10:15 mount-dbfs.sh
```

15. Change the file ownership and permissions for `mount-dbfs.sh`:

```
[OS prompt]# chown oracle.dba mount-dbfs.sh  
[OS prompt]# chmod 750 mount-dbfs.sh  
[OS prompt]# ls -l  
-rwxr-x--- 1 oracle dba 11610 Apr 1 10:35 mount-dbfs.sh
```

16. Repeat the steps for the `ol6-121-rac2` node. You must end up with the `mount-dbfs.sh` file located in `/u01/app/12.1.0.2/grid/crs/script` also on the second node, and that file must belong to `oracle.dba` and must have octal permissions 750.
17. Manually run the `/u01/app/12.1.0.2/grid/crs/script/mount-dbfs.sh` script with the "start" option as the "oracle" OS user on both nodes to mount the DBFS file system for the first time. Start with `ol6-121-rac1`. Click the `ol6-121-rac1` virtual machine window and type in the following commands:

```
[OS prompt]# exit  
[OS prompt]$ id  
uid=54321(oracle) gid=54321(oinstall)  
groups=54321(oinstall),492(vboxsf),54322(dba)  
[OS prompt]$ /u01/app/12.1.0.2/grid/crs/script/mount-dbfs.sh  
start  
mount-dbfs.sh mounting DBFS at /oracle/dbfs_direct from database  
OGGRAC  
ORACLE_SID is OGGRAC1  
spawning dbfs_client command using SID oggrac1  
nohup: redirecting stderr to stdout  
Start - ONLINE  
[OS prompt]$
```

Note: The first time the DBFS file system is mounted, it could take some time. The `mount_dbfs.sh` script sleeps for an interval of time and then checks for the status of the DBFS mount. If the time required to mount DBFS is longer than the sleep interval, `mount_dbfs.sh` could erroneously report the status as OFFLINE. If you see OFFLINE rather than ONLINE in your output, do not panic. Wait for a few seconds and issue the command:

```
[OS] $ /u01/app/12.1.0.2/grid/crs/script/mount-dbfs.sh status
```

This time around, the status should be ONLINE. Run the `mount_dbfs.sh` script with the status parameter until you see ONLINE. If after a couple of minutes, the DBFS status is still OFFLINE, alert your instructor.

18. Use `df -h` to check the newly mounted file system:

```
[OS prompt]$ df -h
Filesystem           Size   Used  Avail Use% Mounted on
/dev/mapper/vg_ol6121rac1-lv_root
                           45G   20G   23G  47% /
tmpfs                 1.9G   632M  1.3G  34% /dev/shm
/dev/sda1              477M   127M  321M  29% /boot
install                699G   525G  174G  76% /media/sf_install
dbfs-dbfs_user@:/      3.0G   120K  3.0G   1% /oracle/dbfs_direct
[OS prompt]$
```

19. Verify that you can perform I/O operations on the DBFS file system:

```
[OS prompt]$ touch /oracle/dbfs_direct/FS1/try.txt
[OS prompt]$ ls -l /oracle/dbfs_direct/FS1
total 0
-rw-r--r-- 1 oracle oinstall 0 Apr  1 14:03 try.txt
```

20. Mount the DBFS file system on the second node and verify that you can access the file (`try.txt`), which you created from the first node. Click the `ol6-121-rac2` virtual machine window, select the terminal shell you left open before (which was connected as root), and type the following commands:

```
[OS prompt]# exit
[OS prompt]$ id
uid=54321(oracle) gid=54321(oinstall)
groups=54321(oinstall),492(vboxsf),54322(dba)
[OS prompt]$ /u01/app/12.1.0.2/grid/crs/script/mount-dbfs.sh
start
mount-dbfs.sh mounting DBFS at /oracle/dbfs_direct from database
OGGRAC
ORACLE_SID is OGGRAC2
spawning dbfs_client command using SID oggrac2
nohup: redirecting stderr to stdout
Start - ONLINE
[OS prompt]$ ls -l /oracle/dbfs_direct/FS1
total 0
-rw-r--r-- 1 oracle oinstall 0 Apr  1 14:03 try.txt
```

This last step ends Practice 4-2. Leave all windows open as they are.

Continue with Practice 4-3.

Practice 4-3: Configuring DBFS as a CRS service

Overview

In this practice, you configure the DBFS file system to be mounted by the Oracle Clusterware services.

Assumptions

Practices 4-1 and 4-2 have been successfully carried out, the RAC instance is running, and DBFS has been successfully deployed and manually started.

Tasks

1. On both nodes, you must dismount the DBFS file system. Start with `ol6-121-rac1`. Click the `ol6-121-rac1` virtual machine window, select the terminal shell open there, and type in the following commands:

```
[OS] $ /u01/app/12.1.0.2/grid/crs/script/mount-dbfs.sh stop
unmounting DBFS from /oracle/dbfs_dir
umounting the filesystem using '/bin/fusermount -u
/oracle/dbfs_direct'
Stop - stopped, now not mounted
[OS] $ df -h
Filesystem           Size   Used  Avail Use% Mounted on
/dev/mapper/vg_ol6121rac1-lv_root
                      45G   20G   23G  47% /
tmpfs                1.9G  632M  1.3G  34% /dev/shm
/dev/sda1              477M  127M  321M  29% /boot
install                699G  525G  174G  76% /media/sf_install
[OS prompt]$
```

2. Dismount DBFS on the second node. Click the VirtualBox window that contains the `ol6-121-rac2` virtual machine and reuse the terminal shell open there:

```
[OS] $ /u01/app/12.1.0.2/grid/crs/script/mount-dbfs.sh stop
unmounting DBFS from /oracle/dbfs_dir
umounting the filesystem using '/bin/fusermount -u
/oracle/dbfs_direct'
Stop - stopped, now not mounted
[OS] $ df -h
Filesystem           Size   Used  Avail Use% Mounted on
/dev/mapper/vg_ol6121rac1-lv_root
                      45G   20G   23G  47% /
tmpfs                1.9G  628M  1.3G  33% /dev/shm
/dev/sda1              477M  127M  321M  29% /boot
install                699G  525G  174G  76% /media/sf_install
[OS prompt]$
```

3. Only on the first node, copy the `add-dbfs-resource.sh` file from `/media/sf_install` to `/home/oracle`. Make the file executable and display its contents on the screen. Click the `ol6-121-rac1` virtual machine window, select the terminal shell open there, and type the commands below:

```
[OS prompt]$ cp /media/sf_install/add-dbfs-resource.sh ~
[OS prompt]$ cd ~
[OS prompt]$ chmod +x add-dbfs-resource.sh
[OS prompt]$ cat add-dbfs-resource.sh
#####
start script add-dbfs-resource.sh
#!/bin/bash
ACTION_SCRIPT=/u01/app/12.1.0.2/grid/crs/script/mount-dbfs.sh
RESNAME=dbfs_mount
DBNAME=OGGRAC
DBNAMEL=`echo $DBNAME | tr A-Z a-z`
ORACLE_HOME=/u01/app/12.1.0.2/grid
PATH=$ORACLE_HOME/bin:$PATH
export PATH ORACLE_HOME
crsctl add resource $RESNAME \
-type local_resource \
-attr "ACTION_SCRIPT=$ACTION_SCRIPT, \
CHECK_INTERVAL=30,RESTART_ATTEMPTS=10, \
START_DEPENDENCIES='hard(ora.$DBNAMEL.db) pullup(ora.$DBNAMEL.db)', \
STOP_DEPENDENCIES='hard(ora.$DBNAMEL.db)', \
SCRIPT_TIMEOUT=300"
#####
end script add-dbfs-resource.sh
```

Note: Make sure that the "`-type`" subcommand for `crsctl` is "`local_resource`." You will modify that entry in Practice 12, but here the `dbfs_mount` resource must be defined as "local" rather than "cluster."

The shell file sets some environment variables that are needed for `crsctl` to work. Then it invokes `crsctl`, defining the `dbfs_mount` service.

4. Execute the `add-dbfs-resource.sh` shell file. Then use the `grid_env` alias to set the environment for Oracle Clusterware and verify that CRS is aware of the newly added `dbfs_mount` service:

```
[OS prompt]$ ./add-dbfs-resource.sh
[OS prompt]$ grid_env
[OS prompt]$ crsctl stat res dbfs_mount
NAME=dbfs_mount
TYPE=local_resource
TARGET=OFFLINE, OFFLINE
STATE=OFFLINE, OFFLINE
[OS prompt]$
```

5. Use `crsctl` to start the `dbfs_mount` service, and check that the service was effectively started:

```
[OS prompt]$ crsctl start resource dbfs_mount
CRS-2672: Attempting to start 'dbfs_mount' on 'ol6-121-rac1'
CRS-2672: Attempting to start 'dbfs_mount' on 'ol6-121-rac2'
CRS-2676: Start of 'dbfs_mount' on 'ol6-121-rac2' succeeded
CRS-2676: Start of 'dbfs_mount' on 'ol6-121-rac1' succeeded
[OS prompt]$ crsctl stat res dbfs_mount
NAME=dbfs_mount
TYPE=local_resource
TARGET=ONLINE , ONLINE
STATE=ONLINE on ol6-121-rac1, ONLINE on ol6-121-rac2
[OS prompt]$
```

6. On both nodes, verify that DBFS is mounted and that you can read and write in the FS1 directory:

```
[OS prompt]$ df -h
Filesystem           Size   Used  Avail Use% Mounted on
/dev/mapper/vg_ol6121rac1-lv_root
                           45G   20G   23G  47% /
tmpfs                 1.9G  632M  1.3G  34% /dev/shm
/dev/sda1              477M  127M  321M  29% /boot
install                699G  525G  174G  76% /media/sf_install
dbfs-dbfs_user@:/     3.0G  120K  3.0G   1% /oracle/dbfs_direct
[OS prompt]$ ls -l /oracle/dbfs_direct/FS1
total 0
-rw-r--r-- 1 oracle oinstall 0 Apr  1 14:03 try.txt
[OS prompt]$ touch /oracle/dbfs_direct/FS1/try2.txt
[OS prompt]$ ls -l /oracle/dbfs_direct/FS1/
total 0
-rw-r--r-- 1 oracle oinstall 0 Apr  1 15:45 try2.txt
-rw-r--r-- 1 oracle oinstall 0 Apr  1 14:03 try.txt
[OS prompt]$
```

7. Click the VirtualBox window that contains the o16-121-rac2 virtual machine and reuse the terminal shell open there. Verify that DBFS was in fact mounted by CRS. Also, verify that you can access the file that you just created (try2.txt) and also write files from the second node:

```
[OS prompt]$ df -h
Filesystem           Size   Used  Avail Use% Mounted on
/dev/mapper/vg_o16121rac1-lv_root
                                45G   20G   23G  47% /
tmpfs                 1.9G  628M  1.3G  33% /dev/shm
/dev/sda1              477M  127M  321M  29% /boot
install                699G  525G  174G  76% /media/sf_install
dbfs-dbfs_user@:/      3.0G  120K  3.0G   1% /oracle/dbfs_direct
[OS prompt]$ ls -l /oracle/dbfs_direct/FS1
total 0
-rw-r--r-- 1 oracle oinstall 0 Apr  1 15:45 try2.txt
-rw-r--r-- 1 oracle oinstall 0 Apr  1 14:03 try.txt
[OS prompt]$ touch /oracle/dbfs_direct/FS1/try3.txt
[OS prompt]$ ls -l /oracle/dbfs_direct/FS1
total 0
-rw-r--r-- 1 oracle oinstall 0 Apr  1 15:45 try2.txt
-rw-r--r-- 1 oracle oinstall 0 Apr  1 15:52 try3.txt
-rw-r--r-- 1 oracle oinstall 0 Apr  1 14:03 try.txt
```

8. DBFS was in fact mounted by the Oracle Clusterware services, and both nodes can read and write in the shared device. This last step concludes Practice 4-3. You must now gracefully terminate the VirtualBox session. First, stop the OGGRAC database. You must use the FORCE option of the srvctl command (-f) because DBFS is mounted and the database cannot be stopped unless the force qualifier is used. Then "hibernate" the two virtual machines comprising the RAC configuration. Click the o16-121-rac1 virtual machine window, select the terminal shell open there, and type the commands below:

```
[OS prompt]$ db_env
[OS prompt]$ srvctl stop database -d OGGRAC -f
[OS prompt]$
```

9. Click the ol6-121-rac1 virtual machine window and use the `su` command to log in as root. Shut down the virtual machine by using the Linux `shutdown` command:

```
[OS prompt]$ su -  
Password: *****  
[OS prompt]# shutdown -h now  
Broadcast message from oracle@ol6-121-rac1.localdomain  
        (/dev/pts/0) at 22:41 ...  
The system is going down for halt NOW!  
[OS prompt]#
```

10. Click the ol6-121-rac2 virtual machine and use the `su` command to log in as root. Shut down also this virtual machine by using the Linux `shutdown` command.

```
[OS prompt]$ su -  
Password: *****  
[OS prompt]# shutdown -h now  
Broadcast message from oracle@ol6-121-rac2.localdomain  
        (/dev/pts/0) at 22:43 ...  
The system is going down for halt NOW!  
[OS prompt]#
```

This last step ends practice 4-3. This completes Practice 4 for *Oracle GoldenGate 12c Advanced Configuration for Oracle*.

Practices for Lesson 5: Oracle GoldenGate Event Marker System

Chapter 5

Practices for Lesson 5: Overview

Practices Overview

In these practices, you will familiarize yourself with the Oracle GoldenGate Event Marker system, generate events on the replication source side, and trapping the events on the replication target side.

After the events are trapped, the Replicat process will carry out instructions associated with the various EVENTACTIONS clauses.

All these practices use Integrated Extract groups. The environment for this type of Extract is set up as a preliminary task for Practice 5-1.

Helper scripts provided in /home/oracle/labs/Practice05/configs:

Catch-up/Clean-up script	Purpose
catch-up-prac-5_1.sh	Performs all steps for Practice 5-1
catch-up-prac-5_2.sh	Performs all steps for Practice 5-2
catch-up-prac-5_3.sh	Performs all steps for Practice 5-3
catch-up-prac-5_4.sh	Performs all steps for Practice 5-4
clean-up-prac-5_1.sh	Undoes all steps performed in Practice 5-1
clean-up-prac-5_2.sh	Undoes all steps performed in Practice 5-2
clean-up-prac-5_3.sh	Undoes all steps performed in Practice 5-3
clean-up-prac-5_4.sh	Undoes all steps performed in Practice 5-4

Note: You should either run the clean-up scripts or manually delete and unregister the various Extract and Replicat groups at the end of each practice.

Practice 5-1: Triggering an End-of-Day Job (Database Backup)

Overview

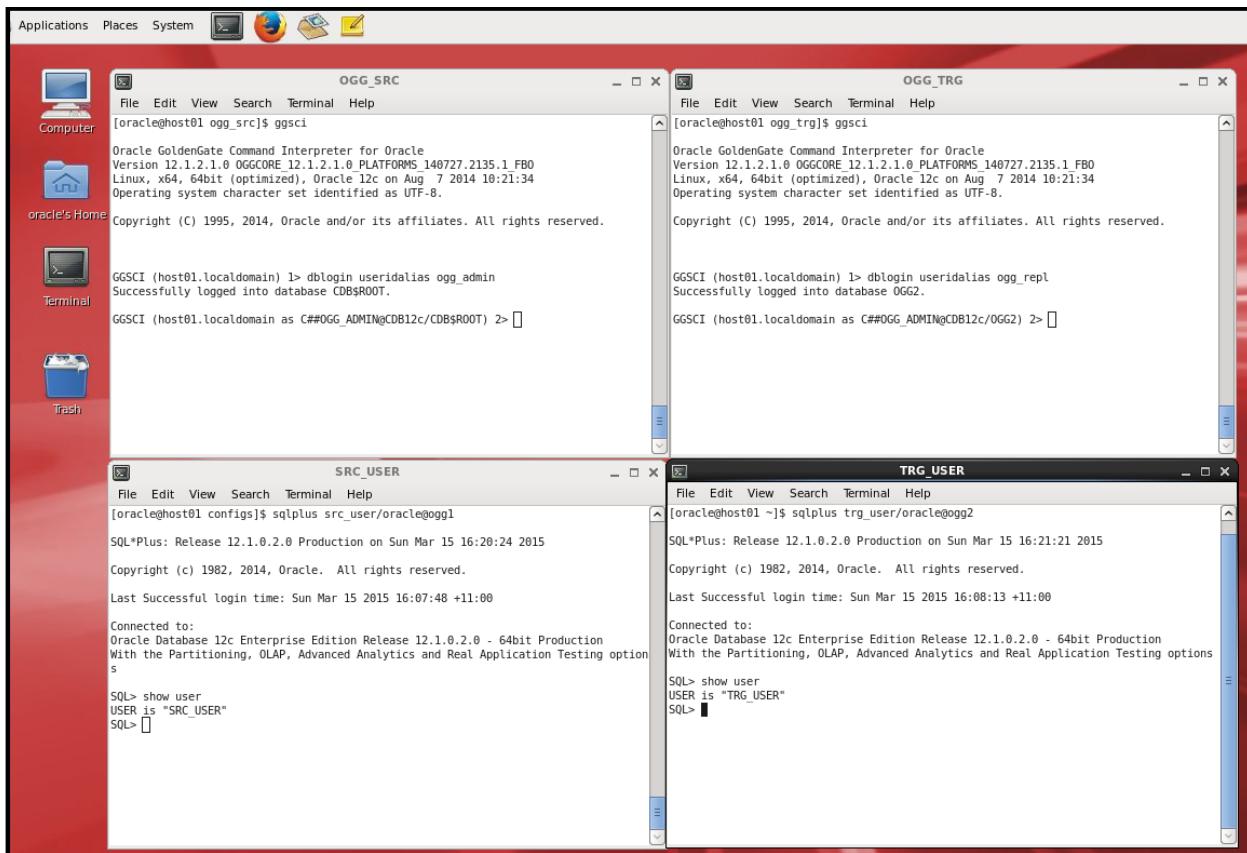
In this practice, you set up the change capture on the source database, together with its associated ExtTrail, and the change delivery on the target side. The Replicat parameters will force the Replicat process to back up the tablespace that is hosting the database objects that belong to the `trg_user` user when the `PERFORM BACKUP` string is stored in a predefined table.

Assumptions

The Oracle database `CDB12c` is running in `ARCHIVELOG` mode; the `src_user` and `trg_user` users are created in the database and have the necessary privileges for replication. The default tablespace for `src_user` and `trg_user` is `OGGDATA`.

Tasks

- Rearrange your desktop environment in such a way that you have four windows aligned with each other. The upper left window has the caption "OGG_SRC" and in it, GGSCI is connected to the replication source (`/u03/ogg/ogg_src`). The window to its right has the caption "OGG_TRG" and in it, GGSCI is connected to the replication target (`/u03/ogg/ogg_trg`). Below these windows, you have on the left a window with the caption "SRC_USER," which hosts a `sqlplus` session connected to the `src_user` schema in the `ogg1` PDB database. On the right, your environment displays a window with the caption "TRG_USER," which hosts a `sqlplus` session connected to the `trg_user` schema in the `ogg2` PDB database:



2. Make sure that the two sqlplus sessions, SRC_USER and TRG_USER, have been started while the default directory is ~/labs/Practice05/configs for both environments. This practice will instruct you to run scripts that are stored in this directory, so the SQL scripts that start with the at (@) sign assume that the current directory is ~/labs/Practice05/configs. If you are not sure that the sqlplus sessions have been started from the current directory, exit sqlplus, navigate to the ~/labs/Practice05/configs directory, and relaunch sqlplus.
3. Create a table called EVENT_TRIGGER in both the source and target environments. Click the SRC_USER window and create the table in the replication source environment:

```
SQL> create table event_trigger (id number not null primary key,  
trigger_type varchar2(255));  
Table created.  
SQL>
```

4. Click the TRG_USER window and create the same table in the replication target environment:

```
SQL> create table event_trigger (id number not null primary key,  
trigger_type varchar2(255));  
Table created.  
SQL>
```

5. Click the OGG_SRC window and use the DBLogin command to connect to the CDB12c database by using the ogg_admin alias; verify that the Oracle GoldenGate Manager is running by issuing the INFO MGR command:

```
GGSCI ([HOST]) > DBLogin UserIdAlias ogg_admin  
Successfully logged into database CDB$ROOT.  
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info mgr  
Manager is running (IP port SRCHOST.7809, Process ID 1576).
```

6. Verify that the table that you just created has supplemental logging enabled in the src_user schema. The src_user schema in the ogg1 PDB should automatically enable supplemental logging because you issued the command "ADD SCHEMATRANSACTIONAL src_user ALLCOLS" in Practice 2-2, step 5.

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info transactional  
ogg1.src_user.*  
...many lines omitted for clarity...  
  
Logging of supplemental redo log data is enabled for table  
OGG1.SRC_USER.EVENT_TRIGGER.  
Columns suplementally logged for table OGG1.SRC_USER.EVENT_TRIGGER:  
ID, TRIGGER_TYPE.  
...many lines omitted for clarity...  
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) >
```

7. Edit the EUEVT primary Extract, which captures all table activity in the `src_user` schema (ogg1 PDB):

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
euevt
```

```
Extract euevt
Exttrail ./dirdat/eu
UseridAlias ogg_admin
LOGALLSUPCOLS
tranlogoptions integratedparams (max_sga_size 128)
UPDATERECORDFORMAT COMPACT
SOURCECATALOG ogg1
Table src_user.*;
```

8. Exit from the editor, saving the parameter file. In GGSCI, register the EUEVT Extract with the database (ogg1) and add the newly created Extract:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > REGISTER
EXTRACT euevt DATABASE CONTAINER (ogg1)
Extract EUEVT successfully registered with database at SCN
2709383.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
euevt, integrated Tranlog, begin now
EXTRACT added.
```

9. Connect the Extract to the ExtTrail, setting the exttrail file size to 10 MB:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add exttrail
./dirdat/eu, Extract euevt, megabytes 10
EXTTRAIL added.
```

10. Add the puevt Data Pump Extract, which physically transfers the captured data to the target host:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
puevt
```

```
Extract puevt
SETENV (ORACLE_SID='CDB12c')
rmthost ogg_target, mgrport 7909, Compress
rmttrail ./dirdat/pu
passthru
SOURCECATALOG ogg1
table src_user.*;
```

11. Connect the Data Pump to the local exttrail file:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
puevt, exttrailsource ./dirdat/eu
EXTRACT added.
```

12. Connect the remote trail file to the Data Pump, setting the remote trail file size to 10 MB:

```
GGSCI ([SRCHOST] ) 9> add rmtrail ./dirdat/pu, Extract puevt,
megabytes 10
RMTTRAIL added.
```

13. Start the Extract and Data Pump processes, check their status, and exit from GGSCI:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start extract
euevt
Sending START request to MANAGER ...
EXTRACT EUEVT starting

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start extract
puevt
Sending START request to MANAGER ...
EXTRACT PUEVT starting

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program      Status       Group      Lag at Chkpt  Time Since
Chkpt

MANAGER      RUNNING
EXTRACT      RUNNING     EUEVT      00:00:00    00:41:39
EXTRACT      RUNNING     PUEVT      00:00:00    00:00:06

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > exit
```

14. Navigate to the /home/oracle directory and use an editor of your choice to create the backup.cmd file, which contains the following RMAN syntax:

```
connect target sys/<password>@ogg2
run {
allocate channel t1 device type disk format
'/home/oracle/rman_backup/%U';
backup tablespace oggdata;
backup archivelog all;
}
```

15. In the same directory, create the rman_backup directory and the shell file that invokes RMAN, passing the parameters contained in the backup.cmd file to it.

```
[OS prompt] $ mkdir rman_backup
[OS prompt] $
```

16. The shell file name is rman_bck.sh. Use an editor of your choice to create rman_bck.sh and add the lines below:

```
#!/bin/bash
rman @/home/oracle/backup.cmd
```

17. Make the script executable by using the Linux chmod command:

```
[OS prompt] $ chmod +x rman_bck.sh
```

18. Select the OGG_TRG window and use the DBLogin command to connect to CDB12c by using the `ogg_repl` alias; verify that the Oracle GoldenGate Manager is running:

```
GGSCI ([HOST]) > DBLogin UserIdAlias ogg_repl
Successfully logged into database OGG2.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) 12> info mgr
Manager is running (IP port edddr60p1.7909, Process ID 1571).
```

19. Create the Replicat parameter file for the Replicat REVTA, specifying the filter clause that will cause the RMAN backup when the "PERFORM BACKUP" string is inserted into the EVENT_TRIGGER table:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > edit params revta
```

```
Replicat revta
SETENV (ORACLE_SID='CDB12c')
DBOPTIONS INTEGRATEDPARAMS (parallelism 4)
AssumeTargetDefs
Discardfile ./dirrpt/revta.dsc, purge
UserIdAlias ogg_repl
map ogg1.src_user.event_trigger, TARGET
ogg2.trg_user.event_trigger,
FILTER(ON INSERT, @streq(TRIGGER_TYPE,'PERFORM BACKUP')),
EVENTACTIONS (SHELL '/home/oracle/rman_bck.sh', REPORT);
map ogg1.src_user.gdp_by_year, target ogg2.trg_user.gdp_by_year;
map ogg1.src_user.gdp_growth_by_year, target
ogg2.trg_user.gdp_growth_by_year;
map ogg1.src_user.economic_entity, target
ogg2.trg_user.economic_entity;
```

20. Save the parameter file. Add the Replicat definition, linking the Integrated Replicat process to its trail file:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > add replicat revta,
Integrated exttrail ./dirdat/pu
REPLICAT (Integrated) added.
```

21. Start the Replicat process:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > start replicat
revta
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > info all
Program      Status       Group        Lag at Chkpt   Time Since
oggChkpt

MANAGER      RUNNING
REPLICAT     RUNNING      REVTA        00:00:00      00:00:06
```

22. Select the SRC_USER window. Exit sqlplus and navigate to the `~/labs/Practice05/configs` directory. Launch sqlplus again, connecting to Oracle as `src_user@ogg1`. Generate some activity by running the `gdp_by_year_2010.sql` script:

```

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
[OS prompt] > cd ~/labs/Practice05/configs
[OS prompt] > sqlplus src_user@ogg1
SQL*Plus: Release 12.1.0.2.0 Production on Sun Mar 15 17:39:30
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Enter password: *****
Last Successful login time: Sun Mar 15 2015 17:38:37 +11:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> @gdp_by_year_2010.sql
1 row created.

...many lines omitted for clarity...

```

23. Now you insert the row that will trigger the RMAN backup performed by a remote shell that is invoked by the Replicat process:

```

SQL> insert into event_trigger values (1,'PERFORM BACKUP');
SQL> commit;

```

24. Select the OGG_TRG window, where GGSCI is connected to the replication target environment. Enter a view report command on the REVTA Replicat to check that the triggering record has been intercepted, and that the shell action has been performed.

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > view report revta
```

25. You should see lines similar to the ones below, which show that the event was fired by the Replicat process:

```

2015-03-15 17:42:20  INFO      OGG-06505  MAP resolved (entry
ogg1.src_user.event_trig
ger): map "OGG1"."SRC_USER"."EVENT_TRIGGER", TARGET
ogg2.trg_user.event_trigger, FILTER(ON INSERT,
@streq(TRIGGER_TYPE,'PERFORM BACKUP')), EVENTACTIONS (SHELL
'/home/oracle/rman_bck.sh', REPORT).
2015-03-15 17:42:21  INFO      OGG-01286  Executing shell command
'/home/oracle/rman_bck.sh' due to SHELL event for target table
OGG2.TRG_USER.EVENT_TRIGGER in file ./dirdat/pu000000, RBA
34527.

```

26. The instructions contained in the shell command file are all listed in the report produced by Replicat for the REVTA Replicat, together with the output generated by the shell.

27. Open an additional shell window and, at the OS prompt, check that the RMAN backup files exist in the directory specified by the RMAN script. Then type `exit` to close the shell window:

```
[OS prompt] > ls -lh rman_backup/
total 3.3M
-rw-r-----. 1 oracle oinstall 3.3M Mar 15 17:42 07q1th7u_1_1
[OS prompt] > exit
```

The names of the RMAN backup files will vary in your environment; the important thing is to make sure that RMAN backup files have been created.

28. Select the `OGG_TRG` window. In the `GGSCI` session connected to the `OGG_TRG` Oracle GoldenGate instance, stop and delete the `REVTA` Replicat. Erase the `RMTTRAIL` file associated with the Replicat (`./dirdat/pu*`) by using a SHELL command (`sh rm -f`):

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > stop replicat revta
Sending STOP request to REPLICAT REVTA ...
Request processed.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > delete replicat
revta
2015-03-15 17:59:34 INFO      OGG-02529 Successfully
unregistered REPLICAT REVTA inbound server OGG$REVTA from
database.

Deleted REPLICAT REVTA.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) >
```

29. Select the `OGG_SRC` window and launch `GGSCI`. Connect to the database as `ogg_admin` and stop and delete the `EUEVT` and `PUEVT` Extract groups. Then unregister the `EUEVT` Extract from the `CDB12c` database:

```
[OS propmt] $ oggsrc
[OS propmt] $ ggsci
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.1.0 OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014 10:21:34
Operating system character set identified as UTF-8.
Copyright (C) 1995, 2014, Oracle and/or its affiliates. All rights reserved.

GGSCI ([HOST]) DBLogin UserIdAlias ogg_admin
Successfully logged into database CDB$ROOT.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since Chkpt

MANAGER RUNNING
EXTRACT RUNNING EUEVT 00:00:00 00:00:03
EXTRACT RUNNING PUEVT 00:00:00 00:00:01

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop er *
Sending STOP request to EXTRACT EUEVT ...
```

```
Request processed.

Sending STOP request to EXTRACT PUEVT ...
Request processed.
GGSCI ( [HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > delete er * !
Deleted EXTRACT EUEVT.
Deleted EXTRACT PUEVT.
GGSCI ( [HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > UNREGISTER
EXTRACT euevt DATABASE
Successfully unregistered EXTRACT EUEVT from database
GGSCI ( [HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) >.
```

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

Practice 5-2: Stopping Processing on Detecting Data Anomalies

Overview

In this practice, you create a Replicat process with a FILTER clause in its parameter file, which provokes the abrupt termination of the process if a data anomaly is detected. The GDP_GROWTH_BY_YEAR table stores the economic growth of worldwide economic entities. A growth of 150% is highly unlikely anywhere in the world, so a value greater than 150.00 in the GDP_VALUE column of the GDP_GROWTH_BY_YEAR table is an indication of a serious data entry error. Such a value would make Replicat abend, so that an investigation can be performed on why such an abnormal value was entered into the system.

Assumptions

The GDP_GROWTH_BY_YEAR table exists in both SRC_USER and TRG_USER schemas.

Tasks

1. Select the OGG_SRC window. In the GGSCI session that is connected to the OGG_SRC Oracle GoldenGate instance, edit the parameter file for the FUEVT primary Extract group:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
fuevt
```

```
Extract fuevt
Exttrail ./dirdat/fu
UserIdAlias ogg_admin
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
SOURCECATALOG ogg1
Table src_user.*;
```

2. Exit from the editor after saving the parameter file. In GGSCI, register the FUEVT Extract with the ogg1 database and then add the newly created FUEVT Extract:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > REGISTER
EXTRACT fuevt DATABASE CONTAINER (ogg1)
Extract FUEVT successfully registered with database at SCN
2765725.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
fuevt, Integrated Tranlog, begin now
EXTRACT added.
```

3. Connect the Extract to the ExtTrail, setting the exttrail file size to 10 MB:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add exttrail
./dirdat/fu, Extract fuevt, megabytes 10
EXTTRAIL added.
```

- Add the `pfevt` Data Pump Extract, which physically transfers the captured data to the target host:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params  
pfevt
```

```
Extract pfevt  
SETENV (ORACLE_SID='CDB12c')  
UserIdAlias ogg_admin  
RMTHOST ogg_target, MGRPORT 7909  
rmttrail ./dirdat/ff  
passthru  
SOURCECATALOG oggl  
TABLE src_user.*;
```

- Connect the Data Pump to the local `exttrail` file:

```
GGSCI GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add  
extract pfevt, exttrailsource ./dirdat/fu  
EXTRACT added.
```

- Connect the remote trail file to the Data Pump, setting the remote trail file size to 10 MB:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add rmttrail  
./dirdat/ff, Extract pfevt, megabytes 10  
RMTTRAIL added.
```

- Start the Extract and Data Pump processes, and check their status:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start extract  
fuevt  
Sending START request to MANAGER ...  
EXTRACT FUEVT starting  
  
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start extract  
pfevt  
Sending START request to MANAGER ...  
EXTRACT PFEVT starting  
  
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all  
Program Status Group Lag at Chkpt Time Since  
Chkpt  
  
MANAGER RUNNING  
EXTRACT RUNNING FUEVT 00:00:00 00:41:39  
EXTRACT RUNNING PFEVT 00:00:00 00:00:06  
  
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) >
```

8. Select the OGG_TRG window. Make sure the manager is running, and create the parameter file for the REVTB Replicat:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > info all
Program      Status      Group      Lag at Chkpt  Time Since
Chkpt

MANAGER      RUNNING

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit param
revtb
```

9. Create the associations between the source and target tables in the parameter files, and add the FILTER clause, which will cause the Replicat process to ABEND after detecting an abnormal value:

```
Replicat revtb
AssumeTargetDefs
SETENV (ORACLE_SID='CDB12c')
DBOPTIONS INTEGRATEDPARAMS (parallelism 4)
Discardfile ./dirrpt/revtb.dsc, purge
UserIdAlias ogg_repl
map ogg1.src_user.gdp_by_year, target ogg2.trg_user.gdp_by_year,
map ogg1.src_user.gdp_growth_by_year, target
ogg2.trg_user.gdp_growth_by_year;
map ogg1.src_user.economic_entity, target
ogg2.trg_user.economic_entity;
TABLE ogg1.src_user.gdp_growth_by_year, FILTER (GDP_VALUE >
150.00), EVENTACTIONS (ABORT);
```

10. Save the parameter file and add the Replicat, linking it to the exttrail ./dirdat/pu, and then start the newly defined REVTB Replicat process:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add replicat
revtb, Integrated exttrail ./dirdat/ff
REPLICAT (Integrated) added.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program      Status      Group      Lag at Chkpt  Time Since
Chkpt

MANAGER      RUNNING
REPLICAT     STOPPED     REVTB      00:00:00      00:00:03
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start replicat
revtb
Sending START request to MANAGER ...
REPLICAT REVTB starting
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program      Status      Group      Lag at Chkpt  Time Since
Chkpt
```

```

MANAGER      RUNNING
REPLICAT    RUNNING     REVTB      00:00:00      00:00:02
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) >

```

11. Select the SRC_USER window, where sqlplus is running as the src_user user. Generate activity in the GDP_GROWTH_BY_YEAR table by running the script gdp_growth_by_year_2008.sql. In the previous practice, you changed the directory to ~/labs/Practice05/configs before launching sqlplus. You should be able to invoke the gdp_growth_by_year_2008.sql script simply by running the script name preceded by the at (@) sign:

```

SQL> @gdp_growth_by_year_2008.sql
1 row created.
...many lines omitted for clarity...

```

12. Format the columns. Execute the SELECT statement that fetches the GDP growth for the year 2008 for countries whose ID is between 192 and 208:

```

SQL> set linesize 132
SQL> set pagesize 20
SQL> column gdp_value format 999.99
SQL> column economic_entity format a60
SQL> select a.entity_id, a.economic_entity,b.gdp_value from
economic_entity a, gdp_growth_by_year b where a.entity_id =
b.entity_id and b.gdp_year=2008 and a.entity_id between 192 and
208;
ENTITY_ID ECONOMIC_ENTITY          GDP_VALUE
----- -----
-
192 Cote D'Ivoire                3.00
193 Djibouti                     7.00
194 Equatorial Guinea            7.40
195 Eritrea                       1.30
196 Ethiopia                      8.50
197 Gabon                         4.30
198 Gambia                        5.80
199 Ghana                          6.20
200 Guinea                        4.30
201 Guinea Bissau                 3.00
202 Kenya                         1.70
203 Lesotho                       1.52
204 Liberia                        7.10
205 Madagascar                    6.30
206 Malawi                        7.44
207 Mali                           4.90
208 Mauritania                   3.00
17 rows selected.

```

13. You now update the row related to the GDP growth for Gabon, and you set it to an abnormal value (200%). The abnormal value should cause the Replicat to stop:

```
SQL> update gdp_growth_by_year set gdp_value=200.00 where
gdp_year = 2008 and entity_id = 197;

1 row updated.

SQL> commit;

Commit complete.
```

14. Select the OGG_TRG window. Enter the "INFO ALL" command. The REVTB Replicat should be in the ABENDED state:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > info all

Program      Status      Group      Lag at Chkpt  Time Since
Chkpt

MANAGER      RUNNING
REPLICAT    ABENDED    REVTB      00:00:00    00:02:57
```

15. Enter a view report revtb command to check why the Replicat processes abended.

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > view report revtb
```

16. The answer is in the line below:

```
2015-03-15 19:27:22  ERROR   OGG-01289  Aborting process due to
ABORT event for source table OGG1.SRC_USER.GDP_GR
OWTH_BY_YEAR in file ./dirdat/ff000000, RBA 36172.
```

17. In your environment, the Relative Byte Address of the record at which the checkpoint was made will possibly be different, but you should find a similar line in your report. Delete the REVTB Replicat to clean the environment for the next practice:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > delete replicat
revtb !
2015-03-15 19:31:59  INFO    OGG-02529  Successfully
unregistered REPLICAT REVTB inbound server OGG$REVTB from
database.
Deleted REPLICAT REVTB.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) >
```

18. Select the OGG_SRC window and stop and delete the FUEVT and PFEVT Extract groups. Unregister the Extract from the ogg1 database:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
EXTRACT     RUNNING    FUEVT      00:00:00    00:00:03
EXTRACT     RUNNING    PFEVT      00:00:00    00:00:01
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop er *
Sending STOP request to EXTRACT FUEVT ...
Request processed.
```

```
Sending STOP request to EXTRACT PFEVT ...
Request processed.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > delete er * !
Deleted EXTRACT FUEVT.
Deleted EXTRACT PFEVT.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > unregister EXTRACT
FUEVT database
Successfully unregistered EXTRACT FUEVT from database.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT)
```

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

Practice 5-3: Tracing the Insertion of Rows Where Specific Columns Carry Abnormally High Values

Overview

In this practice, you use the Oracle GoldenGate Event Marker system to trace the records that match specific conditions, which are evaluated when the Replicat process scans the exttrail file, before updating the target database with the inserted or changed data.

Assumptions

The GDP_GROWTH_BY_YEAR table exists in both SRC_USER and TRG_USER schemas.

Tasks

1. Select the OGG_SRC window. In the GGSCI session that is connected to the OGG_SRC Oracle GoldenGate instance, create the parameter file for the GUEVT Extract group:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params  
guevt  
  
Extract guevt  
Exttrail ./dirdat/gu  
UserIdAlias ogg_admin  
LOGALLSUPCOLS  
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)  
UPDATERECORDFORMAT COMPACT  
SOURCECATALOG ogg1  
Table src_user.*;
```

2. Exit from the editor after saving the parameter file. In GGSCI, add the newly created Extract:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > REGISTER  
EXTRACT guevt DATABASE CONTAINER (ogg1)  
Extract GUEVT successfully registered with database at SCN  
2816097.  
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract  
guevt, Integrated Tranlog, begin now  
EXTRACT added.
```

3. Connect the Extract to the ExtTrail, setting the exttrail file size to 10 MB:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add exttrail  
.dirdat/gu, Extract guevt, megabytes 10  
EXTTRAIL added.
```

4. Add the pgevt Data Pump Extract, which physically transfers the captured data to the target host:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params  
pgevt
```

```

Extract pgevt
SETENV (ORACLE_SID='CDB12c')
UserIdAlias ogg_admin
RMTHOST ogg_target, MGRPORT 7909
rmttrail ./dirdat/gg
passthru
SOURCECATALOG ogg1
table src_user.*;

```

5. Connect the Data Pump to the local exttrail file:

```

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
pgevt, exttrailsource ./dirdat/gu
EXTRACT added.

```

6. Connect the remote trail file to the Data Pump, setting the remote trail file size to 10 MB:

```

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add rmttrail
./dirdat/gg, Extract pgevt, megabytes 10
RMTTRAIL added.

```

7. Start the Extract and Data Pump processes, and check their status:

```

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start extract
guevt
Sending START request to MANAGER ...
EXTRACT GUEVT starting

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start extract
pgevt
Sending START request to MANAGER ...
EXTRACT PGEVT starting

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING GUEVT 00:00:00 00:41:39
EXTRACT RUNNING PGEVT 00:00:00 00:00:06

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) >

```

8. Select the OGG_TRG window, where GGSCI is connected to the OGG_TRG Oracle GoldenGate instance, and enter the edit params revtc command to define the parameters for the new Replicat group:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > edit param revtc
```

9. Define the parameters for the `revtc` Replicat group. The focus is the syntax for the FILTER option of the MAP command.

```
Replicat revtc
AssumeTargetDefs
SETENV (ORACLE_SID='CDB12c')
DBOPTIONS INTEGRATEDPARAMS (parallelism 4)
UserIDAlias ogg_repl
Discardfile ./dirrpt/revtc.dsc, purge
map ogg1.src_user.gdp_by_year, target ogg2.trg_user.gdp_by_year,
map ogg1.src_user.gdp_growth_by_year, target
ogg2.trg_user.gdp_growth_by_year,
WHERE (ENTITY_ID < 80 AND GDP_VALUE > 4.00), EVENTACTIONS
(TRACE /home/oracle/eu_trace.trc );
map ogg1.src_user.economic_entity, target
ogg2.trg_user.economic_entity;
```

10. Add the new Replicat, linking its input to the local exttrail file. Then start the new REVTC Replicat, and make sure it is running:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > add replicat revtc,
Integrated exttrail ./dirdat/gg
REPLICAT (Integrated) added.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > start rep revtc
Sending START request to MANAGER ...
REPLICAT REVTC starting
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > INFO ALL
Program      Status      Group      Lag at Chkpt  Time Since
Chkpt

MANAGER      RUNNING
REPLICAT     RUNNING      REVTC      00:00:00      00:00:07
```

11. Select the SRC_USER window where sqlplus is running. Run the `gdp_growth_by_year_2010.sql` file to generate activity in the `GDP_GROWTH_BY_YEAR` table and to trigger the tracing of those countries that experienced a growth of more than 4% during the year 2010:

```
SQL> @gdp_growth_by_year_2010.sql
1 row created.
...many lines omitted for clarity...
```

12. Open an additional shell window. At the OS prompt, make sure that you are in the default directory (`/home/oracle`). Verify that the trace file has been created:

```
[OS prompt]$ pwd
/home/oracle
[OS prompt]$ ls -lh *.trc
-rw-r-----. 1 oracle oinstall 65K Mar 15 21:30 eu_trace.trc
```

13. Use an editor of your choice to edit the `eu_trace.trc` file to see what information Oracle GoldenGate has dumped into the trace file.
14. Select the `OGG_TRG` window, where `GGSCI` runs connected to the Oracle GoldenGate `OGG_TRG` instance. Stop and delete the `REVTC` Replicat to clean the environment for the next practice:

```
GGSCI GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > stop replicat revtc
Sending STOP request to REPLICAT REVTC ...
Request processed.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > delete replicat revtc
2015-03-15 21:31:58 INFO      OGG-02529 Successfully
unregistered REPLICAT REVTC inbound server OGG$REVTC from
database.
Deleted REPLICAT REVTC.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) >
```

15. Select the `OGG_SRC` window and stop and delete the `GUEVT` and `PGEVT` Extract groups:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program      Status      Group      Lag at Chkpt  Time Since
Chkpt

MANAGER      RUNNING
EXTRACT      RUNNING    GUEVT      00:00:00      00:00:03
EXTRACT      RUNNING    PGEVT      00:00:00      00:00:01

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop er *
Sending STOP request to EXTRACT GUEVT ...
Request processed.

Sending STOP request to EXTRACT PGEVT ...
Request processed.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > delete er * !
Deleted EXTRACT GUEVT.
Deleted EXTRACT PGEVT.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) >
```

16. Unregister the `GUEVT` Extract from the `ogg1` database:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > UNREGISTER
EXTRACT guevt DATABASE
Successfully unregistered EXTRACT GUEVT from database.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) >
```

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

Practice 5-4: Executing Shell Scripts to Validate Data and Sending an Email when Validation Fails

Overview

In this practice, you create two stored procedures and a shell script to perform data validation, which is driven by a combination of SQLLEXEC and EVENTACTIONS commands performed while mapping (MAP) the source table taken from the Data Pump with the target table in the database. The shell script receives three parameters. One of the parameters is a lookup performed through a stored procedure.

The MAP statement invokes SQLLEXEC to perform a database lookup via a stored procedure and uses the value returned from the database lookup as a parameter to the SHELL action triggered by the EVENTACTIONS command.

Tasks

1. Select the OGG_SRC window. In the GGSCI session that is connected to the OGG_SRC Oracle GoldenGate instance, create the parameter file for the HUEVT Extract group:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params  
huevt
```

```
Extract huevt  
Exttrail ./dirdat/hu  
UserIdAlias ogg_admin  
LOGALLSUPCOLS  
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)  
UPDATERECORDFORMAT COMPACT  
SOURCECATALOG ogg1  
Table src_user.*;
```

2. Exit from the editor after saving the parameter file. In GGSCI, register the HUEVT Extract with the ogg1 database and then add the newly created Extract:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > REGISTER  
EXTRACT huevt DATABASE CONTAINER (ogg1)  
Extract HUEVT successfully registered with database at SCN  
2860877.  
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract  
huevt, Integrated Tranlog, begin now  
EXTRACT added.
```

3. Connect the Extract to the ExtTrail, setting the exttrail file size to 10 MB:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add exttrail  
.dirdat/hu, Extract huevt, megabytes 10  
EXTTRAIL added.
```

4. Add the phevt Data Pump Extract, which physically transfers the captured data to the target host:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
phevt
```

```
Extract phevt
SETENV (ORACLE_SID='CDB12c')
UserIdAlias ogg_admin
RMTHOST ogg_target, MGRPORT 7909
rmttrail ./dirdat/hh
passthru
SOURCECATALOG oggl
table src_user.*;
```

5. Connect the Data Pump to the local exttrail file:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
phevt, exttrailsource ./dirdat/hu
EXTRACT added.
```

6. Connect the remote trail file to the Data Pump, setting the remote trail file size to 10 MB:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add rmttrail
./dirdat/hh, Extract phevt, megabytes 10
RMTTRAIL added.
```

7. Start the Extract and Data Pump processes and check their status:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start extract
huevt
Sending START request to MANAGER ...
EXTRACT HUEVT starting

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start extract
phevt
Sending START request to MANAGER ...
EXTRACT PHEVT starting

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program      Status       Group        Lag at Chkpt   Time Since
Chkpt

MANAGER      RUNNING
EXTRACT      RUNNING      HUEVT       00:00:00      00:41:39
EXTRACT      RUNNING      PHEVT       00:00:00      00:00:06

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) >
```

8. Open an additional shell window and navigate to the /home/oracle directory. Using an editor of your choice, create the gdp_check.sh shell script, which will be invoked from the event action SHELL by Replicat.

```
[OS prompt]$ cd ~  
[OS prompt]$ pwd  
/home/oracle  
[OS prompt]$ vi gdp_check.sh  
  
#!/bin/bash  
# SHELL gdp_check.sh  
# Invoke sqlplus and call the gdp_check() stored procedure  
# passing the 3 parameters:  
# $1 is GDP_VALUE  
# $2 is GDP_YEAR  
# $3 is CONTINENT  
RET_STATUS=`sqlplus -s src_user/<password>@ogg1 <<EOF  
set pages 0 lines 120 trimout on trimspool on tab off echo off  
set verify off feed off serverout on  
var outvar varchar2(100);  
exec gdp_check($1,$2,'$3',:outvar);  
print outvar;  
exit;  
EOF`  
  
if [ "$RET_STATUS" == "0" ]  
then  
    exit 0  
else  
# Validation check failed. Send an email detailing what  
# provoked the failure  
    echo "Sanity check for $3 in $2 failed!" | mailx -s"GDP  
Report" oracle  
    exit 1  
fi
```

The example uses the vi editor, but if you prefer you can use the graphical editor gedit, which is available in your environment. Make sure to replace the *<password>* entry with the real password. The files for this practice are available at *~/labs/Practice05/configs*.

9. Make the script executable using the `chmod` command:

- ```
[OS prompt]$ chmod +x gdp_check.sh
```
- The shell script receives three parameters:
    - The GDP value being added
    - The calendar year for which the GDP value has been collected
    - The continent where the country belongs

These parameters are passed to the `gdp_check` Oracle stored procedure, which performs its sanity check against the GDP value and returns to the `outvar` output parameter either a zero (if the check was successful) or a one (if the check failed).

- The shell script tests the output value. If it is zero, it exits gracefully, setting 0 as the value to be returned to the OS. If the output value is one, the shell script uses mailx to send an email detailing the values that caused the failure of the sanity check.
10. Before you can test the shell script, the Oracle stored procedures must be created. Use either Oracle SQL Developer or sqlplus to accomplish the task. The first procedure is called `get_contininent` and returns the continent name for a country whose entity ID is passed to the procedure as an input parameter. This procedure is called by `SQLEXEC`, which retrieves the entity ID from the trail file while processing the `GDP_BY_YEAR` records. The stored procedure file is stored in the `~/labs/Practice05/configs` directory. Select the `TRG_USER` window and execute the `get_contininent.sql` script, which creates the `GET_CONTINENT` lookup procedure. Load data into the `ECONOMIC_ENTITY` table by running the `economic_entity.sql` script:

```
SQL> @get_continente.sql
Procedure created.

SQL> @economic_entity.sql
1 row created.

...Many lines omitted for brevity...

Commit;

SQL>
```

The `get_continente` procedure is very simple:

```
create or replace
procedure get_continente(p_entity IN NUMBER, p_cont_out OUT
 VARCHAR2)
IS
BEGIN
 SELECT continent INTO p_cont_out FROM economic_entity WHERE id
 = p_entity;
END;
```

A simple lookup is performed to determine the continent for a country identified by its entity ID.

The second procedure is called `gdp_check` and it requires four parameters, three as input parameters and one as an output parameter. Given a GDP value, the year for which it was collected, and the continent of the country, the procedure checks that the sum of all GDP values for that continent does not exceed 40 trillion. If it does, the output parameter is set to 1; otherwise, it is set to 0.

```
create or replace
procedure check_gdp(p_value IN NUMBER, p_year IN INTEGER, p_cont
 IN VARCHAR2, p_out OUT VARCHAR2)
IS
 sum_gdp NUMBER;
 sql_string VARCHAR2(256);
BEGIN
 IF(p_cont = 'N/A')
```

```

THEN
 p_out := '0';
 RETURN;
END IF;
sql_string :='select sum(a.gdp_value) from gdp_by_year a,
economic_entity b where a.entity_id = b.entity_id and
a.gdp_year='||p_year||' and b.continent = '''||p_cont||'''';
execute immediate sql_string into sum_gdp;
IF(SUM_GDP+p_value > 40000) THEN
 p_out:='1';
ELSE
 p_out:='0';
END IF;
end;

```

To create the stored procedure, select the TRG\_USER window and run the following command:

```

SQL> @gdp_check.sql
Procedure created.

```

- Now that the infrastructure has been created, you can focus on the Oracle GoldenGate Replicat process, which will make use of the stored procedure created in the previous steps. Select the OGG\_TRG window and create the parameter file for the REVTD Replicat:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > edit params revtd
```

```

Replicat revtd
SETENV (ORACLE_SID='CDB12c')
DBOPTIONS INTEGRATEDPARAMS (parallelism 4)
UserIdAlias ogg_repl
AssumeTargetDefs
Discardfile ./dirrpt/revtd.dsc, purge
map ogg1.src_user.gdp_by_year, target ogg2.trg_user.gdp_by_year,
 SQLEXEC(SPNAME trg_user.get_continent, ID lookup,
PARAMS (p_entity =
 ENTITY_ID), EVENTACTIONS (SHELL
 ("/home/oracle/gdp_check.sh $1
 $2 $3", VAR $1 = gdp_value, VAR $2 = gdp_year, VAR $3 =
 lookup.p_cont_out), REPORT);
map ogg1.src_user.gdp_growth_by_year, target
 ogg2.trg_user.gdp_growth_by_year;
map ogg1.src_user.economic_entity, target
 ogg2.trg_user.economic_entity;

```

The key lines are the MAP statements for the GDP\_BY\_YEAR table. The SQLEXEC subcommand forces the get\_continent stored procedure to be called, passing the ENTITY\_ID fetched from the trail file. The output parameter from the stored procedure—the continent name—is then passed to the SHELL subcommand as the third parameter.

The first two parameters for the shell script are taken straight from the trail file, as part of the standard Replicat record processing. Note that the third parameter uses the stored procedure logical name plus the dot notation and the output parameter from the stored procedure (`lookup.p_cont_out`) to pass the continent name to the shell script.

Save the parameter file and, when you are back in GGSCI, add the Replicat, connecting it to `extrail pu`. Then start the REVTD Replicat and make sure it is running:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > add replicat revtd,
integrated exttrail ./dirdat/hh
REPLICAT (Integrated) added.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > start replicat
revtd
Sending START request to MANAGER ...
REPLICAT REVTD starting
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > INFO ALL
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
REPLICAT RUNNING REVTD 00:00:00 00:00:09
```

12. Select the `SRC_USER` window. Perform an insert on the `GDP_BY_YEAR` table, storing a reasonable GDP value for 2008 for Belgium (entity ID 50).

```
SQL> update gdp_by_year set gdp_value = 500.00 where entity_id =
50 and gdp_year = 2008;
1 row updated.

SQL> commit;
Commit complete.
```

13. Select the window where GGSCI is running and enter a view report command on REVTD. Scroll down the report until you see the section on the EVENTACTIONS that occurred while processing the `GDP_BY_YEAR` table:

```
GGSCI ([TRGHOST]) 5> view report revtd
```

```

** Run Time Messages
**

...many lines omitted for clarity...
```

```

2015-03-15 22:28:56 INFO OGG-06505 MAP resolved (entry
ogg1.src_user.gdp_by_year) : map "OGG1"."SRC_USER"."GDP_BY_YEAR",
target ogg2.trg_user.gdp_by_year, SQLEXE
C(SPNAME trg_user.get_continent, ID lookup, PARAMS(p_entity =
ENTITY_ID), EVENTACTIONS(SHELL ("/home/oracle/gdp_check.sh $1 $2
$3", VAR $1 = gdp_value, VAR $2 = gdp_
year, VAR $3 = lookup.p_cont_out), REPORT).

2015-03-15 22:28:58 INFO OGG-01286 Executing shell command
'/home/oracle/gdp_check.sh 500.00000 2008 EUROPE' due to SHELL
event for target table OGG2.TRG_USER.G
DP_BY_YEAR in file ./dirdat/hh000000, RBA 1466.

2015-03-15 22:28:58 INFO OGG-01287 Successfully executed
shell command '/home/oracle/gdp_check.sh 500.00000 2008 EUROPE'.
...many lines omitted for clarity...

```

14. Select the window where `sqlplus` is running. This time update the GDP value for Belgium for the year 2008 to an abnormally high value, and check the report on the `REVTD` Replicat. Then, check your email to see if the procedure triggered by `EVENTACTIONS` alerted you about the sanity check failure:

```

SQL> update gdp_by_year set gdp_value=25000 where entity_id = 50
and gdp_year=2008;
1 row updated.
SQL> commit;

```

15. Select the window where `GGSCI` is running and enter the `view report` command again. This time you should see the abnormal value being passed to the shell script:

```

GGSCI ([TRGHOST]) 6> view report revtd
...many lines omitted for clarity...
2015-03-15 22:34:02 INFO OGG-01286 Executing shell command
'/home/oracle/gdp_check.sh 25000.00000 2008 EUROPE' due to SHELL
event for target table OGG2.TRG_USER
.GDP_BY_YEAR in file ./dirdat/hh000000, RBA 1680.

2015-03-15 22:34:03 INFO OGG-01287 Successfully executed
shell command '/home/oracle/gdp_check.sh 25000.00000 2008
EUROPE'.
...many lines omitted for clarity...

```

16. Select the window where `sqlplus` is running and `exit sqlplus;` launch `mail` to check your email messages. The shell script should have sent you an email that communicates the failure of the data check at the database level:

```
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
[OS prompt]$ mail
Heirloom Mail version 12.4 7/29/08. Type ? for help.
"/var/spool/mail/oracle": 1 message
> 1 oracle@host01.localdomain Sun Mar 15 22:28 19/640 "GDP
Report" &1
```

17. Enter 1 and press the Enter key to read the first email:

```
& 1
Message 1:
From oracle@EDDDR53P1.localdomain Sun Mar 15 22:28:58 2015
Return-Path: <oracle@EDDDR53P1.localdomain>
X-Original-To: oracle
Delivered-To: oracle@EDDDR53P1.localdomain
Date: Sun, 15 Mar 2015 22:28:58 +1100
To: oracle@host01.localdomain
Subject: GDP Report
User-Agent: Heirloom mailx 12.4 7/29/08
Content-Type: text/plain; charset=us-ascii
From: oracle@host01.localdomain
Status: RO

Sanity check for EUROPE in 2008 failed!
> Enter q to quit mail.
q <enter>
```

18. Select the `OGG_TRG` window. Stop and delete the `REVTD` Replicat to clean the environment for the next practice:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > stop replicat revtd
Sending STOP request to REPLICAT REVTD ...
Request processed.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > delete replicat
revtd
2015-03-15 22:43:20 INFO OGG-02529 Successfully
unregistered REPLICAT REVTD inbound server OGG$REVTD from
database.
Deleted REPLICAT REVTD.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) >
```

19. Select the OGG\_SRC window and stop and delete the HUEVT and PHEVT Extract groups. Unregister the HUEVT Extract from the ogg1 database.

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING HUEVT 00:00:00 00:00:03
EXTRACT RUNNING PHEVT 00:00:00 00:00:01

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop er *
Sending STOP request to EXTRACT HUEVT ...
Request processed.

Sending STOP request to EXTRACT PHEVT ...
Request processed.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > delete er * !
Deleted EXTRACT HUEVT.
Deleted EXTRACT PHEVT.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > UNREGISTER
EXTRACT huevt DATABASE
Successfully unregistered EXTRACT HUEVT from database.
```

This completes Practice 5-4. Leave the desktop environment with the four windows running sqlplus and GGSCI up for the next practice.

This completes Practice 5. Stop here.



# **Practices for Lesson 6: Data Mapping, Data Selection/Filtering, and Data Transformation**

## **Chapter 6**

## Practices for Lesson 6: Overview

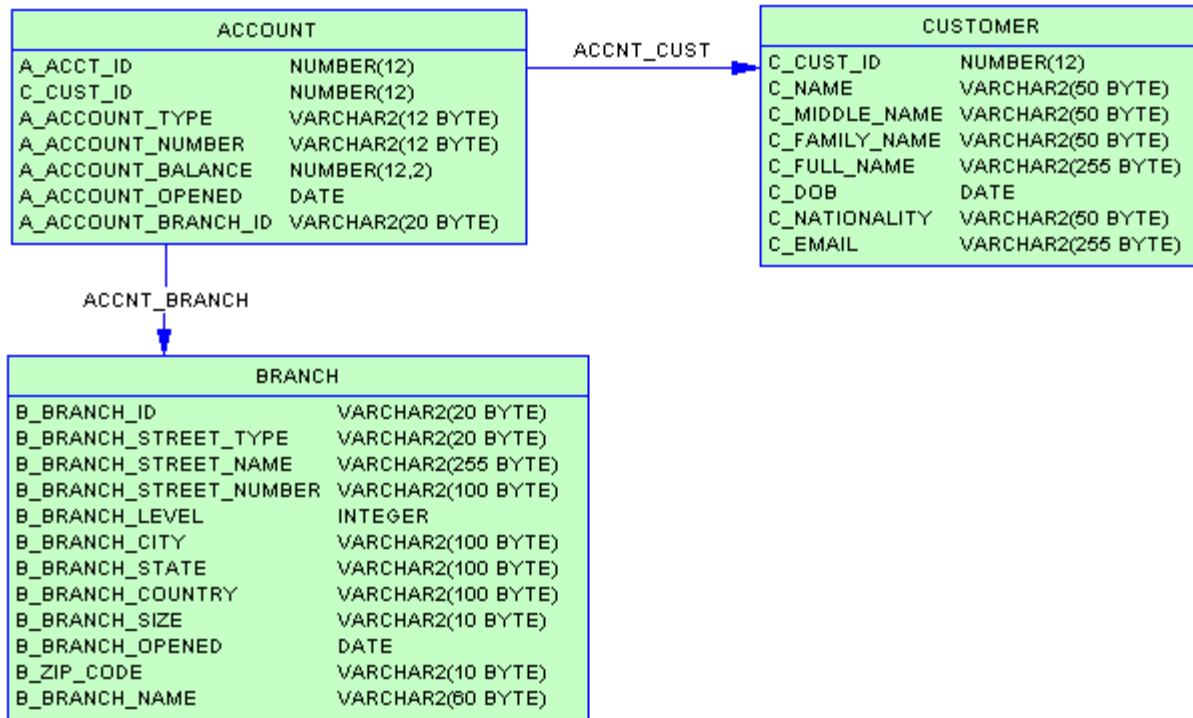
---

### Practices Overview

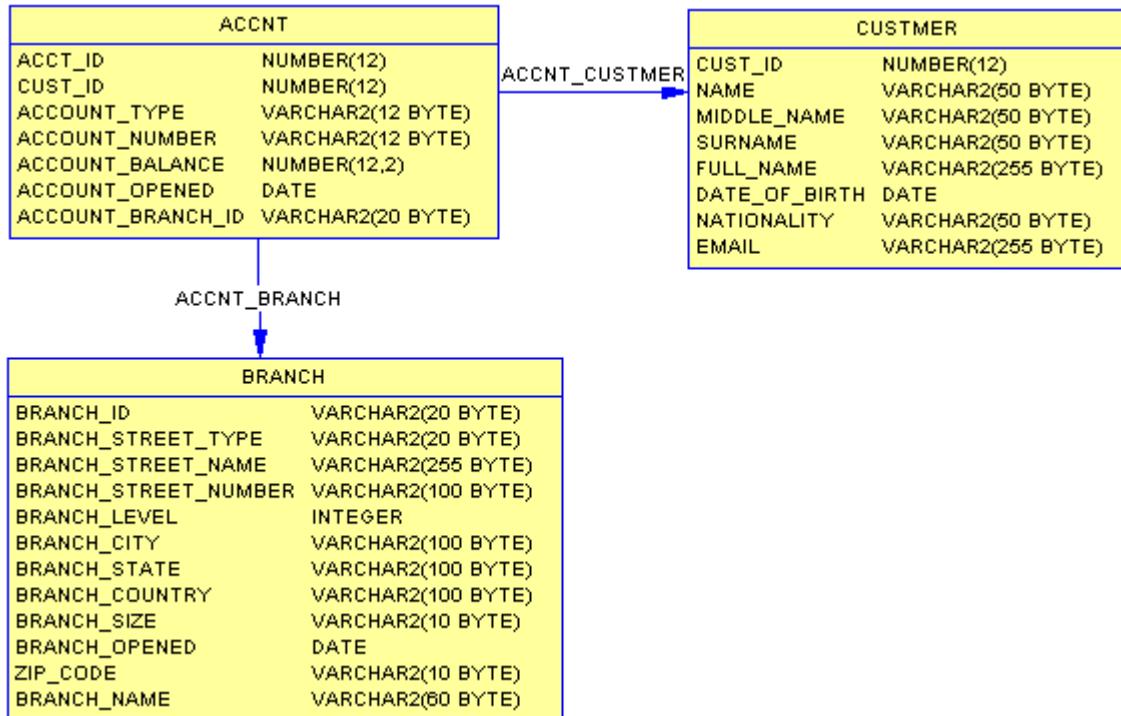
In these practices, you will use Oracle GoldenGate to perform data mapping, data selection and filtering, and data transformation. The table below lists the various commands and functions you will use in Practice 6:

| Purpose             | Oracle GoldenGate Commands/Functions                      |
|---------------------|-----------------------------------------------------------|
| Mapping             | COLMATCH and COLMAP                                       |
| Selection/Filtering | WHERE, FILTER, COLS, and COLSEXCEPT                       |
| Transformation      | @STREQ, @STRCAT, @STREXT, @COLTEST, @STRFIND, and SQLEXEC |

In this practice, you create three tables in the source database: ACCOUNT, CUSTOMER, and BRANCH. The Entity/Relationship (E/R) diagram of the tables you are going to create in the SRC\_USER schema is shown below:



In the TRG\_USER schema, you will create similar tables called ACCNT, CUSTMER, and BRANCH.



Although the three tables are similar, two of them are named differently and their column names are also slightly different. Oracle GoldenGate, however, provides several commands to facilitate the mapping of different table and column names, as well as facilities for transforming data while mapping is being applied.

**Note:** When the source table/column definitions differ from the target, the `defgen` utility must be used.

**Helper scripts provided in `/home/oracle/labs/Practice06/configs`:**

| Catch-up/Clean-up script            | Purpose                                                                    |
|-------------------------------------|----------------------------------------------------------------------------|
| <code>catch-up-prac-6_1.sh</code>   | Executes steps for Practice 6-1                                            |
| <code>catch-up-prac-6_2.sh</code>   | Executes steps for Practice 6-2                                            |
| <code>catch-up-prac-6_3.sh</code>   | Executes steps for Practice 6-3                                            |
| <code>catch-up-prac-6_4.sh</code>   | Executes steps for Practice 6-4                                            |
| <code>clean-up-prac-6_1.sh</code>   | Undoes all steps performed during Practice 6-1                             |
| <code>clean-up-prac-6_2.sh</code>   | Undoes all steps performed during Practice 6-2                             |
| <code>clean-up-prac-6_3.sh</code>   | Undoes all steps performed during Practice 6-3                             |
| <code>clean-up-prac-6_4.sh</code>   | Undoes all steps performed during Practice 6-4                             |
| <code>lab-6-total-cleanup.sh</code> | Executes the steps necessary to wipe out all objects created in Practice 6 |

**Note:** At the end of Practice 6, you should run the `lab-6-total-cleanup.sh` script.

## Preliminary Tasks

Before starting the practice per se, you must create the required infrastructure such as the tables which have similar but not identical column names and size across the replication source and the replication target.

- Select the SRC\_USER window and exit sqlplus. Navigate to the ~/labs/Practice06/configs directory, list the files in the directory, start sqlplus, and execute the src\_user.sql script to create the three tables: ACCOUNT, CUSTOMER, and BRANCH:

```
[OS prompt]$ cd ~/labs/Practice06/configs/
[OS prompt]$ ls
account_1.sql catch-up-prac-6_1.sh clean-up-prac-6_3.sh
customer.sql account_add.sql catch-up-prac-6_2.sh clean-up-prac-
6_4.sh lab-6-total-cleanup.sh account.sql catch-up-prac-6_3.sh
constraints_disable.sql src_user.sql branch_1.sql catch-up-
prac-6_4.sh constraints_enable.sql toggle_constraints.sql
branch_add.sql clean-up-prac-6_1.sh customer_1.sql
trg_user.sql branch.sql clean-up-prac-6_2.sh customer_add.sql
[OS prompt]$ sqlplus src_user/<password>@ogg1
SQL*Plus: Release 12.1.0.2.0 Production on Sun Mar 15 09:27:02
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Sat Mar 14 2015 17:26:18 +11:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> @src_user.sql
Table created.
...many lines omitted for clarity...
SQL>
```

- Describe the tables that you just created to make sure the script ran correctly:

| SQL> desc account   |          |              |
|---------------------|----------|--------------|
| Name                | Null?    | Type         |
| A_ACCT_ID           | NOT NULL | NUMBER(12)   |
| C_CUST_ID           | NOT NULL | NUMBER(12)   |
| A_ACCOUNT_TYPE      | NOT NULL | VARCHAR2(12) |
| A_ACCOUNT_NUMBER    | NOT NULL | VARCHAR2(12) |
| A_ACCOUNT_BALANCE   | NOT NULL | NUMBER(12,2) |
| A_ACCOUNT_OPENED    | NOT NULL | DATE         |
| A_ACCOUNT_BRANCH_ID | NOT NULL | VARCHAR2(20) |

```
SQL> desc branch
Name Null? Type

B_BRANCH_ID NOT NULL VARCHAR2(20)
B_BRANCH_STREET_TYPE NOT NULL VARCHAR2(20)
B_BRANCH_STREET_NAME NOT NULL VARCHAR2(255)
B_BRANCH_STREET_NUMBER NOT NULL VARCHAR2(100)
B_BRANCH_LEVEL NUMBER(38)
B_BRANCH_CITY NOT NULL VARCHAR2(100)
B_BRANCH_STATE NOT NULL VARCHAR2(100)
B_BRANCH_COUNTRY NOT NULL VARCHAR2(100)
B_BRANCH_SIZE NOT NULL VARCHAR2(10)
B_BRANCH_OPENED NOT NULL DATE
B_ZIP_CODE NOT NULL VARCHAR2(10)
B_BRANCH_NAME NOT NULL VARCHAR2(60)

SQL> desc customer
Name Null? Type

C_CUST_ID NOT NULL NUMBER(12)
C_NAME NOT NULL VARCHAR2(50)
C_MIDDLE_NAME VARCHAR2(50)
C_FAMILY_NAME NOT NULL VARCHAR2(50)
C_FULL_NAME NOT NULL VARCHAR2(255)
C_DOB NOT NULL DATE
C_NATIONALITY NOT NULL VARCHAR2(50)
C_EMAIL VARCHAR2(255)
SQL>
```

3. Select the TRG\_USER window and exit sqlplus. Navigate to the ~/labs/Practice06/configs directory, restart sqlplus, connecting to the replication target environment (trg\_user@ogg2), and run the trg\_user.sql SQL script:

```
SQL> connect trg_user/<password>@ogg2
Connected.
SQL> show user
USER is "TRG_USER"
SQL> @trg_user.sql

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

4. Describe the tables that you just created to make sure the `trg_user.sql` script ran correctly:

```
SQL> desc acctnt
Name Null? Type

ACCT_ID NOT NULL NUMBER(12)
CUST_ID NOT NULL NUMBER(12)
ACCOUNT_TYPE NOT NULL VARCHAR2(12)
ACCOUNT_NUMBER NOT NULL VARCHAR2(12)
ACCOUNT_BALANCE NOT NULL NUMBER(12,2)
ACCOUNT_OPENED NOT NULL DATE
ACCOUNT_BRANCH_ID NOT NULL VARCHAR2(20)

SQL> desc custmer
Name Null? Type

CUST_ID NOT NULL NUMBER(12)
NAME NOT NULL VARCHAR2(50)
MIDDLE_NAME VARCHAR2(50)
SURNAME NOT NULL VARCHAR2(50)
FULL_NAME NOT NULL VARCHAR2(255)
DATE_OF_BIRTH NOT NULL DATE
NATIONALITY NOT NULL VARCHAR2(50)
EMAIL VARCHAR2(255)

SQL> desc branch
Name Null? Type

BRANCH_ID NOT NULL VARCHAR2(20)
BRANCH_STREET_TYPE NOT NULL VARCHAR2(20)
BRANCH_STREET_NAME NOT NULL VARCHAR2(255)
BRANCH_STREET_NUMBER NOT NULL VARCHAR2(100)
BRANCH_LEVEL NUMBER(38)
BRANCH_CITY NOT NULL VARCHAR2(100)
BRANCH_STATE NOT NULL VARCHAR2(100)
BRANCH_COUNTRY NOT NULL VARCHAR2(100)
BRANCH_SIZE NOT NULL VARCHAR2(10)
BRANCH_OPENED NOT NULL DATE
ZIP_CODE NOT NULL VARCHAR2(10)
BRANCH_NAME NOT NULL VARCHAR2(60)

SQL>
```

## Practice 6-1: Column Mapping Between Dissimilar Source and Target Tables

### Overview

In this practice you configure a Replicat process, which takes care of all differences between the source and target tables.

### Assumptions

The GGSCI command “ADD SCHEMATRANDATA src\_user ALLCOLS” has been issued and schema-level supplemental logging for all current and future tables has been enabled. You should have issued this command in Practice 2-2, step 5.

### Tasks

This is an overview of the steps that will be explained in detail in the Solution section of this practice. Try to perform the tasks without looking at the solution first. The solution is provided to show you one way to accomplish the goals for the practice, but it is not necessarily the only way or the better way.

1. Run the `defgen` utility to generate the mapping file in the `dirdef` subdirectory.
2. Create an Extract group called `EMAP` that will extract all rows from the `SRC_USER` schema.
3. Using a combination of `COLMATCH` and `COLMAP` statements, create a Data Pump Extract called `PMAP` that can map all source table/column definitions to their respective targets.
4. Define a Replicat group (`RMAP`) that will store the mapped column in the target schema.
5. After starting the `RMAP` Replicat process, select the `SRC_USER` window and run the `account.sql`, `branch.sql`, and `customer.sql` scripts. Then select the `TRG_USER` window and verify that all data has been correctly replicated across the source and target tables.

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

## Solution 6-1: Column Mapping Between Dissimilar Source and Target Tables

---

### Steps

1. Select the OGG\_SRC window, where GGSCI is running connected to the OGG\_SRC Oracle GoldenGate instance. Use the DBLOGIN command to log in to the replication source database and enter the INFO TRANDATA command for all tables in the SRC\_USER schema in the ogg1 PDB. Verify that the three tables ACCOUNT, CUSTOMER, and BRANCH appear in the list:

```
GGSCI ([SRCHOST]) > DBLogin UserIdAlias ogg_admin
Successfully logged into database.
GGSCI ([HOST]) as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info trandata
ogg1.src_user.*

...many lines omitted for clarity...

Logging of supplemental redo log data is enabled for table
OGG1.SRC_USER.ACOUNT.

Columns supplementally logged for table OGG1.SRC_USER.ACOUNT:
A_ACCOUNT_BALANCE, A_ACCOUNT_BRANCH_ID, A_ACCOUNT_NUMBER,
A_ACCOUNT_OPENED, A_ACCOUNT_TYPE, A_ACCT_ID, C_CUST_ID.

...many lines omitted for clarity...

GGSCI ([HOST]) as C##OGG_ADMIN@CDB12c/CDB$ROOT) >
```

2. Create the parameter file for the EMAP Extract group:

```
GGSCI ([SRCHOST]) 1> edit param emap
Extract emap
UserIdAlias ogg_admin
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
Exttrail ./dirdat/mp
SOURCECATALOG ogg1
Table src_user.*;
```

3. Save the parameter file. Add the Extract and connect it to the trail file (./dirdat/mp):

```
GGSCI ([HOST]) as C##OGG_ADMIN@CDB12c/CDB$ROOT) > REGISTER
EXTRACT emap DATABASE CONTAINER (ogg1)
Extract EMAP successfully registered with database at SCN
2436867.
GGSCI ([HOST]) as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
emap, integrated tranlog, begin now
EXTRACT added.
```

```
GGSCI ([HOST]) as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add exttrail
./dirdat/mp, extract emap, megabytes 10
EXTTRAIL added.
```

4. Create a defgen parameter that will be run through the defgen utility to create the mapping file ./dirdef/usr\_tgt.def.

```
GGSCI ([HOST]) as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit param
defgen
defsfile ./dirdef/usr_tgt.def, purge
UserIdAlias ogg_admin
SOURCECATALOG ogg2
Table trg_user.*;
```

5. Exit GGSCI and invoke the defgen utility.

```
GGSCI ([SRCHOST]) 5> exit
[OS prompt]$./defgen paramfile ./dirprm/defgen.prm
```

```

*
 Oracle GoldenGate Table Definition Generator for Oracle
 Version 12.1.2.1.0
 OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1
 Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
 03:26:23
 Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
 rights reserved.

 Starting at 2015-03-15 10:02:24

 ...many lines omitted for clarity...

 Retrieving definition for OGG2.TRG_USER.ACCTN.
 Retrieving definition for OGG2.TRG_USER.ACOUNT.
 Retrieving definition for OGG2.TRG_USER.BRANCH.
 Retrieving definition for OGG2.TRG_USER.CUSTMER.
 Retrieving definition for OGG2.TRG_USER.CUSTOMER.

 ...many lines omitted for clarity...
```

6. Relaunch GGSCI and edit the parameter file for the PMAP Data Pump Extract:

```
[OS prompt]$ ggsci
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
10:21:34
Operating system character set identified as UTF-8.
Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
rights reserved.
```

```
GGSCI ([SRCHOST]) 1> edit param pmap
Extract pmap
RmtHost ogg_target, Mgrport 7909, Compress
RmtTrail ./dirdat/vv
TargetDefs ./dirdef/usr_tgt.def
UserIdAlias ogg_admin
COLMATCH PREFIX A_
Map ogg1.src_user.account, target ogg2.trg_user.acct, COLMAP
(usedefaults, CUST_ID = C_CUST_ID);
COLMATCH PREFIX B_
Map ogg1.src_user.branch, target ogg2.trg_user.branch, COLMAP(
USEDDEFAULTS);
COLMATCH PREFIX C_
Map ogg1.src_user.customer, target ogg2.trg_user.customer, COLMAP
(USEDDEFAULTS, SURNAME = C_FAMILY_NAME, DATE_OF_BIRTH = C_DOB);
```

7. Save the parameter file.

In this case, the Data Pump Extract carries out most of the transformational work. Note that even if the COLMATCH prefix directive has been used, the name of the source column specified through the COLMAP directive must still be provided with the prefix.

8. Use DBLogin to connect to the database as the Oracle GoldenGate administrative user:

```
GGSCI ([SRCHOST]) > DBLogin UserIdAlias ogg_admin
Successfully logged into database CDB$ROOT.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) >
```

9. Add the PMAP Data Pump, connecting it with its trail file, and then add the remote trail file:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
pmap, exttrailsource ./dirdat/mp
EXTRACT added.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add rmttrail
./dirdat/vv, extract pmap, megabytes 10
RMTTRAIL added.
```

10. Start the primary Extract EMAP and the Data Pump PMAP:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start emap
Sending START request to MANAGER ...
EXTRACT EMAP starting

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start pmap
Sending START request to MANAGER ...
EXTRACT PMAP starting
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING EMAP 00:26:00 00:00:07
EXTRACT RUNNING PMAP 00:00:00 00:01:04
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) >
```

11. Select the OGG\_TRG window, where GGSCI is connected to the Oracle GoldenGate target environment. Use DBLogin to connect to the database by using the `ogg_repl` alias. Edit the parameter file for the RMAP Replicat group:

```
GGSCI ([TRGHOST]) > dblogin UserIdAlias ogg_repl
Successfully logged into database OGG2.
```

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > edit param rmap
```

```
Replicat rmap
Discardfile ./dirrpt/rmap.dsc, purge
UserIdAlias ogg_repl
assumetargetdefs
Map ogg2.trg_user.accnt, Target ogg2.trg_user.accnt,
handlecollisions;
Map ogg2.trg_user.branch, Target ogg2.trg_user.branch,
handlecollisions;
Map ogg2.trg_user.custmer, Target ogg2.trg_user.custmer,
handlecollisions;
```

Note that the mapping occurs between `trg_user` and `trg_user`. It appears counterintuitive, but it makes sense. The mapping occurred in the Data Pump Extract. The transformations between the `src_user` table definitions and the `trg_user` table definitions occur on the data capture side. By the time the trail file is transferred to the target host, the transformations are already stored in `trg_user`.

12. Save the parameter file. Add the Replicat, connecting it to its trail file (./dirdat/vv):

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > add rep rmap,
integrated exttrail ./dirdat/vv
REPLICAT (Integrated) added.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > start rmap
Sending START request to MANAGER ...
REPLICAT RMAP starting
```

13. Leave the GGSCI session open. Open a new terminal shell or reuse an open shell window where you can use sqlplus. Change the directory to ~/labs/Practice06/configs. Connect to the SRC\_USER schema by using sqlplus and load the CUSTOMER, ACCOUNT, and BRANCH tables by using the customer.sql, account.sql, and branch.sql scripts:

```
[OS prompt]$ cd ~/labs/Practice06/configs
[OS prompt]$ sqlplus src_user/<password>@ogg1
SQL> @account.sql
1 row created.
... Many lines omitted for clarity ...
Commit complete.
SQL> @customer.sql
1 row created.
... Many lines omitted for clarity ...
Commit complete.
SQL> @branch.sql
1 row created.
... Many lines omitted for clarity ...
Commit complete.
SQL> select count(*) from account;
COUNT(*)

179
SQL> select count(*) from customer;

COUNT(*)

99
SQL> select count(*) from branch;

COUNT(*)

4
```

14. Without leaving `sqlplus`, connect to `TRG_USER` and check that the `ACCNT`, `CUSTMER`, and `BRANCH` tables have been populated, after the data transformation has been performed. You should find 179 account entries, 99 customer entries, and 4 branch entries in the `ACCNT`, `CUSTMER`, and `BRANCH` tables—the same number as the replication source.
15. Select the `OGG_TRG` window, where `GGSCI` is connected to the Oracle GoldenGate replication target environment and issue the `stats rmap` command to verify that replication in fact occurred.
16. After you have verified that the transformations have been successfully carried out, you must stop and delete the Extract and Replicat groups (`EMAP`, `PMAP`, and `RMAP`) and unregister the `EMAP` Extract from the database. Alternatively, you can run the `clean-up-prac-6_1.sh` file found in `~/labs/Practice06/configs`. You must `TRUNCATE` the `ACCOUNT` table in the `SRC_USER` schema (`ogg1 PDB src_user schema`).

## Practice 6-2: Data Selection in Extract and Replicat Streams

### Overview

In this practice, you use a combination of WHERE and FILTER clauses to select specific data during replication between the source schema SRC\_USER and the target schema TRG\_USER.

### Assumptions

You have access to the SRC\_USER and TRG\_USER schemas on the CDB12c database (ogg1 PDB and ogg2 PDB, respectively). A table called ACCOUNT exists and is identically defined in the source and target schemas. The table ACCOUNT in the SRC\_USER schema has been truncated at the end of Practice 6-1 and has 0 rows stored in it.

### Tasks

1. You use the Oracle GoldenGate instance located in /u03/ogg/ogg\_src to configure a primary Extract group called ESTRM, which extracts only CHECKING accounts from the ACCOUNT table in the SRC\_USER Oracle schema (ogg1 PDB).
2. The trail file (./dirdat/es) will then be read by a Data Pump called PSTRM, which will create a remote trail (./dirdat/rs) that is used to transfer the extracted data to the remote system.
3. You will then connect to the Oracle GoldenGate instance located in /u03/ogg/ogg\_trg and configure a Replicat group called RSTRM, which will filter out all the accounts that have less than \$1,000 as balance. You will store the account rows that pass through the filter into the ACCOUNT table in the TRG\_USER schema (ogg2 PDB).
4. To test your filtering criteria, navigate to the ~/labs/Practice06/configs directory and connect to the source schema (src\_user@ogg1). Load the table by using the account.sql script. To verify that data selection works, connect to the target schema by using sqlplus and verify that only CHECKING accounts holding more than \$1,000 as balance are replicated to the target schema.

This completes Practice 6-2. Continue with Practice 6-3.

## Solution 6-2: Data Selection in Extract and Replicat Streams

### Steps

- Select the OGG\_SRC window, where GGSCI runs connected to the Oracle GoldenGate OGG\_SRC instance. Connect to the CDB12c database and create the parameter file for the ESTRM Extract group:

```
GGSCI ([SRCHOST]) > DBLogin UserIdAlias ogg_admin
Successfully logged into database.
GGSCI ([HOST]) as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit param
estrm
Extract estrm
ExtTrail ./dirdat/es
UserIdAlias ogg_admin
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
SOURCECATALOG ogg1
Table src_user.account, WHERE (A_ACCOUNT_TYPE = 'CHECKING');
```

- Save the parameter file. Register the ESTRM Extract with the database (ogg1). Add the Extract and its corresponding trail file:

```
GGSCI ([HOST]) as C##OGG_ADMIN@CDB12c/CDB$ROOT) > REGISTER
EXTRACT estrm DATABASE CONTAINER (ogg1)
Extract ESTRM successfully registered with database at SCN
2497729.
GGSCI ([SRCHOST]) > add extract estrm, integrated tranlog, begin
now
EXTRACT added.
GGSCI ([SRCHOST]) > add exttrail ./dirdat/es, Extract estrm,
megabytes 10
EXTTRAIL added.
```

- Edit the parameter file for the PSTRM Data Pump Extract:

```
GGSCI ([SRCHOST]) > edit param pstrm
```

```
Extract pstrm
RmtHost ogg_target, Mgrport 7909, Compress
RmtTrail ./dirdat/rs
Passthru
SOURCECATALOG ogg1
Table src_user.account;
```

- Save the parameter file and add the Data Pump, connecting it to the ./dirdat/es trail file:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
pstrm, exttrailsource ./dirdat/es
EXTRACT added.
```

5. Add the remote trail file and start the Extract and the Data Pump:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add rmttrail
./dirdat/rs, extract pstrm, megabytes 10
RMTTRAIL added.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start estrm
Sending START request to MANAGER ...
EXTRACT ESTRM starting

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start pstrm
Sending START request to MANAGER ...
EXTRACT PSTRM starting
```

6. Select the OGG\_TRG window, where GGSCI runs connected to the OGG\_TRG Oracle GoldenGate instance, and edit the parameter file for the RSTRM Replicat group:

```
GGSCI ([TRGHOST]) > dblogin UserIdAlias ogg_repl
Successfully logged into database OGG2.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > edit param rstrm
```

```
Replicat rstrm
AssumeTargetDefs
DiscardFile ./dirrpt/rstrm.dsc, Purge
UserIdAlias ogg_repl
Map ogg1.src_user.account, target ogg2.trg_user.account, FILTER
(A_ACCOUNT_BALANCE > 1000);
```

7. Save the parameter file and add the Replicat, connecting it to the ./dirdat/es trail file. Use the NODBCHECKPOINT clause to avoid creating a checkpoint table for this practice:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > add replicat rstrm,
Integrated exttrail ./dirdat/rs
REPLICAT (Integrated) added.
```

8. Start the RSTRM Replicat group:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > start rstrm
Sending START request to MANAGER ...
REPLICAT RSTRM starting

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > info all

Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
REPLICAT RUNNING RSTRM 00:00:00 00:00:07

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) >
```

9. Open a new shell window and change the directory to ~/labs/Practice06/configs. Connect to the SRC\_USER schema (PDB ogg1) by using sqlplus, and load the ACCOUNT table using the account.sql script:

```
[OS prompt]$ cd ~/labs/Practice06/configs
[OS prompt]$ sqlplus src_user/<password>@ogg1
SQL*Plus: Release 12.1.0.2.0 Production on Sun Mar 15 10:28:35
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Sun Mar 15 2015 09:27:02 +11:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> @account.sql
1 row created.
... Many lines omitted for clarity ...
Commit complete.
```

10. Without leaving sqlplus, connect to the TRG\_USER schema and verify that the replicated rows have been filtered according to the selection criteria specified in the Extract and Replicat groups:

```
SQL> connect trg_user/<password>@ogg2
Connected.
SQL> select count(*) from account;

COUNT(*)

82
SQL> select * from account where a_account_balance < 1000;
no rows selected
SQL> select * from account where a_account_type = 'SAVINGS';
no rows selected
```

11. Only 82 rows were replicated. The source ACCOUNT table contains 179 rows. Only checking accounts with a balance of more than \$1,000 were replicated to the target.
12. After you verify that only the required rows were replicated to the target, stop and delete all Extract and Replicat groups (ESTRM, PSTRM, and RSTRM). Unregister the ESTRM Extract from the database. You must TRUNCATE the CUSTOMER table in the SRC\_USER schema (ogg1 PDB src\_user schema). Alternatively, you can run the clean-up-prac-6\_2.sh file found in ~/labs/Practice06/configs to clean up the replication environment.

## Practice 6-3: Applying Data Transformation Functions

### Overview

In this practice, you use Oracle GoldenGate transformation functions to perform string concatenation, composing a full name from a first name, a middle name, and a family name. The source CUSTOMER table contains the columns C\_NAME, C\_MIDDLE\_NAME, and C\_FAMILY\_NAME. The C\_FULL\_NAME column defined in the target CUSTOMER table will store the result of the string concatenation operation by the Replicat group during the replication process.

### Assumptions

You have access to the SRC\_USER and TRG\_USER schemas on the CDB12C database. A table called CUSTOMER exists and is identically defined in the source and target schemas.

### Tasks

1. You will use the Oracle GoldenGate instance located in /u03/ogg/ogg\_src to configure a primary Extract group called ETRNF, which extracts all rows from the CUSTOMER table in the SRC\_USER Oracle schema.
2. The trail file (./dirdat/ts) will then be read by a Data Pump called PTRNF, which will create a remote trail (./dirdat/ws) that is used to transfer the extracted data to the remote system.
3. You will then connect to the Oracle GoldenGate instance located in /u03/ogg/ogg\_trg and configure a Replicat group called RTRNF, which will concatenate the name, the middle name, and the family name into the C\_FULL\_NAME column.
4. To test your transformation function, navigate to the ~/labs/Practice06/configs directory and connect to the source schema (src\_user). Truncate the CUSTOMER table and reload it using the customer.sql script. To verify that data selection works, connect to the target schema by using sqlplus and verify that the C\_FULL\_NAME column contains the correct concatenation of first, middle, and family names.

This completes Practice 6-3. Continue with Practice 6-4.

## Solution 6-3: Applying Data Transformation Functions

### Steps

- Select the OGG\_SRC window, where GGSCI is connected to the OGG\_SRC Oracle GoldenGate instance, and create the parameter file for the ETRNF Extract group:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit param etrnf
```

```
Extract etrnf
ExtTrail ./dirdat/ts
UserIdAlias ogg_admin
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
SOURCECATALOG ogg1
Table src_user.customer;
Table src_user.account;
```

- Save the parameter file. Register the ETRNF Extract with the database (ogg1). Add the Extract and its corresponding trail file:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > REGISTER EXTRACT
etrnf DATABASE CONTAINER (ogg1)
```

Extract ETRNF successfully registered with database at SCN  
2541784.

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
etrnf, Integrated tranlog, begin now
EXTRACT added.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add exttrail
./dirdat/ts, Extract etrnf, megabytes 10
EXTTRAIL added.
```

- Edit the parameter file for the PTRNF Data Pump Extract:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit param ptrnf
```

```
Extract ptrnf
RmtHost ogg_target, Mgrport 7909, Compress
RmtTrail ./dirdat/ws
Passthru
SOURCECATALOG ogg1
Table src_user.customer;
Table src_user.account;
```

- Save the parameter file and add the Data Pump, connecting it to the ./dirdat/es trail file:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
ptrnf, exttrailsource ./dirdat/ts
EXTRACT added.
```

5. Add the remote trail file and start the Extract and the Data Pump:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add rmttrail
./dirdat/ws, extract ptrnf, megabytes 10
RMTTRAIL added.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start etrnf
Sending START request to MANAGER ...
EXTRACT ETRNF starting

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start ptrnf
Sending START request to MANAGER ...
EXTRACT PTRNF starting
```

6. Select the OGG\_TRG window, where GGSCI is connected to the replication target environment. Edit the parameter file for the RTRNF Replicat group:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > edit param rtrnf
```

```
Replicat rtrnf
AssumeTargetDefs
DiscardFile ./dirrpt/rtrnf.dsc, Purge
UserIDAlias ogg_repl
Map ogg1.src_user.customer, target ogg2.trg_user.customer,
COLMAP (USEDDEFAULTS, C_FULL_NAME = @STRCAT(C_NAME, ' ',
C_MIDDLE_NAME, ' ', C_FAMILY_NAME));
```

7. Save the parameter file and add the Replicat, connecting it to the ./dirdat/ws trail file. Use the NODBCHECKPOINT clause to avoid creating a checkpoint table for this practice:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > add replicat rtrnf,
Integrated exttrail ./dirdat/ws
REPLICAT (Integrated) added.
```

8. Start the Replicat and exit GGSCI.

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > start rtrnf
Sending START request to MANAGER ...
REPLICAT RTRNF starting

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
REPLICAT RUNNING RTRNF 00:00:00 00:00:07

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > exit
```

9. Change the directory to ~/labs/Practice06/configs and connect to the SRC\_USER schema by using sqlplus. Reload the CUSTOMER table by using the customer.sql script:

```
[OS prompt]$ cd ~/labs/Practice06/configs
[OS prompt]$ sqlplus src_user/<password>@ogg1
SQL*Plus: Release 12.1.0.2.0 Production on Sun Mar 15 10:28:35
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Sun Mar 15 2015 09:27:02 +11:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> @customer.sql
1 row created.
... Many lines omitted for clarity ...
Commit complete.
```

10. Without leaving sqlplus, connect to the TRG\_USER schema and verify that the replicated rows in the CUSTOMER table have a full name, which is the result of the string concatenation of the name, middle name, and family name:

```
SQL> connect trg_user/<password>@ogg2
Connected.
SQL> select c_full_name from customer;
C_FULL_NAME

Cameron Neville Cantu
Austin Kennan Alston
Orson Wyatt Powers
Arsenio Omar Hampton
Addison Baxter Thomas
Chester Nissim Mendoza
Amos Jackson Greer
Ferris Wayne Atkins
Benedict Travis Carey
Evan Quentin McGowan
Vladimir Buckminster Finley
... Many lines omitted for clarity ...
```

11. Leave the Replicat group (RTRNF) and the Extract groups ETRNF and PTRNF running because they will be used in Practice 6-4.

## Practice 6-4: Using SQLEXEC to Invoke a Stored Procedure

### Overview

In this practice, you use `SQLEXEC` as a stand-alone statement at the root level of a parameter file to execute a SQL stored procedure or query. In the practices for Lesson 5, “Oracle GoldenGate Event Marker System,” you used `SQLEXEC` as a parameter to a `MAP` statement for a Replicat group.

As a stand-alone statement, `SQLEXEC` executes independently of a `TABLE` or `MAP` statement during Oracle GoldenGate processing. Stand-alone `SQLEXEC` can therefore be used to invoke a stored procedure (or to execute a query), which performs, for instance, preliminary steps to prepare the target database for the incoming replicated rows.

### Assumptions

- You have access to the `src_user` and `trg_user` accounts on the Oracle RDBMS CDB12c, and the three tables `ACCNT`, `BRANCH`, and `CUSTMER` exist in the `TRG_USER` schema.
- Practice 6-3 has been successfully completed, and the Extract groups `ETRNF` and `PTRNF` and the Replicat group `RTRNF` are running.

### Tasks

1. Open a Gnome terminal shell (or reuse a terminal shell in your environment) and navigate to the `~/labs/Practice06/configs` directory.
2. Create a procedure, called `CONSTRAINTS_DISABLE`, which disables the primary key constraints for the `ACCNT`, `BRANCH`, and `CUSTMER` tables defined in the `TRG_USER` schema.
3. Modify the `RTRNF` Replicat group configured in Practice 6-3 and add a call to `SQLEXEC` to disable the constraints before loading the `ACCOUNT` table. To test the stand-alone `SQLEXEC` statement, connect to `SRC_USER` and truncate the `ACCOUNT` table. Reload it by using the `account.sql` script, and display the report for the `RTRNF` Replicat to verify that `SQLEXEC` was called.

This completes Practice 6-4.

This completes Practice 6. Stop here.

## Solution 6-4: Using SQL\*EXEC to Invoke a Stored Procedure

### Steps

1. Use your favorite editor to edit the CONSTRAINTS\_DISABLE procedure. Change directory to /home/oracle/labs/Practice06/configs and edit the constraints\_disable.sql file. Understand what the stored procedure does:

```
[OS prompt]$ cd ~/labs/Practice06/configs
[OS prompt]$ vi constraints_disable.sql
```

```
create or replace PROCEDURE constraints_disable IS
begin
for cur in (select owner,constraint_name, table_name from
user_constraints where constraint_type = 'P' and table_name in
('ACCOUNT','BRANCH','CUSTOMER')) loop
 execute immediate 'ALTER TABLE
'||cur.owner||'.'||cur.table_name||' MODIFY CONSTRAINT
'||cur.constraint_name||' DISABLE';
end loop;
end;
/
```

2. Exit the editor. Connect to sqlplus by using the TRG\_USER account. Create the CONSTRAINTS\_DISABLE stored procedure by invoking the constraints\_disable.sql script:

```
[OS prompt]$ sqlplus trg_user/<password>@ogg2
SQL*Plus: Release 12.1.0.2.0 Production on Sun Mar 15 10:28:35
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Sun Mar 15 2015 09:27:02 +11:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> @constraints_disable.sql
Procedure created.
SQL>
```

3. Select the OGG\_TRG window, launch GGSCI and connect to the target database using the ogg\_repl alias. Edit the parameter file for the RTRNF Replicat group:

```
[OS prompt] $ ggsci
Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
10:21:34
Operating system character set identified as UTF-8.
```

```
Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
rights reserved.

GGSCI ([TRGHOST]) > dblogin UserIdAlias ogg_repl
Successfully logged into database OGG2.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > edit param rtrnf
Replicat rtrnf
AssumeTargetDefs
DiscardFile ./dirrpt/rtrnf.dsc, Purge
UserIDAlias ogg_repl
SQLEXEC 'CALL trg_user.CONSTRAINTS_DISABLE ()'
Map ogg1.src_user.account, target ogg2.trg_user.account, FILTER
(A_ACCOUNT_BALANCE > 1000);
```

- Save the parameter file. When you are back in GGSCI, stop the Replicat group and restart it, to make it reload the modified parameter file:

```
GGSCI ([Host]) 2> stop rtrnf
Sending STOP request to REPLICAT RTRNF ...
Request processed.

GGSCI ([Host]) 3> start rtrnf
Sending START request to MANAGER ...
REPLICAT RTRNF starting
GGSCI ([Host]) 4> info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
REPLICAT RUNNING RTRNF 00:00:00 00:00:04
```

- Exit GGSCI and change the directory to ~/labs/Practice06/configs. Connect to the SRC\_USER schema and truncate the ACCOUNT table. Use the account.sql script to reload it. Repeat the sequence twice. The replication framework put in place for this practice replicates only DML statements, so the TRUNCATE statement is not executed on the replication target (trg\_user). When you truncate the ACCOUNT table and reload it for the second time, the TRUNCATE statement has effect only on SRC\_USER. Reloading the table for the second time would cause a primary key constraint violation on the target system. Because the Replicat process executes the stored procedure that disables primary key constraints before applying the changes to the ACCOUNT table, no constraint violation occurs.

**IMPORTANT:** The primary key constraint for the ACCOUNT table, TRG\_ACCOUNT\_PK, must be defined as DEFERRABLE INITIALLY DEFERRED, otherwise the unique index associated to the primary key is updated by Oracle and the constraint appears to be still active even if it was disabled by the stored procedure. You may want to check that the primary key constraint is defined as DEFERRED and DEFERRABLE by connecting to the replication target (trg\_user@ogg2) by using sqlplus and submitting the query: select table\_name, constraint\_name, deferrable, deferred from user\_constraints where constraint\_type = 'P';

```

GGSCI ([Host]) 5> exit
[OS prompt]$ cd ~/labs/Practice06/configs
[OS prompt]$ sqlplus src_user/<password>@ogg1
SQL*Plus: Release 12.1.0.2.0 Production on Mon May 4 18:39:23
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> truncate table account;
Table truncated.
SQL> @account.sql
1 row created.
... Many lines omitted for clarity ...
Commit complete.
SQL> truncate table account;
Table truncated.
SQL> @account.sql
1 row created.
... Many lines omitted for clarity ...
Commit complete.

```

- Select the OGG\_TRG window, and make sure the RTRNF Replicat group does not have an ABENDED status. If the TOGGLE\_CONSTRAINTS procedure had not been called, the RTRNF Replicat would ABEND because of the referential integrity errors. If, on the other hand, the primary key constraints are lifted, the Replicat group can successfully replicate duplicate rows. View the report on the RTRNF Replicat to verify that SQLEXEC was called as part of the replication process.

```

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > info all

Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
REPLICAT RUNNING RTRNF 00:00:00 00:01:56
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > view report RTRNF

```

In the report, you should see the lines:

```

SQLEXEC 'CALL trg_user.CONSTRAINTS_DISABLE ()'
Executing SQL statement...
2015-03-15 15:05:02 INFO OGG-00893 SQL statement executed
successfully.

```

This confirms that the stored procedure was executed.

7. Stop and delete all Extract and Replicat groups used during this practice (ETRNF, PTRNF, and RTRNF). Unregister the ETRNF Extract from the database. Alternatively, you can run the clean-up-prac-6\_4.sh file found in ~/labs/Practice06/configs to clean up the replication environment.
8. You must also completely clean up the environment, including Oracle RDBMS objects, Replicat and Extract groups, and trail fails stored in the dirdat directory in both source and target Oracle GoldenGate instances. The shortest way to accomplish this is to run the script lab-6-total-cleanup.sh located in the /home/oracle/labs/Practice06/configs directory:

```
[OS prompt]$ cd /home/oracle/labs/Practice06/configs
[OS prompt]$ pwd
/home/oracle/labs/Practice06/configs
[OS prompt]$./lab-6-total-cleanup.sh

Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
10:21:34
Operating system character set identified as UTF-8.
... Many lines omitted for brevity...
SQL> SQL> SQL> SQL> SQL> SQL> Disconnected from Oracle
Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
lab-6-total-cleanup script completed
[OS prompt]$
```

**Note:** the script can display error messages on the screen, which can be safely disregarded.

## **Practices for Lesson 7: Custom Behavior Through User Exits**

**Chapter 7**

## Practices for Lesson 7: Overview

---

### Practices Overview

In these practices, you will set up the Oracle GoldenGate User Exit mechanism to capture the data changed on the source database. The DML statements submitted to the RDBMS engine are captured and sent to a Java Message Service (JMS) queue, which is implemented using Apache ActiveMQ.

There are no catch-up or clean-up scripts for this practice. However, you will find the following files in the `/home/oracle/labs/Practice07/solutions` directory:

|                                |                                                             |
|--------------------------------|-------------------------------------------------------------|
| <code>euevt.prm</code>         | Primary Extract parameter file                              |
| <code>puevt.prm</code>         | Data pump parameter file                                    |
| <code>javadef.prm</code>       | Parameter file for the <code>defgen</code> utility          |
| <code>javaue.prm</code>        | Data delivery Extract parameter file                        |
| <code>javaue.properties</code> | Java properties file used to communicate with the JMS layer |
| <code>setenv.sh</code>         | Environment variable definitions                            |

**Note:** It is extremely important to correctly edit the `javaue.properties` file. An additional, unwanted space or an end-of-line character in the wrong position would prevent successful replication into the JMS broker. You are encouraged to copy the file from the `/home/oracle/labs/Practice07/solutions` directory into your practice environment (`/u03/ogg/oggjms_trg/dirprm`), rather than editing it.

## Practice 7-1: Verifying the Java Environment and Installing Apache ActiveMQ and Oracle GoldenGate for Java

### Overview

In this practice, you make sure that Java is correctly installed on the system and you install ActiveMQ, the JMS broker provided by the Apache Software Foundation. In addition, the software component Oracle GoldenGate Application Adapters for JMS and Flat File must be installed. The latter provides a user exit, a sharable image called `libggjava_ue.so`, which is a Java Native Interface (JNI) bridge that allows Oracle GoldenGate to interact with Java.

### Assumptions

Your system has the Java Development Kit (JDK) release 1.7 update 75 installed in the `/usr/java/jdk1.7.0_75` directory. You will refer to this directory by using the symbolic link "latest," so the JDK install directory becomes `/usr/java/latest`.

### Tasks

1. If your system has access to the Internet, point your browser to the URL <http://activemq.apache.org/activemq-5100-release.html> to download the Apache ActiveMQ software. Click the download link for Unix/Linux/Cygwin Distribution. Download the `apache-activemq-5.10.0-bin.tar.gz` file to the `/tmp` directory. Alternatively, if you do not have access to the Internet or if you want to save time, you can use the Apache ActiveMQ distribution kit stored in the `/install/activemq` directory.
2. Open a shell window, use the `su` command as the `root` user (you should know the `root` password by now; if unsure, ask your instructor), and change the directory to `/opt`. Expand (untar) the `apache-activemq-5.10.0-bin.tar.gz` file to create the `/opt/apache-activemq-5.10.0` subdirectory. Then use the `chown` command to change the ownership of the ActiveMQ directory and files from `root` to `oracle`:

```
[OS prompt]$ su -
Password: *****
[OS prompt]# cd /opt
[OS prompt]# tar xzvf /install/activemq/apache-activemq-5.10.0-bin.tar.gz
apache-activemq-5.10.0/bin/run.jar
apache-activemq-5.10.0/activemq-all-5.10.0.jar
apache-activemq-5.10.0/lib/activemq-camel-5.10.0.jar

... many lines omitted for clarity...

apache-activemq-5.10.0/webapps/fileserver/WEB-INF/classes/org/apache/activemq/util/FilenameGuardFilter.class
apache-activemq-5.10.0/webapps/fileserver/WEB-INF/classes/org/apache/activemq/util/IOHelper.class
apache-activemq-5.10.0/webapps/fileserver/WEB-INF/classes/org/apache/activemq/util/RestFilter.class

[OS prompt]# chown -R oracle.oinstall apache-activemq-5.10.0/
```

3. Log out from the root account. Make sure that your shell is connected as the default user (oracle) and make sure that Java is correctly installed by executing the command `java -version`.

```
[OS prompt]# exit
[OS prompt]$ id
uid=54321(oracle) gid=54321(oinstall) groups=54321(oinstall),
54322(dba)
[OS prompt]$ java -version
java version "1.7.0_75"
Java(TM) SE Runtime Environment (build 1.7.0_75-b13)
Java HotSpot(TM) 64-Bit Server VM (build 24.75-b04, mixed mode)
[OS prompt ~]$
```

4. You must install the Oracle GoldenGate Adapters release 11.2. You will find the distribution kit stored in the `/install/oggjava` directory. The Oracle GoldenGate Adapters software patch 11.2.1.0.4 is bundled in the zip file `p20447460_112104_Linux-x86-64.zip`. Change the directory to `/u03/ogg` and create the directory `oggjms_trg`. Change the directory to the newly created `oggjms_trg` directory and copy the zip file from `/install/oggjava` to the current directory:

```
[OS prompt]$ cd /u03/ogg
[OS prompt]$ mkdir oggjms_trg
[OS prompt]$ ls -l
drwxr-xr-x. 18 oracle oinstall 4096 Mar 25 21:35 oggjms_trg
drwxr-xr-x. 29 oracle oinstall 4096 Mar 22 00:38 ogg_node1
drwxr-xr-x. 29 oracle oinstall 4096 Mar 22 00:40 ogg_node2
drwxr-xr-x. 29 oracle oinstall 4096 Mar 22 00:41 ogg_node3
drwxr-xr-x. 29 oracle oinstall 4096 Mar 12 23:31 ogg_src
drwxr-xr-x. 29 oracle oinstall 4096 Mar 17 22:08 ogg_trg
[OS prompt]$ cd oggjms_trg
[OS prompt]$ pwd
/u03/ogg/oggjms_trg
[OS prompt]$ cp /install/oggjava/p20447460_112104_Linux-x86-64.zip .
```

5. Unzip the `p20447460_112104_Linux-x86-64.zip` file. Extract (untar) the Java Adapter TAR file:

```
[OS prompt]$ unzip p20447460_112104_Linux-x86-64.zip
Archive: p20447460_112104_Linux-x86-64.zip
 inflating: ggs_Adapters_Linux_x64.tar
 inflating: OGG_Adapters_Rel_Notes_11.2.1.0.4.pdf
 inflating: OGG_WinUnix_Rel_Notes_11.2.1.0.27.pdf
 inflating: Oracle-GoldenGate-11.2.1.0-README.txt
 inflating: Oracle_GoldenGate_Adapters-README_v11.2.1.txt
[OS prompt]$ tar xvf ggs_Adapters_Linux_x64.tar
AdapterExamples/
AdapterExamples/java-delivery/
AdapterExamples/java-delivery/format_op2xml.vm
```

```
... many lines omitted for clarity ...
```

```
./ucharset.h
./libantlr3c.so
./libicui18n.so.38
./tcperrs
[OS prompt]$
```

6. Launch the GGSCI utility and create the required subdirectories by using the create subdirs command:

```
[OS prompt]$ ggsci
Oracle GoldenGate Command Interpreter
Version 11.2.1.0.27 19591627
OGGCORE_11.2.1.0.0OGGBP_PLATFORMS_141006.1156
Linux, x64, 64bit (optimized), Generic on Oct 6 2014 13:09:44
Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
rights reserved.

GGSCI ([HOST]) 1> create subdirs
Creating subdirectories under current directory
/u03/ogg/oggjms_trg
Parameter files /u03/ogg/oggjms_trg/dirprm:
already exists
Report files /u03/ogg/oggjms_trg/dir rpt:
created
Checkpoint files /u03/ogg/oggjms_trg/dir chk:
created
Process status files /u03/ogg/oggjms_trg/dir p cs:
created
SQL script files /u03/ogg/oggjms_trg/dir sql:
created
Database definitions files /u03/ogg/oggjms_trg/dir def:
created
Extract data files /u03/ogg/oggjms_trg/dir dat:
created
Temporary files /u03/ogg/oggjms_trg/dir tmp:
created
Stdout files /u03/ogg/oggjms_trg/dir out:
created
GGSCI ([HOST]) 2>
```

7. Edit the manager parameter file and specify port 7709 where the manager will listen for incoming connections:

```
GGSCI ([HOST]) 2> edit param mgr
```

```
Port 7709
```

8. Exit the editor after saving the file and start the Oracle GoldenGate Manager process:

```
GGSCI ([HOST]) 3> start mgr
```

```
Manager started.
```

```
GGSCI ([HOST]) 4> info mgr detail
```

```
Manager is running (IP port host01.localdomain.7709).
```

9. Leave the shell window open and GGSCI running, connected to the Java Adapter Oracle GoldenGate instance.

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

## Practice 7-2: Configuring Data Capture

### Overview

In this practice, you reuse the primary Extract EUEVT and the Data Pump PUEVT, which were defined in the practices for Lesson 5, “Oracle GoldenGate Event Marker System.” To complete the capture configuration, you use the `defgen` utility to create column definitions that will be used by the delivery Extract on the replication target to map the database columns into Java data types.

### Tasks

1. Open a shell window and use the `oggsrc` alias to change the directory to `/u03/ogg/ogg_src`. Launch `GGSCI` and create (or edit, if it still exists in the `dirprm` directory) the `EUEVT` primary Extract parameter file, which captures all table activity in the `src_user` schema. Note that we want to change the format of the trail file to the (lower) Oracle GoldenGate version of the target environment (11.2). To force the Extract to write a trail file that follows specifications different from the default, the directive "`Format Release <major>.<minor>`" must be used when specifying the trail file in the Extract parameter file:

```
[OS prompt]$ oggsrc
[OS prompt]$ pwd
/u03/ogg/ogg_src
[OS prompt]$ ggsci
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
10:21:34
Operating system character set identified as UTF-8.
Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
rights reserved.
GGSCI ([SRCHOST]) > edit params euevt
```

```
Extract euevt
Exttrail ./dirdat/eu, format release 11.2
UseridAlias ogg_admin
LOGALLSUPCOLS
tranlogoptions integratedparams (max_sga_size 128)
SOURCECATALOG ogg1
Table src_user.*;
```

2. Exit from the editor after saving the parameter file. In GGSCI, log in to the database and register and add the newly created Extract:

```
GGSCI ([SRCHOST]) > DBLogin UserIdAlias ogg_admin
Successfully logged into database CDB$ROOT.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > Register
Extract EUEVT Database Container (ogg1)
Extract EUEVT successfully registered with database at SCN
3079294.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
euevt, Integrated Tranlog, begin now
EXTRACT added.
```

3. Connect the Extract to the ExtTrail, setting the exttrail file size to 10 MB:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add exttrail
./dirdat/eu, Extract euevt, megabytes 10
EXTTRAIL added.
```

4. Add the Data Pump Extract puevt, which physically transfers the captured data to the target host. Note that you specify remote port 7709, which is where the Java Adapter instance is listening for incoming Oracle GoldenGate connections. Also, note that you do not specify the catalog name (SOURCECATALOG ogg1) like you did in the primary Extract. If you do, the Data Pump Extract would abend because the eu trail fail has already been "downgraded" by EUEVT and it has lost all references to the ogg1 container:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
puevt
Extract puevt
SETENV (ORACLE_SID='CDB12c')
rmthost ogg_target, mgrport 7709
rmttrail ./dirdat/pu, format release 11.2
passthru
table src_user.*;
```

5. Connect the Data Pump to the local exttrail file:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
puevt, exttrailsource ./dirdat/eu
EXTRACT added.
```

6. Connect the remote trail file to the Data Pump, setting the remote trail file size to 10 MB:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add rmttrail
./dirdat/pu, Extract puevt, megabytes 10
RMTTRAIL added.
```

7. Start the Extract and Data Pump processes, check their status, and exit from GGSCI:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start extract
euevt
Sending START request to MANAGER ...
EXTRACT EUEVT starting
```

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start extract
puevt
Sending START request to MANAGER ...
EXTRACT PUEVT starting
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING EUEVT 00:00:00 00:41:39
EXTRACT RUNNING PUEVT 00:00:00 00:00:06
```

8. Create the `javadef` parameter file, which will be provided to the `defgen` utility. Note that for the `defgen` definitions also, you must force a format downgrade from 12c to 11g using the `FORMAT RELEASE` directive:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > Edit Param
javadef
```

```
-- defgen column definitions for tables in src_user.
UserIdAlias ogg_admin
defsfile ./dirdef/jmsdef.def format release 11.2
SOURCECATALOG ogg1
table src_user.economic_entity;
table src_user.gdp_by_year;
table src_user.gdp_growth_by_year;
```

9. Save the file and exit the editor. Exit `GGSCI` and, at the OS prompt, invoke the `defgen` utility:

```
([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > Exit
[OS prompt]$./defgen paramfile dirprm/javadef.prm

 Oracle GoldenGate Table Definition Generator for Oracle
 Version 12.1.2.1.0 OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1
 Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014 03:26:23
 Copyright (C) 1995, 2014, Oracle and/or its affiliates. All rights
 reserved.

 Starting at 2015-03-25 23:34:53

```

Operating System Version:  
Linux  
Version #2 SMP Wed Jan 28 17:03:28 PST 2015, Release 3.8.13-  
55.1.5.el6uek.x86\_64  
Node: host01.localdomain  
Machine: x86\_64

|            |            |
|------------|------------|
| soft limit | hard limit |
|------------|------------|

```

Address Space Size : unlimited unlimited
Heap Size : unlimited unlimited
File Size : unlimited unlimited
CPU Time : unlimited unlimited
Process id: 33571

** Running with the following parameters
**

UserIdAlias ogg_admin
defsfile ./dirdef/jmsdef.def
SOURCECATALOG ogg1
table src_user.economic_entity;
Default source catalog name ogg1 will be used for table specification
src_user.economic_entity.
Retrieving definition for SRC_USER.ECONOMIC_ENTITY.
table src_user.gdp_by_year;
Default source catalog name ogg1 will be used for table specification
src_user.gdp_by_year.
Retrieving definition for SRC_USER.GDP_BY_YEAR.
table src_user.gdp_growth_by_year;
Default source catalog name ogg1 will be used for table specification
src_user.gdp_growth_by_year.
Retrieving definition for SRC_USER.GDP_GROWTH_BY_YEAR.
Definitions generated for 3 tables in ./dirdef/jmsdef.def.

```

10. Copy the definition file to its target directory, the `dirprm` directory under `/u03/ogg/oggjms_trg`:

```

[OS prompt]$ pwd
/home/oracle/ggs_src
[OS prompt]$ cp dirdef/jmsdef.def /u03/ogg/oggjms_trg/dirdef
[OS prompt]$ ls -l /u03/ogg/oggjms_trg/dirdef
total 4
-rw-r-----. 1 oracle oinstall 1828 Mar 26 00:12 jmsdef.def
[OS prompt]$

```

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

## Practice 7-3: Configuring Data Delivery to ActiveMQ

### Overview

In this practice, you perform all necessary steps to configure the environment variables and the delivery Extract that will propagate all data changes to the JMS broker.

### Assumptions

Practice 7-1 and 7-2 have been successfully completed.

### Tasks

1. The **GGSCI** process that is running the Oracle GoldenGate Manager must have access to several Linux shared objects stored in subdirectories under the JDK. In addition, the manager process needs access to the **JAVA\_HOME** directory location. The environment variables **LD\_LIBRARY\_PATH** and **JAVA\_HOME** must be set *before* **GGSCI** is launched from the command line. Using the shell window you left open from Practice 7-2, change the directory to **/u03/ogg/oggjms\_trg** and create a **setenv.sh** file. Define the environment variables **LD\_LIBRARY\_PATH** and **JAVA\_HOME** in that file, which will be sourced before invoking **GGSCI**:

```
[OS prompt]$ cd /u03/ogg/oggjms_trg
[OS prompt]$ vi setenv.sh
LD_LIBRARY_PATH=/u01/app/oracle/product/12.1.0.2/db_1/lib:/usr/java/latest/jre/lib/amd64/server
export LD_LIBRARY_PATH
JAVA_HOME=/usr/java/latest
export JAVA_HOME
```

**Note:** The **LD\_LIBRARY\_PATH** line is one line, even if it is shown across multiple lines.

2. Save the **setenv.sh** file and exit the editor. Source the file using the dot command (.) and verify that the environment variables have been correctly set by echoing their values on the screen:

```
[OS prompt]$. setenv.sh
[OS prompt]$ echo $JAVA_HOME
/usr/java/latest
[OS prompt ggs_trg]$ echo $LD_LIBRARY_PATH
/u01/app/oracle/product/12.1.0.2/db_1/lib:/usr/java/latest/jre/lib/amd64:/u03/ogg/oggjms_trg:/usr/java/latest/jre/lib/amd64/server
```

3. You can now start GGSCI, which will inherit the environment variables that will allow the process to locate the shared objects required by the Oracle GoldenGate User Exit to work.

**IMPORTANT:** If the GGSCI Manager process is running, you must stop it and restart it so that it inherits the environment variables set by sourcing the setenv.sh file.

```
[OS prompt]$ ggsci
Oracle GoldenGate Command Interpreter
Version 11.2.1.0.27 19591627
OGGCORE_11.2.1.0.0OGGBP_PLATFORMS_141006.1156
Linux, x64, 64bit (optimized), Generic on Oct 6 2014 13:09:44
Copyright (C) 1995, 2014, Oracle and/or its affiliates. All rights reserved.

GGSCI ([TRGHOST]) > info all
Program Status Group Lag at Chkpt Time Since Chkpt
MANAGER RUNNING

GGSCI ([TRGHOST]) > stop mgr
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 ([TRGHOST]) > start mgr
Manager started.

GGSCI ([TRGHOST]) >
```

4. In the replication target environment, you must create the delivery Extract parameter file. Unlike database-to-database replication, which requires Extract processes on the source and Replicat processes on the target, in this case the delivery of data changes that occurred on the source is propagated to the JMS broker by using an Extract process, rather than a Replicat on the replication target.

```
GGSCI ([TRGHOST]) > Edit Param javaue
EXTRACT javaue
setEnv (GGS_USEREXIT_CONF = "./dirprm/javaue.properties")
GetEnv (JAVA_HOME)
GetEnv (PATH)
GetEnv (LD_LIBRARY_PATH)
GetEnv (LIBPATH)
CUSerExit libggjava_ue.so CUSEREXIT PASSTHRU INCLUDEUPDATEBEFRES
sourceDefs ./dirdef/jmsdef.def
getUpdateBefres
TABLE src_user.economic_entity;
TABLE src_user.gdp_by_year;
TABLE src_user.gdp_growth_by_year;
```

5. Save the parameter file and leave the GGSCI session open. Open a new shell window, change the directory to /u03/oggjms\_trg/dirprm, and configure the javaue.properties file.

```
[OS prompt ggs_trg]$ cd /u03/ogg/oggjms_trg/dirprm
[OS prompt dirprm]$ vi javaue.properties
javaue.properties (GoldenGate Java Extract properties file)
gg.handlerlist=advjms
java.naming.factory.initial=org.apache.activemq.jndi.ActiveMQInitialContextFactory
java.naming.provider.url=tcp://localhost:61616
gg.handler.advjms.type=jms
gg.handler.advjms.destination=dynamicQueues/oggjmsQueue
gg.handler.advjms.connectionFactory=ConnectionFactory
gg.handler.advjms.format=xml
native library config
goldengate.userexit.nochkpt=TRUE
goldengate.userexit.timestamp=utc
goldengate.log.logname=cuserexit
goldengate.log.level=INFO
goldengate.log.tofile=true
goldengate.userexit.writers=javawriter
#javawriter.stats.time=3600
#javawriter.stats.numrecs=10000
javawriter.stats.full=TRUE
javawriter.stats.display=TRUE
javawriter.bootoptions=-Xmx1024m -Xms512m -
Djava.class.path=ggjava/ggjava.jar:dirprm:/opt/apache-activemq-5.10.0/activemq-all-5.10.0.jar:/opt/apache-activemq-5.10.0/lib/optional/log4j-1.2.17.jar:/opt/apache-activemq-5.10.0/lib/optional/slf4j-log4j12-1.7.5.jar -
Dlog4j.configuration=log4j.properties
```

The lines specifying `java.naming.factory.initial` and `javawriter.bootoptions` are one line each even if, for formatting reasons, they are shown over multiple lines. It is *extremely important* to make sure that the lines are not broken up and no white spaces exist between keywords; otherwise, the delivery Extract process will not start. You can edit or display the `javaue.properties` file stored in `/home/oracle/labs/Practice07/solutions` to see what the properties file should look like.

Also, note the `gg.handler.advjms.format` parameter, which specifies that messages coming from the source database should be queued in XML format.

6. Leave the OS shell window open and resume the GGSCI session you left open, connected to the ggs\_trg Oracle GoldenGate instance. Add the data delivery Extract process, making sure that the Extract process was created and registered correctly:

```
GGSCI ([TRGHOST]) > ADD extract javaue, extTrailSource ./dirdat/pu
EXTRACT added.

GGSCI ([TRGHOST]) > Info All
Program Status Group Lag at Chkpt Time Since Chkpt

MANAGER RUNNING
EXTRACT STOPPED JAVAUE 00:00:00 00:00:33
```

7. Using the OS shell window you left open, change the directory to /opt/apache-activemq-5.10.0/bin and launch ActiveMQ in Console mode. Do not close the shell because ActiveMQ takes over the shell, occasionally printing log messages to the screen:

```
[OS prompt]$ cd /opt/apache-activemq-5.10.0/bin
[OS prompt bin]$./activemq console
INFO: Using default configuration
(you can configure options in one of these file: /etc/default/activemq
/home/user01/.activemqrc)

INFO: Invoke the following command to create a configuration file
./activemq setup [/etc/default/activemq | /home/oracle/.activemqrc]

INFO: Using java '/usr/bin/java'
INFO: Starting in foreground, this is just for debugging purposes
(stop process by pressing CTRL+C)
Java Runtime: Oracle Corporation 1.7.0_09 /usr/java/jdk1.7.0_75/jre
 Heap sizes: current=251264k free=247327k max=251264k
 JVM args: -Xms256M -Xmx256M -
Dorg.apache.activemq.UseDedicatedTaskRunner=true -
Djava.util.logging.config.file=logging.properties
 -Dcom.sun.management.jmxremote -Dactivemq.classpath=/opt/apache-
activemq-5.10.0/conf; -Dactivemq.home=/opt/apache-activemq-5.10.0
 -Dactivemq.base=/opt/apache-activemq-5.10.0
ACTIVEMQ_HOME: /opt/apache-activemq-5.10.0
ACTIVEMQ_BASE: /opt/apache-activemq-5.10.0
Loading message broker from: xbean:activemq.xml
 INFO | Refreshing
org.apache.activemq.xbean.XBeanBrokerFactory$1@bbf4061: startup date
[sat Oct 20 08:58:44 EST 2012]; root of context hierarchy
 WARN | destroyApplicationContextOnStop parameter is deprecated,
please use shutdown hooks instead
 INFO | PListStore:/opt/apache-activemq-
5.10.0/data/localhost/tmp_storage started
 INFO | Using Persistence Adapter:
KahaDBPersistenceAdapter[/opt/apache-activemq-5.10.0/data/kahadb]
 INFO | ActiveMQ 5.10.0 JMS Message Broker (localhost) is starting
 INFO | For help or more information please see:
http://activemq.apache.org/
```

```
INFO | Listening for connections at: tcp://EDDDR53P1:61616
INFO | Connector openwire Started
INFO | ActiveMQ JMS Message Broker (localhost, ID:host01.example.com-33804-1350683930829-0:1) started
INFO | jetty-7.1.6.v20100715
INFO | ActiveMQ WebConsole initialized.
INFO | Initializing Spring FrameworkServlet 'dispatcher'
INFO | ActiveMQ Console at http://0.0.0.0:8161/admin
INFO | ActiveMQ Web Demos at http://0.0.0.0:8161/demo
INFO | RESTful file access application at
http://0.0.0.0:8161/fileserver
INFO | Started SelectChannelConnector@0.0.0.0:8161
```

8. Resume the GGSCI session that is connected to the ggs\_trg Oracle GoldenGate instance and start the javaue data delivery Extract process.

```
GGSCI ([Host]) > start ext javaue

Sending START request to MANAGER ...
EXTRACT JAVAUE starting

GGSCI ([Host]) > info all

Program Status Group Lag at Chkpt Time Since Chkpt
MANAGER RUNNING
EXTRACT RUNNING JAVAUE 00:00:00 00:00:04
```

9. The first time JAVAUE is started, it takes several seconds before the status changes to RUNNING. A Java virtual machine is instantiated and a connection is made to Apache ActiveMQ, creating the dynamic queue oggjmsQueue. If, after 30 seconds, the status of JAVAUE is either STOPPED or ABENDED, you should investigate what happened by using the VIEW REPORT JAVAUE command, which lists the causes of the failure. An environment variable pointing to a wrong disk location, or not having stopped and restarted the manager process after sourcing the setenv.sh file are usually the reasons why the delivery Extract fails to start.

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

## Practice 7-4: Generating Data and Verifying Replication

### Overview

In this practice, you experience the results of your efforts so far. You generate database activity on the source database and you verify that the SQL statements submitted to the RDBMS engine are captured by the JMS queue and displayed in the browser connected to the administrative pages of Apache ActiveMQ.

### Assumptions

- Practice 7-1 to 7-3 have been successfully completed.
- The Extract processes EUEVT and PUEVT are running on the ggs\_src Oracle GoldenGate instance and the delivery Extract JAVAUE is running on the ggs\_trg Oracle GoldenGate instance.

### Tasks

1. Open a shell window and change the directory to ~/labs/Practice07/configs. Connect to the replication target account (trg\_user) in Oracle RDBMS by using SQLPLUS. Truncate the GDP\_BY\_YEAR table. Without leaving SQLPLUS, connect to the replication source account (src\_user) and execute the GDP\_BY\_YEAR\_2008.SQL script to populate the GDP\_BY\_YEAR table with economic data for the year 2008:

```
[OS prompt ~]$ cd ~/labs/Practice07/configs
[OS prompt configs]$ sqlplus trg_user/<password>@ogg2
SQL*Plus: Release 12.1.0.2.0 Production on Mon May 4 18:39:23 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options

SQL> truncate table gdp_by_year;
Table truncated.

SQL> connect src_user/<password>@ogg1
Connected.
SQL> truncate table gdp_by_year;
Table truncated.

SQL> @gdp_by_year_2008.sql

1 row created.

1 row created.

... many lines omitted for clarity ...
```

```
1 row created.
```

```
Commit complete.
```

```
SQL>
```

2. Launch your web browser and connect to localhost, port 8161, which is the URL of the administrative application that is used to manage Apache ActiveMQ. The credentials to connect to the ActiveMQ administrative application are: username: admin password: admin. Click the Manage ActiveMQ broker link and, on the next page, click the Queues tab at the top of the window, just below the ActiveMQ logo. The oggjmsQueue queue is displayed, showing 235 messages currently in the queue:

The screenshot shows a Google Chrome browser window with the title 'localhost : Queues - Google Chrome'. The address bar displays 'localhost : Queues' and the URL 'localhost:8161/admin/queues.jsp'. The page header features the ActiveMQ logo and the Apache Software Foundation logo with the URL 'http://www.apache.org/'. Below the header, there is a navigation menu with links: Home, Queues, Topics, Subscribers, Connections, Network, Scheduled, and Send. On the right side, there is a 'Support' link. A search bar labeled 'Queue Name' with a 'Create' button is present. The main content area is titled 'Queues' and contains a table with one row. The table columns are: Name, Number Of Pending Messages, Number Of Consumers, Messages Enqueued, Messages Dequeued, Views, and Operations. The single row shows 'oggjmsQueue' in the 'Name' column, '235' in 'Number Of Pending Messages', '0' in 'Number Of Consumers', '235' in 'Messages Enqueued', '0' in 'Messages Dequeued', and 'Views' and 'Operations' buttons. To the right of the table, there are links for 'Browse Active Consumers', 'Send To', 'Purge', and 'Delete'. Below the table, there is a 'Queue Views' section with 'Graph' and 'XML' options, a 'Topic Views' section with 'XML' option, and a 'Useful Links' section with links to Documentation, FAQ, Downloads, and Forums. At the bottom of the page, there is a copyright notice: 'Copyright 2005-2011 The Apache Software Foundation. ([printable version](#))'.

3. The queue name `oggjmsQueue` is displayed as a link. Click the link to display a list of all messages pending on the `oggjmsQueue` queue. Click any message link to display its content:

The screenshot shows a Mozilla Firefox browser window displaying an ActiveMQ message details page. The URL in the address bar is `localhost:8161/admin/message.jsp?id=ID%3ahost01.localdomain-43625-1427362103529-1:1:1:10`. The page title is "localhost : Message ID:host01.localdomain-43625-1427362103529-1:1:1:10 - Mozilla Firefox". The main content area displays a message with the following details:

| Headers        |                                                    | Properties                 |                            |
|----------------|----------------------------------------------------|----------------------------|----------------------------|
| Message ID     | ID:host01.localdomain-43625-1427362103529-1:1:1:10 | GG_XID                     | 00000000020000064940       |
| Destination    | queue://oggjmsQueue                                | GG_JMS_TIMESTAMP           | 1427362256180              |
| Correlation ID |                                                    | GG_TX_TIMESTAMP_STR        | 2015-03-26 09:30:52.001242 |
| Group          |                                                    | GG_SCHEMA                  | SRC_USER                   |
| Sequence       | 0                                                  | GG_JMS_MSG_CREATE_DURATION | 0                          |
| Expiration     | 0                                                  | GG_TX_READ_TIMESTAMP       | 1427362256125              |
| Persistence    | Persistent                                         | GG_TXIND                   | 1                          |
| Priority       | 4                                                  | GG_TABLE                   | GDP_BY_YEAR                |
| Redelivered    | false                                              | GG_ID                      | 00000000020000066194       |
| Reply To       |                                                    | GG_OBJNAME                 | SRC_USER.GDP_BY_YEAR       |
| Timestamp      | 2015-03-26 20:30:56:180 AEDT                       | GG_OPTYPE                  | INSERT                     |
| Type           | SRC_USER.GDP_BY_YEAR                               |                            |                            |

**Message Actions:**

- Delete
- Copy
- Move

**Message Details:**

```
<operation table='SRC_USER.GDP_BY_YEAR' type='INSERT' ts='2015-03-26 09:30:52.001242' pos='00000000020000066194' numCols='3'>
<col name='ENTITY_ID' index='0'>
<after><![CDATA[139]]></after>
</col>
<col name='GDP_YEAR' index='1'>
<after><![CDATA[2008]]></after>
</col>
<col name='GDP_VALUE' index='2'>
<after><![CDATA[31.45098]]></after>
</col>
</operation>
```

4. According to the directives specified in the `javeue.properties` file (Practice 6-3, task 6, directive: `gg.handler.advjms.format=xml`), the SQL statement entered on the replication source has been transformed into an XML operation. The various XML tags specify the column names and their values.
5. After you finish browsing through a few XML messages, close the web browser, select the shell window where ActiveMQ is running, and press **Ctrl + C** to stop the broker. Terminate the window. Select the `GGSCI` session where the `JAVAUE Extract` group is running. Stop and delete the Extract group `JAVAUE`.

6. Stop and delete the EUEVT and PUEVT Extract groups. Open a new terminal shell. At the OS prompt, use the `oggsrc` alias to navigate to the `/u03/ogg/ogg_src` directory and launch GGSCI. Use the DBLOGIN command to connect to the database instance, and then stop and delete the Extract groups. Finally, unregister the EUEVT Extract from the database:

```
[OS prompt]$ oggsrc
[OS prompt]$ pwd
/u03/ogg/ogg_src
[OS prompt]$ ggsci
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
10:21:34
Operating system character set identified as UTF-8.
Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
rights reserved.
GGSCI ([HOST]) 1> DBLogin UserIdAlias ogg_admin
Successfully logged into database.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop er *
Sending STOP request to EXTRACT EUEVT...
Request processed.
Sending STOP request to EXTRACT PUEVT...
Request processed.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > delete er * !
Deleted EXTRACT EUEVT.
Deleted EXTRACT PUEVT.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > Unregister Extract EUEVT Database
Successfully unregistered EXTRACT EUEVT from database.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > exit
```

This completes Practice 7-4.

This completes Practice 7. Stop here.



## **Practices for Lesson 8: Configuring Zero-Downtime Migration Replication**

**Chapter 8**

## Practices for Lesson 8: Overview

---

### Practices Overview

In these practices, you will configure a zero-downtime migration and simulate an upgrade. The upgrade will fail, and you will revert to the point in time prior to the attempted migration, making sure that no data is lost during the interim period.

**Helper scripts provided in /home/oracle/labs/Practice08/configs:**

Catch-up/Clean-up script	Purpose
catch-up-prac-8_1.sh	Executes steps for Practice 8-1
catch-up-prac-8_2.sh	Executes steps for Practice 8-2
catch-up-prac-8_3.sh	Executes steps for Practice 8-3
catch-up-prac-8_4.sh	Executes steps for Practice 8-4
clean-up.sh	Complete clean-up script. It resets the environment as it was before starting Practice 8.

**Note:** The clean-up script completely resets your environment. Suppose you complete Practice 8-2, and while completing Practice 8-3, you decide to reset your environment by invoking the clean-up.sh script. Then, you will have to run catch-up-prac-8\_1.sh and catch-up-prac-8\_2.sh before you can start the steps required for Practice 8-3.

## Practice 8-1: Configuring a Zero-Down Time Migration

### Overview

In this practice, you configure the initial replication from the source database to the target database, preparing your environment for migration cutover.

### Assumptions

The CDB12C and DWNSTR databases are available.

### Tasks

1. Configure your desktop in the usual manner, with four terminal shell windows aligned to form a two-by-two matrix. Name the top left window "OGG\_SRC." At the OS prompt, type "oggsrc" to position the default directory to /u03/ogg/ogg\_src and launch GGSCI. Name the top right window "OGG\_TRG" and at the OS prompt, use the alias oggtrg to navigate to the /u03/ogg/ogg\_trg directory and launch GGSCI. Name the bottom left window "CDB12C" and at the OS prompt, navigate to the ~/labs/Practice08/configs directory. Name the bottom right window "DWNSTR" and at the OS prompt, navigate to the ~/labs/Practice08/configs directory.
2. Select the CDB12C window. Set the ORACLE\_SID environment variable to CDB12C and launch sqlplus, connecting to CDB12C as sysdba:

```
[OS prompt] $ pwd
/home/oracle/labs/Practice08/configs
[OS prompt] $ export ORACLE_SID=CDB12C
[OS prompt] $ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Mon Mar 16 22:56:19
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL>
```

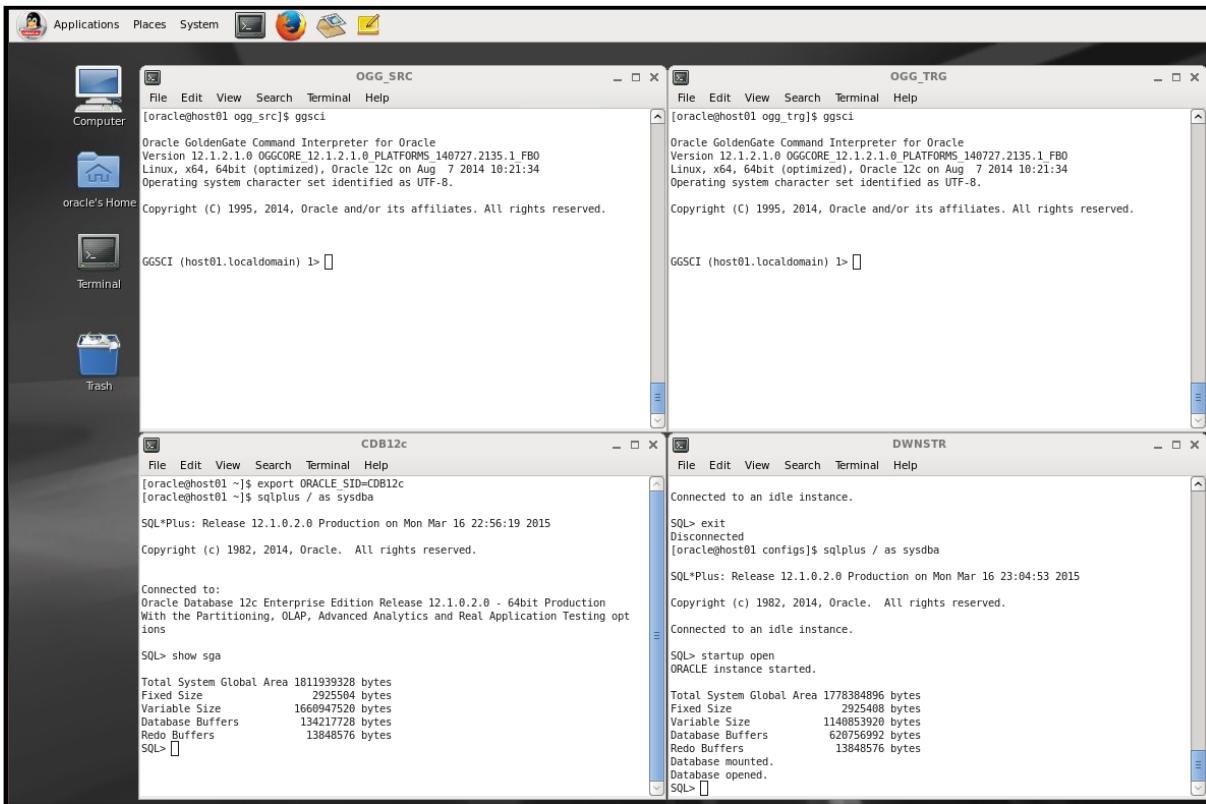
3. Select the DWNSTR window. Set the ORACLE\_SID environment variable to DWNSTR and launch sqlplus, connecting to DWNSTR as sysdba. Start the database:

```
[OS prompt] $ pwd
/home/oracle/labs/Practice08/configs
[OS prompt] $ export ORACLE_SID=DWNSTR
[OS prompt] $ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Mon Mar 16 22:56:19
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

Connected to an idle instance.
SQL> startup open
ORACLE instance started.

Total System Global Area 1778384896 bytes
Fixed Size 2925408 bytes
Variable Size 1140853920 bytes
Database Buffers 620756992 bytes
Redo Buffers 13848576 bytes
Database mounted.
Database opened.
SQL>
```

At this point, your desktop environment should look like the screenshot below:



- Select the CDB12c window, where sqlplus is connected to the CDB12c database as sysdba. Create the simulapp user, giving it the minimum privilege set to connect to the RDBMS and create objects. Make sure you position yourself in the OGG1 PDB by using the "alter session set container" command. Grant the simulapp user read/write privileges on the DATA\_PUMP\_DIR, which will be used for the database dump. Connect to CDB12c by using the simulapp user and create the objects listed in the oracle\_source\_table\_creation.sql file. Then run the scripts to populate the tables:

```

SQL> alter session set container = ogg1;
Session altered.

SQL> create user simulapp identified by simulapp default
tablespace oggdata temporary tablespace temp;
User created.

SQL> grant connect,resource, unlimited tablespace to simulapp;
Grant succeeded.

SQL> create or replace directory OGG1_DATA_PUMP_DIR as
'/home/oracle/dump';
Directory created.

SQL> grant read,write on directory OGG1_DATA_PUMP_DIR to
simulapp;
Grant succeeded.

SQL> grant DATAPUMP_EXP_FULL_DATABASE to simulapp;
Grant succeeded.

SQL> connect simulapp/simulapp@ogg1

```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

```

Connected.

SQL> @oracle_source_table_creation.sql
Table created.
Table created.
Table created.

SQL> @economic_entity.sql
1 row created.
...Many lines omitted for clarity...
SQL> @gdp_by_year.sql
1 row created.
...Many lines omitted for clarity ...
SQL> @gdp_growth_by_year.sql
1 row created.
...Many lines omitted for clarity ...
SQL>

```

- Select the OGG\_SRC window, where GGSCI is running in the replication source environment. Connect to the CDB12c database by using the ogg\_admin alias. Then add the schema trandata information for all objects stored in the SIMULAPP schema:

```

GGSCI ([Host]) > DBLogin UserIdAlias ogg_admin
Successfully logged into database CDB$ROOT.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add
SchemaTranData ogg1.simulapp ALLCOLS
2015-03-17 00:02:28 INFO OGG-01788 SCHEMATRANDATA has been
added on schema simulapp.
2015-03-17 00:02:29 INFO OGG-01976 SCHEMATRANDATA for
scheduling columns has been added on schema simulapp.
2015-03-17 00:02:29 INFO OGG-01977 SCHEMATRANDATA for all
columns has been added on schema simulapp.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) >

```

- Create the Extract EZDTA, specifying the Oracle home directory and the Oracle SID:

```

GGSCI ([Host]) 6> edit params ezdta
extract ezdta
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "CDB12c")
Exttrail ./dirdat/du
UserIdAlias ogg_admin
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
SOURCECATALOG ogg1
TABLE simulapp.*;

```

Save the file and return to GGSCI.

7. Register the EZDTA Extract with the OGG1 PDB. Add the Extract and the ExtTrail, and then edit the parameters for the data pump Extract PZDTA:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > REGISTER
EXTRACT EZDTA DATABASE CONTAINER (ogg1)

Extract EZDTA successfully registered with database at SCN
2996338.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
ezdta, Integrated tranlog, begin now

EXTRACT added.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add exttrail
./dirdat/du, Extract ezdta, megabytes 10

EXTTRAIL added.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
pzdt
```

8. Create the Data Pump Extract as PASSTHRU to improve performance, given that the database is the same (Oracle) and the tables have the same structure on both source and target. Add the Data Pump Extract, connecting it to the ExtTrail produced by EZDTA (./dirdat/du). Then add the remote trail file (./dirdat/qu), specifying its size as 10 MB:

```
extract pzdt
SETENV (ORACLE_SID='CDB12c')
UserIdAlias ogg_admin
RMTHOST ogg_target, MGRPORT 7909
rmttrail ./dirdat/qu
passthru
SOURCECATALOG ogg1
table simulapp.*;
```

Save the file and return to GGSCI.

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
pzdt, exttrailsource ./dirdat/du

EXTRACT added.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add rmttrail
./dirdat/qu, extract pzdt, megabytes 10

RMTTRAIL added.
```

9. Start the primary Extract and the Data Pump, verifying their status using `INFO ALL`:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start er *
Sending START request to MANAGER ...
EXTRACT EZDTA starting

Sending START request to MANAGER ...
EXTRACT PZDTA starting
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING EZDTA 00:00:00 00:00:04
EXTRACT RUNNING PZDTA 00:00:00 00:20:21
```

10. After the primary Extract and the Data Pump are running, you can back up the source database and transfer it over to the target system. Any activity occurring at the application level and involving the database is captured and stored in the trail files, so no data is lost while transferring the database. You should note that no down time is required while cloning the source database into the target system. The Replicat process on the target will be run with the `HANDLECOLLISIONS` option, which will take care of fixing the inconsistencies that could occur while importing the data on the target, given that data can still be changed on the source database while the target database is being loaded. To export the source database and to copy the dump file to the target system, follow the steps below. Open a new shell window; first, you export the source schema by using an Oracle Data Pump export (`expdp`). The `simulapp` user was granted read and write access to the `OGG1_DATA_PUMP_DIR` directory (`/home/oracle/dump`), which is now used to store the schema dump. First, create the directory, and then invoke `expdp` to perform the schema dump. Leave the shell open:

```
[OS prompt]$ cd ~
[OS prompt]$ mkdir dump
[OS prompt]$ expdp simulapp/simulapp@ogg1 schemas=simulapp
directory=ogg1_data_pump_dir dumpfile=ogg1.dmp logfile=ogg1.log

Connected to: Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
FLASHBACK automatically enabled to preserve database integrity.
Starting "SIMULAPP"."SYS_EXPORT_SCHEMA_01":
simulapp/********@ogg1 schemas=simulapp
directory=ogg1_data_pump_dir dumpfile=ogg1.dmp logfile=ogg1.log
Estimate in progress using BLOCKS method...
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
Total estimation using BLOCKS method: 1.5 MB
Processing object type SCHEMA_EXPORT/USER
```

```

Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
Processing object type SCHEMA_EXPORT/ROLE_GRANT
Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
Processing object type SCHEMA_EXPORT/TABLE/PROCACT_INSTANCE
Processing object type SCHEMA_EXPORT/TABLE(TABLE)
Processing object type SCHEMA_EXPORT/TABLE/COMMENT
Processing object type SCHEMA_EXPORT/TABLE/INDEX(INDEX)
Processing object type SCHEMA_EXPORT/TABLE/CONSTRAINT(CONSTRAINT)
Processing object type
SCHEMA_EXPORT/TABLE/INDEX/STATISTICS(INDEX_STATISTICS)
Processing object type
SCHEMA_EXPORT/TABLE/STATISTICS(TABLE_STATISTICS)
Processing object type SCHEMA_EXPORT/STATISTICS(MARKER)
exported "SIMULAPP"."ECONOMIC_ENTITY" 11.88 KB
235 rows
exported "SIMULAPP"."GDP_BY_YEAR" 18.23 KB
705 rows
exported "SIMULAPP"."GDP_GROWTH_BY_YEAR" 17.57 KB
705 rows
Master table "SIMULAPP"."SYS_EXPORT_SCHEMA_01" successfully
loaded/unloaded

Dump file set for SIMULAPP.SYS_EXPORT_SCHEMA_01 is:
 /home/oracle/dump/ogg1.dmp
Job "SIMULAPP"."SYS_EXPORT_SCHEMA_01" successfully completed at
Tue Mar 17 00:50:48 2015 elapsed 0 00:00:43

```

11. Select the DWNSTR window. Set the container to be oggdwn1 and create the oggdata tablespace for the oggdwn1 PDB; create the simulapp user on the OGGDWN1 PDB database and grant the connect, resource and unlimited tablespace privileges to simulapp. Also, you must create the OGGDWN1\_DATA\_PUMP\_DIR directory and grant the read and write privileges to that directory so that simulapp can import the dump file:

```

SQL> alter session set container = oggdwn1;
Session altered.

SQL> create tablespace oggdata datafile
 '/u02/oradata/DWNSTR/oggdwn1/oggdata01.dbf' size 50M extent
 management local uniform size 512K;
Tablespace created.

SQL> create user simulapp identified by simulapp default
 tablespace oggdata temporary tablespace temp;
User created.

SQL> grant connect,resource, unlimited tablespace to simulapp;
Grant succeeded.

SQL> create or replace directory OGGDWN1_DATA_PUMP_DIR as
 '/home/oracle/dump';

```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

```
Directory created.
SQL> grant read, write on directory OGGDWN1_DATA_PUMP_DIR to
simulapp;
Grant succeeded.
SQL> grant DATAPUMP_IMP_FULL_DATABASE to simulapp;
Grant succeeded.
SQL>
```

12. You can now import the dump file into the target database by using the impdp utility. Select the shell window you previously used for the Data Pump export and launch impdp. You can safely disregard the error message ORA-31684 if it occurs in your environment. When the import terminates, you can exit from the shell, terminating the window:

```
[OS prompt]$ impdp simulapp/simulapp@oggdwn1 schemas=simulapp
directory=oggdwn1_data_pump_dir dumpfile=ogg1.dmp
logfile=ogg1_imp.log
Import: Release 12.1.0.2.0 - Production on Tue Mar 17 07:25:29
2015
Copyright (c) 1982, 2014, Oracle and/or its affiliates. All
rights reserved.
Connected to: Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
Master table "SIMULAPP"."SYS_IMPORT_SCHEMA_01" successfully
loaded/unloaded
Starting "SIMULAPP"."SYS_IMPORT_SCHEMA_01":
simulapp/********@oggdwn1 schemas=simulapp
directory=oggdwn1_data_pump_dir dumpfile=ogg1.dmp
logfile=ogg1_imp.log
Processing object type SCHEMA_EXPORT/USER
ORA-31684: Object type USER:"SIMULAPP" already exists
Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
Processing object type SCHEMA_EXPORT/ROLE_GRANT
Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
Processing object type SCHEMA_EXPORT/TABLE/PROCACT_INSTANCE
Processing object type SCHEMA_EXPORT/TABLE/TABLE
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
. . imported "SIMULAPP"."ECONOMIC_ENTITY" 11.88
KB 235 rows
. . imported "SIMULAPP"."GDP_BY_YEAR" 18.23
KB 705 rows
. . imported "SIMULAPP"."GDP_GROWTH_BY_YEAR" 17.57
KB 705 rows
Processing object type SCHEMA_EXPORT/TABLE/INDEX/INDEX
Processing object type SCHEMA_EXPORT/TABLE/CONSTRAINT/CONSTRAINT
```

```

Processing object type
SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type
SCHEMA_EXPORT/TABLE/STATISTICS(TABLE_STATISTICS)
Processing object type SCHEMA_EXPORT/STATISTICS/MARKER
Job "SIMULAPP"."SYS_IMPORT_SCHEMA_01" completed with 1 error(s)
at Tue Mar 17 07:26:04 2015 elapsed 0 00:00:27 [OS prompt]$ exit

```

13. Select the CDB12C window and connect to the ogg1 PDB as simulapp. Create some activity on the source database. This step simulates the application updating the source database while it is being cloned on the target system. The data added to the source database is being captured by the Extract process and stored in the trail files. You will execute the gdp\_by\_year\_2008.sql script, which adds GDP data for all economic entities for the year 2008:

```

SQL> connect simulapp/simulapp@ogg1
Connected.
SQL> @gdp_by_year_2008.sql
1 row created.
...Many lines omitted for brevity...
SQL>

```

14. You now create the Replicat process on the target system. Select the OGG\_TRG window and add the credentials to connect to the C##OGG\_DWNADM user on the oggdwn1 PDB. You created the user C##OGG\_DWNADM in Practice 3-3, step1 and gave it dba, cdb\_dba, and pdb\_dba privileges across all containers ("container=all"). Create an alias for it so that the replicat parameter files do not have to store the password in clear text. If you do not remember the password, ask your instructor. Use DBLogin to log in to the database via the newly created alias. Edit the parameters for the RZDTA Replicat group:

```

GGSCI ([Host]) > Alter CredentialStore Add User
C##OGG_DWNADM@oggdwn1 Password <password> Alias ogg_dwnadm1
Credential store in ./dircrd/ altered.
GGSCI ([Host]) > DBLogin UserIdAlias ogg_dwnadm1
Successfully logged into database OGGDWN1.

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/OGGDWN1) > edit params
rzdata

```

15. The crucial parameter to be added to the Replicat group is HANDLECOLLISIONS, which forces Replicat to deal with possible discrepancies that occurred while performing the initial load. Because the source and the target are identical in structure, you can specify ASSUMETARGETDEFS:

```
replicat rzdata
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "DWNSTR")
UserIdAlias ogg_dwnadml
discardfile ./dirrpt/RZDTA.dsc, Append
handlecollisions
assumetargetdefs
MAPEXCLUDE oggl.simulapp.SYS_EXPORT_SCHEMA_01;
MAP oggl.simulapp.* , TARGET oggdwn1.simulapp.*;
```

16. Save the parameters and exit the editor. When you are back in GGSCI, add the RZDTA Integrated Replicat and start the Replicat group:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/OGGDWN1) > add replicat
rzdata, Integrated extrail ./dirdat/qu
REPLICAT (Integrated) added.
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/OGGDWN1) > start replicat
rzdata
Sending START request to MANAGER ...
REPLICAT RZDTA starting
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/OGGDWN1) > info all

Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
REPLICAT RUNNING RZDTA 00:00:00 00:00:53
```

17. Select the DWNSTR window, where sqlplus is connected to the DWNSTR database as sysdba. Connect as the target simulapp user on the OGGDWN1PDB. Make sure that all rows have been replicated, including the GDP values for 2008, which were added after you took the database dump:

```
SQL> connect simulapp/simulapp@oggdwn1
Connected.
SQL> select count(*) from gdp_by_year;
COUNT(*)

940

SQL> select count(*) from gdp_by_year where gdp_year=2008;
COUNT(*)

235
```

18. Select the OGG\_TRG window, this time to make sure that the Replicat has finished processing the rows added after the initial load. Use the LAG command to make sure that the RZDTA Replicat process has completed:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/OGGDWN1) > lag replicat
rzdata

Sending GETLAG request to REPLICAT RZDTA ...
Last record lag 10 seconds.
Low watermark lag: 0.
High watermark lag: 12.
Low watermark position: Not Available.
High watermark position: 3119521.
At EOF, no more records to process.
```

19. You might have to submit the LAG command repeatedly until it displays the message, "At EOF, no more records to process." You can now stop the RZDTA Replicat and erase the HANDLECOLLISIONS parameter, because from now onwards, you must resolve collisions without relying on the default resolution provided by Replicat. After you exclude the HANDLECOLLISIONS parameter, you must restart the RZDTA Replicat:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/OGGDWN1) > stop replicat
rzdata

Sending STOP request to REPLICAT RZDTA ...
Request processed.

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/OGGDWN1) > edit params
rzdata
```

20. Remove the HANDLECOLLISIONS line and save the file. Start the Replicat group again:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/OGGDWN1) >start rep rzdata
Sending START request to MANAGER ...
REPLICAT RZDTA starting

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/OGGDWN1) > info all

Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
REPLICAT RUNNING RZDTA 00:00:00 00:00:04
```

21. The environment is completely set up. You are replicating the source database into the target database, all rows have been correctly transferred, and your environment is ready for cutover. In addition, you built the infrastructure required to implement zero-downtime migration without requiring any application down time. However, before the migration cutover can occur, you must configure the fallback Extract, pump, and Replicat. You will do that in Practice 8-2.

This completes Practice 8-1. Continue with Practice 8-2.

## Practice 8-2: Configuring the Fallback

### Overview

In this practice, you configure (but not start) the fallback Extract, pump, and Replicat groups that will be active at the moment of migration cutover.

### Assumptions

Practice 8-1 has been successfully completed.

### Tasks

1. Select the OGG\_TRG window, use DBLogin to connect to the DWNSTR database as the C##OGG\_DWNADM user, and edit the parameter file for the EFBM Extract:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/OGGDWN1) DBLogin
UserIdAlias ogg_dwnadm
```

Successfully logged into database CDB\$ROOT.

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > edit params
EFBM
```

```
extract efbm
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "DWNSTR")
UserIdAlias ogg_dwnadm
discardfile ./dirrpt/EFBM.dsc, Append
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
exttrail ./dirdat/fb
SOURCECATALOG oggdwn1
TABLE simulapp.*;
```

2. Save the file and, when you are back in GGSCI, register the EFBM extract with the oggdwn1 database. Add the Extract and the ExtTrail:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > REGISTER
EXTRACT EFBM DATABASE CONTAINER (oggdwn1)
Extract EFBM successfully registered with database at SCN
2067731.
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > add extract
efbm, Integrated tranlog, begin now
EXTRACT added.
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > add exttrail
./dirdat/fb, extract efbm, megabytes 10
EXTTRAIL added.
```

3. Enter the `INFO ALL` command to verify that the Extract has been added. Then edit the parameters for the PFBM Data Pump Extract:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt
```

Program	Status	Group	Lag at Chkpt	Time Since
MANAGER	RUNNING			
EXTRACT	STOPPED	EFBM	00:00:00	00:25:30
REPLICAT	RUNNING	RZDTA	00:00:00	00:00:00

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > edit params
PFBM
```

```
extract pfbm
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "DWNSTR")
UseridAlias ogg_dwnadm
passthru
discardfile ./dirrpt/PFBM.dsc, append
rmthost ogg_source, mgrport 7809
rmttrail ./dirdat/bf
SOURCECATALOG oggdwn1
TABLE simulapp.*;
```

4. Save the parameter file and return to GGSCI.  
 5. Add supplemental logging for all tables in the `simulapp` schema:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > add
SchemaTranData oggdwn1.simulapp
2015-03-17 20:50:52 INFO OGG-01788 SCHEMATRANDATA has been
added on schema simulapp.
2015-03-17 20:50:52 INFO OGG-01976 SCHEMATRANDATA for
scheduling columns has been added on schema simulapp.
```

6. Add the Data Pump Extract and the RMTTRAIL, but do not start the group:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > add extract
pfbm, exttrailsource ./dirdat/fb
EXTRACT added.
```

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > add rmttrail
./dirdat/bf, extract pfbm, megabytes 10
RMTTRAIL added.
```

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt
```

Program	Status	Group	Lag at Chkpt	Time Since
MANAGER	RUNNING			
EXTRACT	STOPPED	EFBM	00:00:00	01:29:54

EXTRACT	STOPPED	PFBM	00:00:00	00:00:44
REPLICAT	RUNNING	RZDTA	00:00:00	00:00:00

7. Select the OGG\_SRC window and create a UserIdAlias for the user that will be used by Replicat to connect to the ogg1 PDB in CDB12c:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > Alter
CredentialStore Add User C##OGG_ADMIN@ogg1 password <password>
Alias ogg1_repl
Credential store in ./dircrd/ altered.
```

8. Edit the parameter file for the RFBM Replicat:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
RFBM
```

```
replicat rfbm
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "CDB12c")
DBOPTIONS INTEGRATEDPARAMS (parallelism 4)
UserIdAlias ogg1_repl
discardfile ./dirrpt/RFBM.dsc, append
assumetargetdefs
map oggdwn1.simulapp.* , target ogg1.simulapp.*;
```

The fallback Replicat re-synchronizes the old database with the new database after migration cutover.

9. Save the parameter file and, when you are back in GGSCI, add the Integrated Replicat group and verify its addition via INFO ALL:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add replicat
rfbm, integrated exttrail ./dirdat/bf
REPLICAT (Integrated) added.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING EZDTA 00:00:00 00:00:08
EXTRACT RUNNING PZDTA 00:00:00 00:00:07
REPLICAT STOPPED RFBM 00:00:00 00:01:04
```

This last step concludes the preparatory phase for migration cutover. The infrastructure is now in place to support a zero-down time migration.

This completes Practice 8-2. Continue with Practice 8-3.

## Practice 8-3: Performing Migration Cutover

### Overview

In this practice, you simulate a migration cutover. The application (in your case `sqlplus`) is stopped after loading one more batch of GDP statistics. After verifying that there is no replication lag, you stop the Extract and then the Data Pump. On the remote node, which is about to host the new database for the application, after you verify that there is no lag, you stop the Replicat. Before allowing the application to use the new database, you begin capturing changes on the new database by enabling the fallback Extract, which will synchronize the old database with all SQL activity performed by the application.

### Assumptions

Practices 8-1 and 8-2 have been completed and the fallback infrastructure is in place.

### Tasks

1. Select the `CDB12c` window, where `sqlplus` is running. Connect to the `simulapp` schema (user `simulapp`). Execute the `gdp_growth_by_year_2008.sql` script, which will populate the `GDP_GROWTH_BY_YEAR` table with data for 2008. You are simulating the last activity on the old database before migration cutover.

```
SQL> connect simulapp/simulapp@ogg1
SQL> @gdp_growth_by_year_2008.sql

1 row created.
...Many lines omitted for brevity...
SQL> exit

Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
```

2. Exiting `sqlplus` symbolizes stopping the application. In a real-life production environment, all application server and external application connections to the database would be stopped to prevent further activity on the database. The migration cutover has begun. Select the `OGG_SRC` window. Enter the `LAG` command before stopping the Data Pump Extract:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING EZDTA 00:00:09 00:00:07
EXTRACT RUNNING PZDTA 00:00:00 00:00:03
REPLICAT STOPPED RFBM 00:00:00 00:09:19

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop extract
ezdta
```

```
Sending STOP request to EXTRACT EZDTA ...
Request processed.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > lag extract
pzdta
Sending GETLAG request to EXTRACT PZDTA ...
Last record lag: 5 seconds.
At EOF, no more records to process.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop extract
pzdta

Sending STOP request to EXTRACT PZDTA ...
Request processed.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all

Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT STOPPED EZDTA 00:00:00 00:03:16
EXTRACT STOPPED PZDTA 00:00:00 00:02:15
REPLICAT STOPPED RFBM 00:00:00 01:06:21
```

3. Select the OGG\_TRG window. Enter a LAG command for the RZDTA Replicat group. If there is no lag, stop the Replicat:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT STOPPED EFBM 00:00:00 03:08:01
EXTRACT STOPPED PFBM 00:00:00 01:38:52
REPLICAT RUNNING RZDTA 00:00:00 00:00:07

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > lag replicat
rzdata

Sending GETLAG request to REPLICAT RZDTA ...
Last record lag: 7 seconds.
At EOF, no more records to process.
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > stop replicat
rzdata

Sending STOP request to REPLICAT RZDTA ...
Request processed.
```

4. The old database is not replicating information to the new database anymore. Before allowing the application to use the new database, you have to enable the Extract that synchronizes the new database with the old database.

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > alter extract
efbm, begin now
EXTRACT altered.
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > start extract
efbm

Sending START request to MANAGER ...
EXTRACT EFBM starting
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > start extract
pfbm

Sending START request to MANAGER ...
EXTRACT PFBM starting

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
```

```

EXTRACT RUNNING EFBM 00:00:00 00:00:05
EXTRACT RUNNING PFBM 00:00:00 20:45:21
REPLICAT STOPPED RZDTA 00:00:00 19:03:44

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) >

```

5. Select the OGG\_SRC window, where GGSCI is connected to the replication source. You will start the RFBM Replicat so that the new database is replicated to the old database:

```

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT STOPPED EZDTA 00:00:00 19:26:13
EXTRACT STOPPED PZDTA 00:00:00 19:25:12
REPLICAT STOPPED RFBM 00:00:00 20:29:18

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start replicat
RFBM
Sending START request to MANAGER ...
REPLICAT RFBM starting

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT STOPPED EZDTA 00:00:00 19:29:31
EXTRACT STOPPED PZDTA 00:00:00 19:28:30
REPLICAT RUNNING RFBM 00:00:00 00:00:10

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) >

```

6. The migration cutover has completed. You now allow the application to start using the new database. You simulate this by using sqlplus to connect to the “new” database, DWNSTR. In real life, the Oracle RDBMS is accessed through middleware, for example, application servers. In this case, the JDBC connection strings are modified within the application server to force the client application to connect to the new database. Select the DWNSTR window and connect to the simulapp schema that is hosted on the oggdwn1 PDB. Execute the gdp\_by\_year\_2009.sql and gdp\_growth\_by\_year\_2009.sql scripts to populate the GDP\_BY\_YEAR and GDP\_GROWTH\_BY\_YEAR tables. Verify that the rows are correctly replicated to the old database, OGG1:

```

SQL> connect simulapp/simulapp@oggdwn1
Connected.
SQL> @gdp_by_year_2009.sql

```

```
1 row created.
...Many lines omitted for brevity...
SQL> @gdp_growth_by_year_2009.sql
1 row created.
...Many lines omitted for brevity...
SQL>
```

7. Connect to the “old” database and verify that GDP and GDP growth data for the year 2009 have been replicated. Select the CDB12c window and launch sqlplus, connecting to the simulapp schema in the ogg1 PDB database:

```
[OS prompt] $ sqlplus simulapp/simulapp@ogg1
SQL*Plus: Release 12.1.0.2.0 Production on Tue Mar 17 21:36:13
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> select max(gdp_year) from gdp_by_year;

MAX(GDP_YEAR)

2009

SQL> select max(gdp_year) from gdp_growth_by_year;

MAX(GDP_YEAR)

2009
SQL>
```

8. The simulated “application,” sqlplus, which is connected to the new database, has generated rows that have been successfully replicated to the old database.

This completes Practice 8-3. Continue with Practice 8-4.

## Practice 8-4: Aborting the Migration

---

### Overview

In this practice, you decide that the migration must be aborted, and you revert to the old database, which was kept synchronized after the migration.

### Assumptions

Practices 8-1 to 8-3 have been successfully completed.

### Tasks

1. Stop the application from accessing the new database (DWNSTR). In your simulated environment, this simply means you do not connect to DWNSTR by using sqlplus.
2. Select the OGG\_TRG window. Verify that no lag exists for the EFBM Extract, and then stop it. Do the same for the PFBM Data Pump Extract:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > stop extract
EFBM
Sending STOP request to EXTRACT EFBM ...
Request processed.

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > lag extract
PFBM
Sending GETLAG request to EXTRACT PFBM ...
Last record lag: 5 seconds.
At EOF, no more records to process.

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > stop extract
PFBM
Sending STOP request to EXTRACT PFBM ...
Request processed.

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT STOPPED EFBM 00:00:00 00:02:33
EXTRACT STOPPED PFBM 00:00:00 00:00:34
REPLICAT STOPPED RZDTA 00:00:00 20:15:10

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) >
```

3. Select the OGG\_SRC window. Verify that there is no lag for the RFBM Replicat, and then stop it:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > lag replicat RFBM
Sending GETLAG request to REPLICAT RFBM ...
Last record lag 6 seconds.
Low watermark lag: 15.
High watermark lag: 6.
Low watermark position: 2127841.
High watermark position: 2127853.
At EOF, no more records to process.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop replicat RFBM
Sending STOP request to REPLICAT RFBM ...
Request processed.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all

Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT STOPPED EZDTA 00:00:00 20:31:43
EXTRACT STOPPED PZDTA 00:00:00 20:30:41
REPLICAT STOPPED RFBM 00:00:00 00:00:51
```

4. You can now re-establish replication from the old database to the new database. First, you need to force the Extract to capture change from this moment. This is accomplished by using the BEGIN NOW syntax. Then you start the EZDTA primary Extract, followed by the PZDTA Data Pump:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > alter extract ezdta, begin now
EXTRACT altered.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start extract ezdta

Sending START request to MANAGER ...
EXTRACT EZDTA starting
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start extract pzdtap

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

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
```

Program Chkpt	Status	Group	Lag at Chkpt	Time Since
MANAGER	RUNNING			
EXTRACT	RUNNING	EZDTA	00:00:00	00:00:09
EXTRACT	RUNNING	PZDTA	00:00:00	20:35:59
REPLICAT	STOPPED	RFBM	00:00:00	00:06:08

```
GGSCI ([Host]) 8> exit
```

5. You must now start the RZDTA Replicat on the Oracle GoldenGate target instance. Select the OGG\_TRG window and start the Replicat RZDTA:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > start replicat rzdata
```

```
Sending START request to MANAGER ...
```

```
REPLICAT RZDTA starting
```

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > info all
```

Program Chkpt	Status	Group	Lag at Chkpt	Time Since
MANAGER	RUNNING			
EXTRACT	STOPPED	EFBM	00:00:00	00:20:38
EXTRACT	STOPPED	PFBM	00:00:00	00:18:40
REPLICAT	RUNNING	RZDTA	00:00:00	00:00:02

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) >
```

6. Select the CDB12c window and execute the `gdp_by_year_2010.sql` and `gdp_growth_by_year_2010.sql` scripts. Connect to the new database and make sure that GDP and GDP growth data have been properly replicated:

```
SQL> @gdp_by_year_2010.sql
1 row created.
...Many lines omitted for brevity...
SQL> @gdp_growth_by_year_2010.sql
1 row created.
...Many lines omitted for brevity...
```

7. Select the DWNSTR window and verify that the latest GDP data related to the year 2010 has been correctly replicated:

```
SQL> select max(gdp_year) from gdp_by_year;

MAX(GDP_YEAR)

2010
SQL> select max(gdp_year) from gdp_growth_by_year;

MAX(GDP_YEAR)

2010
SQL> exit
```

8. The previous step verified that the new database is synchronized with the old database, and it is ready for a new migration cutover attempt.
9. All Extract and Replicat groups must be stopped and deleted in the source and target directories (EFBM, PFMB, and RZDTA in ogg\_trg and EZDTA, PZDTA, and RFBM in ogg\_src). All Oracle RDBMS objects should be dropped and all trail files removed from the dirdat directory. In order to save time, you can run the clean-up.sh script, which you can find in the ~/labs/Practice08/configs directory. **IMPORTANT:** before running the clean-up.sh script, exit from all open sqlplus sessions so that the cleanup script can drop the simulapp user from the database (the drop command will fail if the user being dropped is currently connected.) Make sure no sqlplus session is still connected to the database and then run the clean-up.sh script. A completely new setup will be configured for Practice 9, so you can terminate all terminal shell windows.

This completes Practice 8-4. You can exit GGSCI and sqlplus and terminate all shell windows in your environment. A new setup will be configured for the next practice.

This completes Practice 8. Stop here.



## **Practices for Lesson 9: Bidirectional Replication: Two-Node Configuration**

**Chapter 9**

## Practices for Lesson 9: Overview

---

### Practices Overview

In these practices, you will configure an active-active replication topology that is capable of replicating DML as well as DDL statements. In addition, you will configure Conflict Detection and Resolution (CDR) to automatically detect and resolve data conflicts by using simple rules.

**Helper scripts provided in /home/oracle/labs/Practice02/configs:**

Catch-up/Clean-up script	Purpose
catch-up-prac-9_1.sh	Executes steps for Practice 9-1
catch-up-prac-9_2.sh	Executes steps for Practice 9-2
catch-up-prac-9_3.sh	Executes steps for Practice 9-3
clean-up.sh	Complete clean-up script. It resets the environment as it was before starting Practice 9.

**Note:** The `clean-up.sh` script is available in case you want to reset your environment while you are performing the steps required by the practices. Do not run the `clean-up.sh` script at the end of Practice 9-3, as you need a working environment after all steps in Practice 9-3 are executed, for Practice 10.

## Practice 9-1: Configuring an Active-Active Replication Topology

### Overview

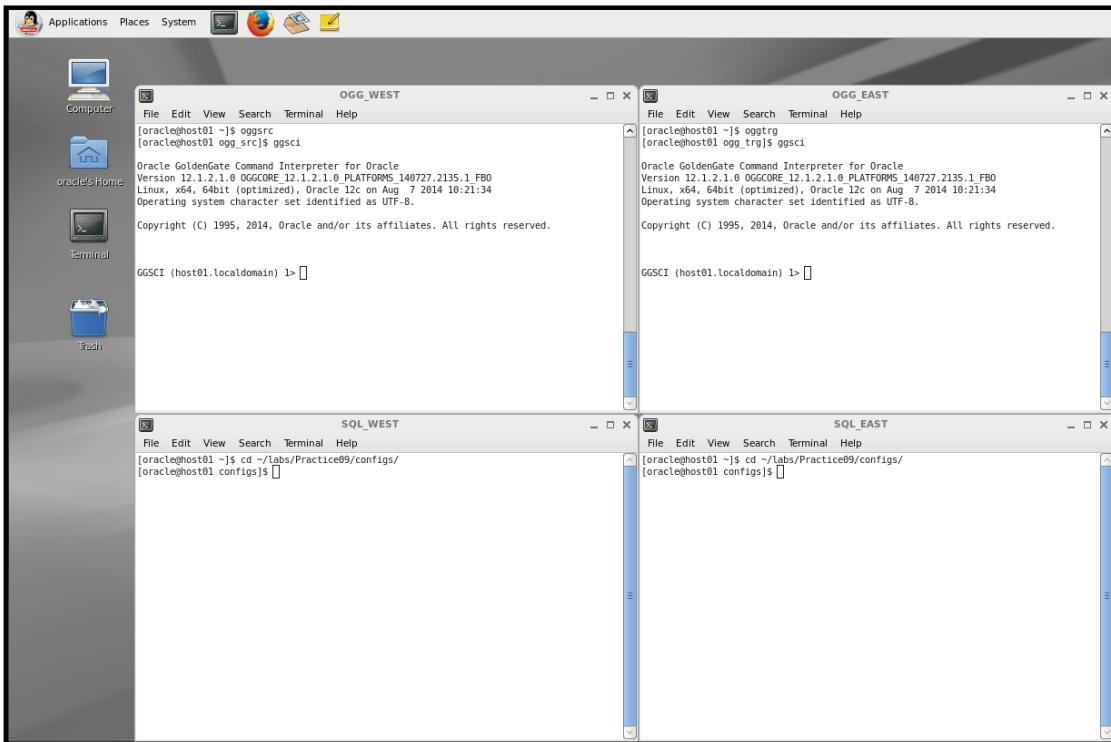
In this practice, you set up a two-way, active-active replication configuration that is capable of replicating DML statements. The WEST schema in the ogg1 PDB in the CDB12c database will be replicated into the EAST schema in the oggdwn1 PDB in the DWNSTR database. In the same manner, the EAST schema in oggdwn1 will be replicated into the WEST schema in ogg1. Initially you will set up simple, DML only replication (Practice 9-1). Subsequently you will enable DDL replication (Practice 9-2), and finally you will enable CDR-based conflict detection and resolution (Practice 9-3).

### Assumptions

- The CDB12c and the DWNSTR databases are up and running.
- The OGGDATA tablespace exists and contains enough free space to accommodate the objects created during this practice.
- You have access as sys and dba to both RDBMS instances (you can connect via slash (/) from your oracle account).

### Tasks

1. Configure your desktop in the usual manner, with four terminal shell windows aligned to form a two-by-two matrix. Name the top left window "OGG\_WEST." At the OS prompt, enter the "oggsrc" alias to position the default directory to /u03/ogg/ogg\_src and launch GGSCI. Name the top right window "OGG\_EAST" and at the OS prompt, use the alias oggtrg to navigate to the /u03/ogg/ogg\_trg directory and launch GGSCI. Name the bottom left window "SQL\_WEST" and at the OS prompt, navigate to the ~/labs/Practice09/configs directory. Name the bottom right window "SQL\_EAST" and at the OS prompt, navigate to the ~/labs/Practice09/configs directory.



Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

2. Select the SQL\_WEST window. Set the ORACLE\_SID environment variable to CDB12c. Launch sqlplus, connecting as sys to the CDB12c database. Alter your session and set the default PDB to be ogg1. Create the west user, assigning the OGGDATA tablespace as the default tablespace. Then grant the roles connect, resource and unlimited tablespace to west:

```
[OS prompt]$ export ORACLE_SID=CDB12c
[OS prompt]$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Wed Mar 18 19:20:29
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> alter session set container = ogg1;
Session altered.
SQL> create user west identified by Welcome1 default tablespace
oggdata temporary tablespace temp;
User created.
SQL> grant connect, resource, unlimited tablespace to west;
Grant succeeded.
```

3. Connect to the CDB12c database, ogg1 PDB as west (password Welcome1) and run the object creation script (objects.sql):

```
SQL> connect west/Welcome1@ogg1
Connected.
SQL> set echo on
SQL> @objects.sql
SQL> CREATE TABLE ACCOUNT
 2 (
 ... Many lines omitted for clarity ...
Table altered.
SQL>
```

4. Select the SQL\_EAST window. Set the ORACLE\_SID environment variable to DWNSTR. Launch sqlplus, connecting as sys to the DWNSTR database. Alter your session and set the default PDB to be oggdwn1. Create the east user, assigning the OGGDATA tablespace as the default tablespace. Then grant the roles connect, resource and unlimited tablespace to east:

```
[OS prompt]$ export ORACLE_SID=DWNSTR
[OS prompt]$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Wed Mar 18 19:20:29
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
```

```

Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production

With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> alter session set container = oggdwn1;
Session altered.

SQL> create user east identified by Welcome1 default tablespace
oggdata temporary tablespace temp;
User created.

SQL> grant connect, resource, unlimited tablespace to east;
Grant succeeded.

```

5. Connect to the DWNSTR database as east (password Welcome1) and run the object creation script (objects.sql):

```

SQL> connect east/Welcome1@oggdwn1
Connected.

SQL> set echo on
SQL> @objects.sql
SQL> CREATE TABLE ACCOUNT
 2 (
... Many lines omitted for clarity ...
Table altered.

SQL>

```

6. 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 logs. In GGSCI, this is accomplished by using the ADD SCHEMATRANDATA command. Select the OGG\_WEST window and connect to the CDB12c database by using the ogg\_admin alias. Then issue the ADD SCHEMATRANDATA command for the WEST schema in the ogg1 PDB:

```

GGSCI ([Host]) > DBLogin UserIdAlias ogg_admin
Successfully logged into database CDB$ROOT.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add
SchemaTranData ogg1.west
2015-03-18 22:29:13 INFO OGG-01788 SCHEMATRANDATA has been
added on schema west.
2015-03-18 22:29:13 INFO OGG-01976 SCHEMATRANDATA for
scheduling columns has been added on schema west.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) >

```

7. You need to add some credentials to allow the Integrated Replicat process to connect to the target database (it cannot connect to the root container, so the ogg\_admin and ogg\_dwnadm aliases cannot be used.) Note: there is no harm in defining the alias credentials for a user already in the credential store. If the user ogg1\_repl is already defined you will simply see an error message ("Entry already exists in credential store.") which you can safely disregard:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > alter
credentialstore add user C##OGG_ADMIN@ogg1 password oracle alias
ogg1_repl
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > alter
credentialstore add user C##OGG_DWNADM@oggdwn1 password oracle
alias ogg_dwnrepl
```

8. Repeat what you have done so far to the east schema on the oggdwn1 PDB of the DWNSTR database. Select the OGG\_EAST window and connect to the DWNSTR database as the Oracle GoldenGate user alias ogg\_dwnadm:

```
GGSCI ([Host]) 1> DBLogin UserIdAlias ogg_dwnadm
Successfully logged into database CDB$ROOT.
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > add
SchemaTranData oggdwn1.east
2015-03-18 22:36:13 INFO OGG-01788 SCHEMATRANDATA has been
added on schema east.
2015-03-18 22:36:13 INFO OGG-01976 SCHEMATRANDATA for
scheduling columns has been added on schema east.
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > alter
credentialstore add user C##OGG_ADMIN@ogg1 password oracle alias
ogg1_repl
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > alter
credentialstore add user C##OGG_DWNADM@oggdwn1 password oracle
alias ogg_dwnrepl
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) >
```

The last step completes the database/Oracle GoldenGate preliminary setup for active-active replication. You must now configure the change data capture from west to east, from east to west, and then the delivery (both ways).

9. Select the OGG\_WEST window. Make sure no other Extract or Replicat groups exist (use INFO ALL to display all groups). If any group that was set up for the previous practices is still present, stop it first, and then delete it by using DELETE EXTRACT \* and DELETE REPLICAT \*. If INFO ALL displays only the manager, there is no need for cleanup.

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
```

10. You now configure the primary Extract group (exwest) for change data capture (west to east):

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
exwest
```

```

Extract exwest
SETENV (ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV (ORACLE_SID = "CDB12c")
ExtTrail ./dirdat/ew
UserIdAlias ogg_admin
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
StatOptions ResetReportStats
ReportCount Every 10 Records
TranLogOptions ExcludeUser C##OGG_ADMIN
SOURCECATALOG ogg1
Table WEST.*;

```

Note the TRANLOGOPTIONS parameter, which excludes the Replicat user from propagating transactions. TRANLOGOPTIONS must precede all TABLE statements.

11. Register the EXWEST Integrated Extract with the database (ogg1). Add the Extract group by executing the following GGSCI commands:

- Add Extract exwest, Integrated Tranlog, Begin Now  
This adds the exwest Extract to GoldenGate, specifying that it will read from Oracle RDBMS Redo/Archive logs, with an initial Redo checkpoint of the current time stamp.
- Add ExtTrail ./dirdat/ew, Extract exwest, Megabytes 10  
This adds the ./dirdat/ew Local Extract Trail, with a maximum file size of 10 megabytes per trail, and links the trail to the exwest Extract.

```

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > REGISTER
EXTRACT EXWEST DATABASE CONTAINER (ogg1)

Extract EXWEST successfully registered with database at SCN
3226647.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > Add Extract
exwest, Integrated TranLog, Begin Now

EXTRACT added.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > Add ExtTrail
./dirdat/ew, Extract exwest, Megabytes 10

EXTTRAIL added.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) >

```

12. On the CDB12c database server, configure the Extract Data Pump to read from the ./dirdat/ew (Extract West) Local Extract Trail, transmit the data to the DWNSTR server, and write it to the ./dirdat/pe (Pump East) Remote Trail:

From the GGSCI prompt, execute the following command:

```
GGSCI ([Host]) 9> Edit Param dpwest
```

Verify that the dpwest parameter file includes the following:

```

Extract dpwest
SETENV (ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV (ORACLE_SID='CDB12c')

```

```

UserIdAlias ogg_admin
RMTHOST ogg_target, MGRPORT 7909
RmtTrail ./dirdat/pe
Passthru
SOURCECATALOG ogg1
Table WEST.*;

```

Save and close the file.

- Add the Extract Group by executing the following GGSCI commands:

```

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > Add Extract
dpwest, ExtTrailSource ./dirdat/ew
EXTRACT added.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > Add RmtTrail
./dirdat/pe, Extract dpwest, Megabytes 10
RMTTRAIL added.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) >

```

- To set up Oracle GoldenGate Delivery on the CDB12c database, ogg1 PDB, west schema, you must configure the Replicat component. From a GGSCI prompt, execute the following command:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > Edit Param
rewest
```

Your text editor of choice starts and opens the `rewest` file. Add the following runtime parameters to the text file:

```

Replicat rewtest
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV (ORACLE_SID='CDB12c')
DiscardFile ./dirrpt/rewest.dsc, Purge
UserIdAlias ogg1_repl
AssumeTargetDefs
Map oggdwn1.EAST.* , Target ogg1.WEST.*;

```

Save and close the file.

- Add the Replicat by executing the following GGSCI command:

```

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > Add Replicat
rewest, integrated ExtTrail ./dirdat/pw
REPLICAT (Integrated) added.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) >

```

16. Repeat the above steps for the DWNSTR database, oggdwn1 PDB, east schema. Click the OGG\_EAST window and make sure no other Extract or Replicat groups exist (use INFO ALL to display all groups). If any group that was set up for the previous practices is still present, delete it by using DELETE EXTRACT \* and DELETE REPLICAT \*. If INFO ALL displays only the manager, there is no need for cleanup.

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
```

17. You now configure the primary Extract group (exeast) for change data capture (east to west):

```
GGSCI ([Host]) > edit params exeast
Extract exeast
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "DWNSTR")
ExtTrail ./dirdat/ee
UserIdAlias ogg_dwnadm
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
StatOptions ResetReportStats
ReportCount Every 10 Records
TranLogOptions ExcludeUser C##OGG_DWNADM
SOURCECATALOG oggdwn1
Table EAST.*;
```

18. Register the EXEAST Extract with the database (oggdwn1). Add the Extract group by executing the following GGSCI commands:

- Add Extract exeast, Tranlog, Begin Now  
This adds the exeast Extract to GoldenGate, specifying that it will read from Oracle DB Redo/Archive logs, with an initial Redo checkpoint of the current time stamp.
- Add ExtTrail ./dirdat/ee, Extract exeast, Megabytes 10  
This adds the ./dirdat/ee Local Extract Trail, with a maximum file size of 10 megabytes per trail, and links the trail to the exeast Extract.

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > REGISTER
EXTRACT EXEAST DATABASE CONTAINER (oggdwn1)
Extract EXEAST successfully registered with database at SCN
2191813.

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > Add Extract
exeast, Integrated TranLog, Begin Now
EXTRACT added.

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > Add ExtTrail
./dirdat/ee, Extract exeast, Megabytes 10
EXTTRAIL added.

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) >
```

19. On the DWNSTR database server, configure the Extract Data Pump to read from the ./dirdat/ee (Extract East) Local Extract Trail, transmit the data to the CDB12c server, and write it to the ./dirdat/pw (Pump West) Remote Trail:

From the GGSCI prompt, execute the following command:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > Edit Param
dpeast
```

Verify that the dpeast parameter file includes the following:

```
Extract dpeast
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "DWNSTR")
RmtHost ogg_source, MgrPort 7809
UserIdAlias ogg_dwnadm
RmtTrail ./dirdat/pw
Passthru
SOURCECATALOG oggdwn1
Table EAST.*;
```

Save and close the file.

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

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > Add Extract
dpeast, ExtTrailSource ./dirdat/ee
EXTRACT added.
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > Add RmtTrail
./dirdat/pw, Extract dpeast, Megabytes 10
RMTTRAIL added.
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) >
```

21. To set up Oracle GoldenGate Delivery on the DWNSTR database, oggdwn1 PDB, east schema, you must configure the Replicat component. From a GGSCI prompt, execute the following command:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > Edit Param
reeast
```

Your text editor of choice starts and opens the `reeast` file. Add the following runtime parameters to the text file:

```
Replicat reeast
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "DWNSTR")
DiscardFile ./dirrpt/reeast.dsc, Purge
UserIDAlias ogg_dwnrepl
AssumeTargetDefs
Map oggl.WEST.* , Target oggdwn1.EAST.*;
```

Save and close the file.

22. Add the Replicat by executing the following GGSCI command:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > Add Replicat
reeast, Integrated ExtTrail ./dirdat/pe
REPLICAT (Integrated) added.
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) >
```

23. The infrastructure for active-active DML replication is now in place. You can start all Extract and Replicat groups for both east and west. Select the OGG\_WEST window and start the Extract and Replicat groups at the same time by using the `start ER *` command:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start ER *
Sending START request to MANAGER ...
EXTRACT DPWEST starting

Sending START request to MANAGER ...
EXTRACT EXWEST starting

Sending START request to MANAGER ...
REPLICAT REWEST starting
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING DPWEST 00:00:00 26:14:36
EXTRACT RUNNING EXWEST 00:00:00 00:00:07
REPLICAT RUNNING REWEST 00:00:00 00:00:02
```

24. Select the OGG\_EAST window and start Extract and Replicat groups at the same time by using the `start ER *` command:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > start ER *
Sending START request to MANAGER ...
EXTRACT DPEAST starting

Sending START request to MANAGER ...
EXTRACT EXEAST starting

Sending START request to MANAGER ...
REPLICAT REEAST starting
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > info all

Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING DPEAST 00:00:00 00:00:01
EXTRACT RUNNING EXEAST 00:05:43 00:00:06
REPLICAT RUNNING REEAST 00:00:00 00:00:06
```

25. Click the SQL\_WEST window, where sqlplus is running connected to the ogg1 PDB as the WEST schema. Execute the account\_west.sql file:

```
SQL> @account_west.sql
1 row created.
... Many lines omitted for clarity ...
Commit complete.
SQL> select count(*) from account;

COUNT(*)

180
```

26. Connected to the west schema, you inserted 180 account rows. Click the SQL\_EAST window. Verify that the 180 rows that were inserted while connected as west on the OGG11R2 database, ogg1 PDB, have been correctly replicated to the east schema in the oggdwn1 PDB of the DWNSTR database:

```
SQL> select count(*) from account;

COUNT(*)

180
```

27. Select the OGG\_WEST window and check the Extract statistics by using the STATS command:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stats exwest,
total

Sending STATS request to EXTRACT EXWEST ...

Start of Statistics at 2015-03-19 00:16:44.

Output to ./dirdat/ew:
Extracting from OGG1.WEST.ACCOUNT to OGG1.WEST.ACCOUNT:
*** Total statistics since 2015-03-19 00:13:09 ***
 Total inserts 180.00
 Total updates 0.00
 Total deletes 0.00
 Total discards 0.00
 Total operations 180.00
End of Statistics.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stats dpwest,
total

Sending STATS request to EXTRACT DPWEST ...
Start of Statistics at 2015-03-19 00:17:35.

Output to ./dirdat/pe:
```

```
Extracting from OGG1.WEST.ACCOUNT to OGG1.WEST.ACCOUNT:
*** Total statistics since 2015-03-19 00:13:09 ***
Total inserts 180.00
Total updates 0.00
Total deletes 0.00
Total discards 0.00
Total operations 180.00
End of Statistics.
```

28. Select the OGG\_EAST window and check the Replicat statistics by using the STATS command:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > stats reeast,
total
Sending STATS request to REPLICAT REEAST ...
Start of Statistics at 2015-03-19 00:19:48.
Integrated Replicat Statistics:
Total transactions 1.00
Redirected 0.00
DDL operations 0.00
Stored procedures 0.00
Datatype functionality 0.00
Event actions 0.00
Direct transactions ratio 0.00%
Replicating from OGG1.WEST.ACCOUNT to OGGDWN1.EAST.ACCOUNT:
*** Total statistics since 2015-03-19 00:13:13 ***
Total inserts 180.00
Total updates 0.00
Total deletes 0.00
Total discards 0.00
Total operations 180.00
End of Statistics.
```

29. Select the SQL\_EAST window, where sqlplus is running connected to the east/OGGDWN1/DWNSTR database. Execute the branch\_east.sql script, and then verify that the rows that were inserted have been replicated to the west schema in the ogg1 PDB of the CDB12c database:

```
SQL> @branch_east.sql
1 row created.
Commit complete.
```

30. Select the SQL\_WEST window, where sqlplus is running connected to the west/OGG1/CDB12c database, and verify that five branches have been added:

```
SQL> show user
USER is "WEST"
SQL> select count(*) from branch;

COUNT(*)

 5
```

31. Select the OGG\_WEST window and check the Replicat statistics by using the STATS command:

```
GGSCI ([Host]) 18> stats rewest, total
Sending STATS request to REPLICAT REWEST ...
Start of Statistics at 2015-03-19 00:24:41.
Integrated Replicat Statistics:
 Total transactions 1.00
 Redirected 0.00
 DDL operations 0.00
 Stored procedures 0.00
 Datatype functionality 0.00
 Event actions 0.00
 Direct transactions ratio 0.00%
Replicating from OGGDWN1.EAST.BRANCH to OGG1.WEST.BRANCH:
*** Total statistics since 2015-03-19 00:22:31 ***
 Total inserts 5.00
 Total updates 0.00
 Total deletes 0.00
 Total discards 0.00
 Total operations 5.00
End of Statistics.
```

32. Repeat the above steps to load the data for the ACCOUNT table on the EAST database and the BRANCH table on the WEST database. The CUSTOMER table must be loaded in both databases. Select the SQL\_EAST window, where sqlplus is running connected to the east/oggdwn1/DWNSTR database, and run the customer\_east.sql and account\_east.sql scripts:

```
SQL> @customer_east.sql
1 row created.
... Many lines omitted for clarity ...
Commit complete.
SQL> select count(*) from customer;

COUNT(*)
```

```

 100
SQL> @account_east.sql
1 row created.
... Many lines omitted for clarity ...
Commit complete.
SQL> select count(*) from account;

COUNT (*)

 360
```

33. There are now 360 rows for the ACCOUNT table. 180 rows were loaded from the WEST side, 180 rows were loaded from the EAST side, and the two databases converged to give a total of 360 rows. Select the SQL\_WEST window, where sqlplus is running connected to the west/ogg1/OGG11R2 database, and run the customer\_west.sql and branch\_west.sql scripts:

```
SQL> @customer_west.sql
1 row created.
... Many lines omitted for clarity ...
Commit complete.
SQL> select count(*) from customer;

COUNT (*)

 200
SQL> @branch_west.sql
1 row created.
... Several lines omitted for clarity ...
Commit complete.
SQL> select count(*) from branch;

COUNT (*)

 10
```

There are now 200 rows for the CUSTOMER table. 100 rows were loaded from the WEST side, 100 rows were loaded from the EAST side, and the two databases converged to give a total of 200 rows. The same applies to the BRANCH table: Five rows were loaded from the WEST side and five rows were loaded from the EAST side, for a total of 10 rows.

34. You have now configured an active-active basic replication topology. It is basic because it replicates only DML statements, and is unable to detect and resolve data conflicts. Adding DDL replication capabilities is the subject of the next practice.

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

## Practice 9-2: Setting Up DDL Replication and Verifying Correct Propagation of DML and DDL Statements

### Overview

In this practice, you expand on what you accomplished in the previous practice by adding DDL replication. You will modify the Extract and Replicat groups that you created in Practice 9-1 to include DDL-related parameters. You will use native DDL capture (also known as triggerless DDL capture). Starting with release 11.2.0.4 of the Oracle database, trigger-based DDL replication is not required. By default, DDL capture is handled transparently through the database logmining server. If your Extract groups capture from a multitenant container database, as in the case of this practice, Integrated Capture mode must be used with the native DDL capture method.

In case of Classic capture (against any version of the Oracle RDBMS), trigger-based DDL replication must be used. In this practice, you will implement native DDL replication via Integrated Capture set up on a multitenant database.

### Assumptions

Practice 9-1 has been completed successfully and all Extract and Replicat groups are running.

### Tasks

- Click the OGG\_WEST window and stop the exwest Extract group and the rewest Replicat group:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING DPWEST 00:00:00 00:00:01
EXTRACT RUNNING EXWEST 00:00:00 00:00:08
REPLICAT RUNNING REWEST 00:00:00 00:00:02

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop extract
exwest
Sending STOP request to EXTRACT EXWEST ...
Request processed.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop replicat
rewest
Sending STOP request to REPLICAT REWEST ...
Request processed.
```

In order to enable DDL replication, you must include the `DDL` and optionally the `DDLOPTIONS` commands in the parameter file for the Extract group. DDL replication is enabled by default in Replicat groups.

Database objects are classified into *scopes*. A scope is a category that defines how DDL operations on an object are handled by Oracle GoldenGate. The scopes are:

- MAPPED

- UNMAPPED
- OTHER

The use of scopes enables granular control over the filtering of DDL operations, string substitutions, and error handling.

Objects that are specified in TABLE and MAP statements are of MAPPED scope.

If a DDL operation is supported for use in a TABLE or MAP statement, but its base object name is not included in one of those parameters, it is of UNMAPPED scope.

DDL operations that cannot be mapped are of OTHER scope. When a DDL operation is of OTHER scope in the Replicat configuration, it is applied to the target with the same owner and object name as in the source DDL. In this example, you specifically instruct Oracle GoldenGate to replicate the MAPPED objects to the west and the east schemas. In addition, you use the ADDTRANDATA option to instruct the Extract group to automatically log the keys for the tables that are created via DDL statements. Finally, you instruct the rewest and reeast Replicat groups to ignore DDL replication errors by using the DDLERROR DEFAULT IGNORE command.

2. Edit the parameter file for the exwest Extract group and add DDL replication:

```
GGSCI ([Host]) 4> edit params exwest
Extract exwest
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "CDB12c")
ExtTrail ./dirdat/ew
UserIdAlias ogg_admin
DDL INCLUDE MAPPED OBJNAME oggl.west.*
DDLOPTIONS ADDTRANDATA
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
StatOptions ResetReportStats
ReportCount Every 10 Records
TranLogOptions ExcludeUser C##OGG_ADMIN
SOURCECATALOG oggl
Table WEST.*;
```

3. Save the parameter file. Edit the parameter file for the rewest Replicat group:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
rewest
```

```
Replicat rewest
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV (ORACLE_SID='CDB12c')
DiscardFile ./dirrpt/rewest.dsc, Purge
UserIdAlias ogg1_repl
DDLERROR DEFAULT IGNORE
AssumeTargetDefs
Map oggdwn1.EAST.* , Target oggl.WEST.*;
```

4. Save the `rewest` parameter file and start both Extract and Replicat groups:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start ext
exwest
Sending START request to MANAGER ...
EXTRACT EXWEST starting
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start rep
rewest
Sending START request to MANAGER ...
REPLICAT REWEST starting
```

5. Click the OGG\_EAST window and apply the same changes to the DWNSTR/oggdwn1/east schema:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt
MANAGER RUNNING
EXTRACT RUNNING DPEAST 00:00:00 00:00:02
EXTRACT RUNNING EXEAST 00:00:00 00:00:03
REPLICAT RUNNING REEAST 00:00:00 00:00:05
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > stop extract
exeast
Sending STOP request to EXTRACT EXEAST ...
Request processed.
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > stop replicat
reeast
Sending STOP request to REPLICAT REEAST ...
Request processed.
```

6. Edit the `exeast` parameter file and enable DDL replication:

```
GGSCI ([Host]) 3> edit params exeast
```

```
Extract exeast
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "DWNSTR")
ExtTrail ./dirdat/ee
UserIdAlias ogg_dwnadm
DDL INCLUDE MAPPED OBJNAME oggdwn1.east.*
DDLOPTIONS ADDTRANDATA
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
StatOptions ResetReportStats
ReportCount Every 10 Records
TranLogOptions ExcludeUser C##OGG_DWNADM
SOURCECATALOG oggdwn1
Table EAST.*;
```

7. Save the parameter file. Edit the parameter file for the `reeast` Replicat group:

```
GGSCI ([Host]) 4> edit params reeast
```

```
Replicat reeast
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "DWNSTR")
DiscardFile ./dirrpt/reeast.dsc, Purge
UserIDAlias ogg_dwnrep1
DDLError DEFAULT IGNORE
AssumeTargetDefs
Map ogg1.WEST.* , Target oggdwn1.EAST.*;
```

8. Save the `reeast` parameter file and start both Extract and Replicat groups:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > start ext
 reeast
```

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

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > start rep
 reeast
```

```
Sending START request to MANAGER ...
REPLICAT REEAST starting
```

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > info all
```

Program Chkpt	Status	Group	Lag at Chkpt	Time Since
MANAGER	RUNNING			
EXTRACT	RUNNING	DPEAST	00:00:00	00:00:08
EXTRACT	RUNNING	EXEAST	00:00:00	00:00:06
REPLICAT	RUNNING	REEAST	00:00:00	00:00:06

9. Click the `SQL_WEST` window and create a new table; add a row and verify that both DDL and DML statements are replicated to the `DWNSTR` database, `oggdwn1` PDB, `east` schema:

```
SQL> create table try_ddl (field_1 varchar2(30) not null primary key,
 field_2 number);
Table created.
SQL> insert into try_ddl values ('GoldenGate',1);
1 row created.
SQL> commit;
Commit complete.
SQL>
```

10. Click the SQL\_EAST window and verify that the TRY\_DDL table has been replicated, together with the values in its two columns:

```
SQL> connect east/Welcomel@oggdwn1
Connected.
SQL> select table_name from user_tables;
TABLE_NAME

ACCOUNT
CUSTOMER
TRY_DDL
BRANCH
SQL> select * from try_ddl;
FIELD_1 FIELD_2

GoldenGate 1
SQL>
```

The try\_ddl table, created in the west schema, as well as the row added to it, has been correctly replicated to the east schema. You have configured an active-active topology that is capable of replicating both DML and DDL statements. Data conflicts are not yet managed; that is the topic of the next practice.

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

## Practice 9-3: Enabling CDR-Based Conflict Detection and Resolution

### Overview

In this practice, you configure CDR conflict detection and resolution by using a simple time stamp-based policy to resolve update conflicts for all columns except account balance and a delta resolution method for account balance. In addition, you create alternate sequence generators to avoid insert conflicts.

Because Oracle GoldenGate is an asynchronous solution, in an active-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 account balance values when the application uses update statements to modify the account balance column. The premise for this demonstration is an online retail environment where customers purchase credit before placing orders, and their balance is updated when they buy goods:

1. Customer David Keen owns the account number 299, which has a balance of \$7,192. He connects to your retail system from his office and places an order for \$30. The load-balancing algorithm routes Mr. Keen to the `west` database.
  - The SQL statement executed is:

```
UPDATE account set account_balance = account_balance - 30
 where acct_id = 299 and account_balance = 7192;
```
2. His wife connects from home to your online retail outlet, using her husband's credentials, and places an order for \$50. The load-balancing algorithm routes Mrs. Keen to the `east` database.
  - The SQL statement executed is:

```
UPDATE account set account_balance = account_balance - 50
 where acct_id = 299 and account_balance = 7192;
```

In this example, the account balance for account ID 299 in the `west` database is \$7,162, and the account balance for the same account in the `east` database is \$7,142.

With conflict detection enabled, when Oracle GoldenGate attempts to update the `east` database with the transaction for account ID 299, routed through `west`, the transaction will fail because the before image (from `west`) of `ACCOUNT_BALANCE` is \$7,192, which does not match the current column value of \$7,142 in `east`. Conversely, the transaction replicated to the `west` database will also fail because the before image (from `east`) of `ACCOUNT_BALANCE`, \$7,192, does not match the current column value of \$7,162 in `west`.

To resolve the conflict, you will add code in the Replicat Delivery configuration to add the sum total of the account balance from both sites, so that the resulting `ACCOUNT_BALANCE`

values after the replicated transactions are applied will be the same on both sites. The result is \$7,112.

## Assumptions

- Practices 9-1 and 9-2 have been successfully completed.
- All database objects are created and available.
- All Extract and Replicat groups are in the RUNNING state.

## Tasks

1. Click the OGG\_WEST window and enable supplemental logging for the tables in the ogg1 PDB, west schema. The COLS option adds more than the primary key to the Before Image. The ADD TRANDATA command enables supplemental logging for primary keys by default, so you do not specify the primary key columns when listing the additional columns that you want to log.

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info trandata
ogg1.west.account
2015-03-19 22:01:23 INFO OGG-06480 Schema level
supplemental logging, excluding non-validated keys, is enabled
on schema WEST.

2015-03-19 22:01:23 INFO OGG-01980 Schema level
supplemental logging is enabled on schema WEST for all
scheduling columns.

Logging of supplemental redo log data is enabled for table
OGG1.WEST.ACCT.

Columns supplementally logged for table OGG1.WEST.ACCT:
ACCT_ID, CUST_ID.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add trandata
ogg1.west.account, cols
(ACCOUNT_TYPE, ACCOUNT_NUMBER, ACCOUNT_BALANCE, ACCOUNT_OPENED, ACCO
UNT_BRANCH_ID, UPDATE_TS)

Logging of supplemental redo data enabled for table
OGG1.WEST.ACCT.

TRANDATA for scheduling columns has been added on table
'OGG1.WEST.ACCT'.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info trandata
ogg1.west.account
2015-03-19 22:02:59 INFO OGG-06480 Schema level
supplemental logging, excluding non-validated keys, is enabled
on schema WEST.

2015-03-19 22:02:59 INFO OGG-01980 Schema level
supplemental logging is enabled on schema WEST for all
scheduling columns.

Logging of supplemental redo log data is enabled for table
OGG1.WEST.ACCT.

Columns supplementally logged for table OGG1.WEST.ACCT:
ACCOUNT_BALANCE, ACCOUNT_BRANCH_ID, ACCOUNT_NUMBER,
ACCOUNT_OPENED, ACCOUNT_TYPE, ACCT_ID, CUST_ID, UPDATE_TS.
```

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add trandata
ogg1.west.branch,
cols(BRANCH_STREET_TYPE,BRANCH_STREET_NAME,BRANCH_STREET_NUMBER,
BRANCH_LEVEL,BRANCH_CITY,BRANCH_STATE,BRANCH_COUNTRY,BRANCH_SIZE
,BRANCH_OPENED,ZIP_CODE,BRANCH_NAME,UPDATE_TS)
Logging of supplemental redo data enabled for table
OGG1.WEST.BRANCH.

TRANDATA for scheduling columns has been added on table
'OGG1.WEST.BRANCH'.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info trandata
ogg1.west.branch
2015-03-19 22:16:33 INFO OGG-06480 Schema level
supplemental logging, excluding non-validated keys, is enabled
on schema WEST.

2015-03-19 22:16:33 INFO OGG-01980 Schema level
supplemental logging is enabled on schema WEST for all
scheduling columns.

Logging of supplemental redo log data is enabled for table
OGG1.WEST.BRANCH.

Columns supplementally logged for table OGG1.WEST.BRANCH:
BRANCH_CITY, BRANCH_COUNTRY, BRANCH_ID, BRANCH_LEVEL,
BRANCH_NAME, BRANCH_OPENED, BRANCH_SIZE, BRANCH_STATE,
BRANCH_STREET_NAME, BRANCH_STREET_NUMBER, BRANCH_STREET_TYPE,
UPDATE_TS, ZIP_CODE.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add trandata
ogg1.west.customer,
cols(NAME,MIDDLE_NAME,SURNAME,FULL_NAME,DOB,NATIONALITY,EMAIL,UP
DATE_TS)
Logging of supplemental redo data enabled for table
OGG1.WEST.CUSTOMER.

TRANDATA for scheduling columns has been added on table
'OGG1.WEST.CUSTOMER'.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info trandata
ogg1.west.customer
2015-03-19 22:18:41 INFO OGG-06480 Schema level
supplemental logging, excluding non-validated keys, is enabled
on schema WEST.

2015-03-19 22:18:41 INFO OGG-01980 Schema level
supplemental logging is enabled on schema WEST for all
scheduling columns.

Logging of supplemental redo log data is enabled for table
OGG1.WEST.CUSTOMER.

Columns supplementally logged for table OGG1.WEST.CUSTOMER:
CUST_ID, DOB, EMAIL, FULL_NAME, MIDDLE_NAME, NAME, NATIONALITY,
SURNAME, UPDATE_TS.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) >
```

2. Repeat the same step for the oggdwn1 PDB, east schema. Click the OGG\_EAST window and enter the ADD TRANDATA command for each table, specifying the list of columns:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > info trandata
oggdwn1.east.account
2015-03-19 22:04:15 INFO OGG-06480 Schema level
supplemental logging, excluding non-validated keys, is enabled
on schema EAST.

2015-03-19 22:04:15 INFO OGG-01980 Schema level
supplemental logging is enabled on schema EAST for all
scheduling columns.

Logging of supplemental redo log data is enabled for table
OGGDWN1.EAST.ACCOUNT.

Columns supplementally logged for table OGGDWN1.EAST.ACCOUNT:
ACCT_ID, CUST_ID.

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > add trandata
oggdwn1.east.account, cols
(ACCOUNT_TYPE,ACCOUNT_NUMBER,ACCOUNT_BALANCE,ACCOUNT_OPENED,ACCO
UNT_BRANCH_ID,UPDATE_TS)

Logging of supplemental redo data enabled for table
OGGDWN1.EAST.ACCOUNT.

TRANDATA for scheduling columns has been added on table
'OGGDWN1.EAST.ACCOUNT'.

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > info trandata
oggdwn1.east.account
2015-03-19 22:30:38 INFO OGG-06480 Schema level
supplemental logging, excluding non-validated keys, is enabled
on schema EAST.

2015-03-19 22:30:38 INFO OGG-01980 Schema level
supplemental logging is enabled on schema EAST for all
scheduling columns.

Logging of supplemental redo log data is enabled for table
OGGDWN1.EAST.ACCOUNT.

Columns supplementally logged for table OGGDWN1.EAST.ACCOUNT:
ACCOUNT_BALANCE, ACCOUNT_BRANCH_ID, ACCOUNT_NUMBER,
ACCOUNT_OPENED, ACCOUNT_TYPE, ACCT_ID, CUST_ID, UPDATE_TS.

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > add trandata
oggdwn1.east.branch,
cols(BRANCH_STREET_TYPE,BRANCH_STREET_NAME,BRANCH_STREET_NUMBER,
BRANCH_LEVEL,BRANCH_CITY,BRANCH_STATE,BRANCH_COUNTRY,BRANCH_SIZE
,BRANCH_OPENED,ZIP_CODE,BRANCH_NAME,UPDATE_TS)

Logging of supplemental redo data enabled for table
OGGDWN1.EAST.BRANCH.

TRANDATA for scheduling columns has been added on table
'OGGDWN1.EAST.BRANCH'.

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > info trandata
oggdwn1.east.branch
```

```

2015-03-19 22:33:37 INFO OGG-06480 Schema level
supplemental logging, excluding non-validated keys, is enabled
on schema EAST.

2015-03-19 22:33:37 INFO OGG-01980 Schema level
supplemental logging is enabled on schema EAST for all
scheduling columns.

Logging of supplemental redo log data is enabled for table
OGGDWN1.EAST.BRANCH.

Columns supplementally logged for table OGGDWN1.EAST.BRANCH:
BRANCH_CITY, BRANCH_COUNTRY, BRANCH_ID, BRANCH_LEVEL,
BRANCH_NAME, BRANCH_OPENED, BRANCH_SIZE, BRANCH_STATE,
BRANCH_STREET_NAME, BRANCH_STREET_NUMBER, BRANCH_STREET_TYPE,
UPDATE_TS, ZIP_CODE.

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > add trandata
oggdwn1.east.customer,
cols(NAME,MIDDLE_NAME,SURNAME,FULL_NAME,DOB,NATIONALITY,EMAIL,UP
DATE_TS)

Logging of supplemental redo data enabled for table
OGGDWN1.EAST.CUSTOMER.

TRANSDATA for scheduling columns has been added on table
'OGGDWN1.EAST.CUSTOMER'.

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > info trandata
oggdwn1.east.customer

2015-03-19 22:34:50 INFO OGG-06480 Schema level
supplemental logging, excluding non-validated keys, is enabled
on schema EAST.

2015-03-19 22:34:50 INFO OGG-01980 Schema level
supplemental logging is enabled on schema EAST for all
scheduling columns.

Logging of supplemental redo log data is enabled for table
OGGDWN1.EAST.CUSTOMER.

Columns supplementally logged for table OGGDWN1.EAST.CUSTOMER:
CUST_ID, DOB, EMAIL, FULL_NAME, MIDDLE_NAME, NAME, NATIONALITY,
SURNAME, UPDATE_TS.

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) >

```

3. Click the OGG\_WEST window and stop the EXWEST Extract and the REWEST Replicat. You will add CDR commands to both, editing their respective parameter files. On the data capture side (Extract), you add the GetUpdateBefores and GetBeforeCols commands. On the change delivery side (Replicat), you add several options to the MAP commands, in addition to GetUpdates, GetInserts, and GetDeletes:

```

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all

Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING DPWEST 00:00:00 00:00:06
EXTRACT RUNNING EXWEST 00:00:00 00:00:07

```

```
REPLICAT RUNNING REWEST 00:00:00 00:00:08
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop exwest
```

Sending STOP request to EXTRACT EXWEST ...  
Request processed.

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop rewest
```

Sending STOP request to REPLICAT REWEST ...  
Request processed.

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
exwest
```

```
Extract exwest
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "CDB12c")
ExtTrail ./dirdat/ew
UserIdAlias ogg_admin
DDL INCLUDE MAPPED OBJNAME oggl.west.*
DDLOPTIONS ADDTRANDATA
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
StatOptions ResetReportStats
ReportCount Every 10 Records
GetUpdateBefores
TranLogOptions ExcludeUser C##OGG_ADMIN
SOURCECATALOG oggl
Table oggl.west.account GetBeforeCols(on update all);
Table oggl.west.branch GetBeforeCols(on update all);
Table oggl.west.customer GetBeforeCols(on update all);
```

The lines in bold are the lines you have to add/modify. Save the file. Edit the Replicat parameter file:

```
GGSCI ([Host]) 5> edit params rewest
```

```
Replicat rewest
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV (ORACLE_SID='CDB12c')
AssumeTargetDefs
DiscardFile ./dirrpt/rewest.dsc, Purge
UserIdAlias oggl_repl
DDLError DEFAULT IGNORE
getupdates
getinserts
```

```

getdeletes
Map oggdwn1.east.account, Target ogg1.west.account,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (delta_resolution_method, usedelta, cols(account_balance)),
 (max_resolution_method, usemax(update_ts),
 cols(account_type,account_number,account_opened,
 account_branch_id,update_ts)),
 (default overwrite));
Map oggdwn1.east.branch, Target ogg1.west.branch,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (max_resolution_method, usemax(update_ts),
 cols(branch_street_type,branch_street_name,
 branch_street_number,branch_level,branch_city,branch_state,
 branch_country,branch_size,branch_opened,zip_code,
 branch_name,update_ts)),
 (default overwrite));
Map oggdwn1.east.customer, Target ogg1.west.customer,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (max_resolution_method, usemax(update_ts),
 cols(name,middle_name,surname,full_name,dob,nationality,
 email,update_ts)),
 (default overwrite));

```

You must add the lines in bold to the Replicat definition. The ACCOUNT table is the only one that contains a numeric column (ACCOUNT\_BALANCE), which could incur the net change value data conflict. The MAP statement for the ACCOUNT table specifies the Delta\_Resolution\_Method for the ACCOUNT\_BALANCE column. The conflicts for all other columns in ACCOUNT and all columns in BRANCH and CUSTOMER are resolved by using the Max\_Resolution\_Method, which uses the update time stamp to choose the “winner” value (the most recently inserted [highest timestamp] value will be selected for data convergence).

Save the file and reset the capture for the Extract; start the Extract and the Replicat groups and exit GGSCI:

```

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > alter exwest begin now
EXTRACT altered.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start exwest
Sending START request to MANAGER ...
EXTRACT EXWEST starting
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start rwest
Sending START request to MANAGER ...

```

```
REPLICAT REWEST starting
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING DPWEST 00:00:00 00:00:02
EXTRACT RUNNING EXWEST 00:02:07 00:00:09
REPLICAT RUNNING REWEST 00:00:00 00:00:04
GGSCI ([Host]) 9>
```

4. You must apply the same modifications to the Oracle GoldenGate instance that connects to the DWNSTR/oggdwn1/east database. Click the OGG\_EAST window, stop the exeast Extract and the reeast Replicat groups, and edit the parameter files for the Extract and the Replicat:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING DPEAST 00:00:00 00:00:07
EXTRACT RUNNING EXEAST 00:00:00 00:00:09
REPLICAT RUNNING REEAST 00:00:00 00:00:02
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > stop exeast
Sending STOP request to EXTRACT EXEAST ...
Request processed.
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > stop reeast
Sending STOP request to REPLICAT REEAST ...
Request processed.
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > edit params
xeast
```

```

Extract exeast
SETENV (ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV (ORACLE_SID = "DWNSTR")
ExtTrail ./dirdat/ee
UserIdAlias ogg_dwnadm
DDL INCLUDE MAPPED OBJNAME oggdwn1.east.*
DDLOPTIONS ADDTRANDATA
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
StatOptions ResetReportStats
ReportCount Every 10 Records
GetUpdateBefores
TranLogOptions ExcludeUser C##OGG_DWNADM
SOURCECATALOG oggdwn1
Table east.account GetBeforeCols(on update all);
Table east.branch GetBeforeCols(on update all);
Table east.customer GetBeforeCols(on update all);

```

The lines in bold are the lines you have to add/modify. Save the file. Edit the Replicat parameter file:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > edit params
reeast
```

```

Replicat reeast
SETENV (ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV (ORACLE_SID = "DWNSTR")
DiscardFile ./dirrpt/reeast.dsc, Purge
UserIDAlias ogg_dwnrepl
DDLERROR DEFAULT IGNORE
AssumeTargetDefs
GetUpdates
GetInserts
GetDeletes
Map ogg1.west.account, Target oggdwn1.east.account,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (delta_resolution_method, usedelta, cols(account_balance)),
 (max_resolution_method, usemax(update_ts),
 cols(account_type,account_number,account_opened,
 account_branch_id,update_ts)),
 (default overwrite));
Map ogg1.west.branch, Target oggdwn1.east.branch,
 comparecols(on update all)

```

```

 resolveconflict(updaterowexists,
 (max_resolution_method, usemax(update_ts),
 cols(branch_street_type,branch_street_name,
 branch_street_number,branch_level,branch_city,branch_state,
 branch_country,branch_size,branch_opened,zip_code,
 branch_name,update_ts)),
 (default overwrite));
Map oggl.west.customer, Target oggdwnl.east.customer,
comparecols(on update all)
resolveconflict(updaterowexists,
(max_resolution_method, usemax(update_ts),
cols(name,middle_name,surname,full_name,dob,nationality,
email,update_ts)),
(default overwrite));

```

Save the file and reset the capture for the Extract; start the Extract and the Replicat:

```

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > alter exeast
begin now
EXTRACT altered.
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > start exeast
Sending START request to MANAGER ...
EXTRACT EXEAST starting
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > start reeast
Sending START request to MANAGER ...
REPLICAT REEAST starting
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING DPEAST 00:00:00 00:00:00
EXTRACT RUNNING EXEAST 00:00:14 00:00:07
REPLICAT RUNNING REEAST 00:00:00 00:00:02

```

- In order to simulate network latency and a lag in replication, stop the two Data Pumps. Click the OGG\_WEST window and stop the dpwest pump:

```

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop dpwest
Sending STOP request to EXTRACT DPWEST ...
Request processed.

```

- Stop the dpeast Data Pump also. Select the OGG\_EAST window, where GGSCI runs connected to the east schema and stop the pump:

```

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > stop dpeast
Sending STOP request to EXTRACT DPEAST ...
Request processed.

```

7. Click the SQL\_WEST window, where sqlplus runs connected to the west schema, and verify the account balance for account ID 299:

```
SQL> select acct_id, cust_id, account_balance from account where
acct_id=299;
```

ACCT_ID	CUST_ID	ACCOUNT_BALANCE
299	99	7192

8. Click the SQL\_EAST window, where sqlplus runs connected to the east schema, and verify the account balance for account ID 299:

```
SQL> select acct_id, cust_id, account_balance from account where
acct_id=299;
```

ACCT_ID	CUST_ID	ACCOUNT_BALANCE
299	99	7192

9. Click the SQL\_WEST window and enter the update that deducts 30 dollars from the account (this simulates Mr. Keen making his purchase from the office).

```
SQL> UPDATE account set account_balance = account_balance - 30
where acct_id = 299 and account_balance = 7192;
1 row updated.
SQL> commit;
```

10. Click the SQL\_EAST window and enter the update that deducts 50 dollars from the account (this simulates Mrs. Keen making her purchase using her husband's account).

```
SQL> UPDATE account set account_balance = account_balance - 50
where acct_id=299 and account_balance = 7192;
1 row updated.
SQL > commit;
```

11. Restart the two Data Pumps and, in both databases, select the balance for account 299. Select the OGG\_WEST window, where GGSCI runs connected to the west schema and start DPWEST:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start dpwest
Sending START request to EXTRACT DPWEST ...
EXTRACT DPWEST starting.
```

12. Select the OGG\_EAST window, where GGSCI runs connected to the east schema and start DPEAST:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > start dpeast
Sending START request to EXTRACT DPEAST ...
EXTRACT DPEAST starting.
```

13. Select the SQL\_WEST window, where sqlplus runs connected to the west schema, and verify the account balance for account 299:

```
SQL> select acct_id, cust_id, account_balance from account where
acct_id=299;
```

ACCT_ID	CUST_ID	ACCOUNT_BALANCE
299	99	7112

14. Select the SQL\_EAST window, where sqlplus runs connected to the east schema, and verify the account balance for account 299:

```
SQL> select acct_id, cust_id, account_balance from account where
acct_id=299;
```

ACCT_ID	CUST_ID	ACCOUNT_BALANCE
299	99	7112

15. In both databases, the final balance reflects the real situation. The balance is now \$7,112, which means that both the 30 dollars spent by Mr. Keen and the 50 dollars spent by Mrs. Keen have been taken into account. The USEDDELTA data conflict policy worked as expected. To verify the proper working of CDR, select the windows where GGSCI runs connected to the west database and request for statistics on the Replicat group, specifying the ReportCDR qualifier to the stats command:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stats rewest
reportcdr
Sending STATS request to REPLICAT REWEST ...
Start of Statistics at 2015-03-19 21:58:15.
Replicating from EAST.ACOUNT to WEST.ACOUNT:

*** Total statistics since 2015-03-19 20:31:50 ***
Total inserts 0.00
Total updates 1.00
Total deletes 0.00
Total discards 0.00
Total operations 1.00
Total CDR conflicts 1.00
CDR resolutions succeeded 1.00
CDR UPDATEROWEXISTS conflicts 1.00
```

The statistics show that one conflict of type UPDATEROWEXISTS was detected and one CDR resolution was applied successfully.

16. Select the window where GGSCI runs connected to the east database and request for statistics on the Replicat group, to verify that the same applies to the east database:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > stats reeast
reportcdr
Sending STATS request to REPLICAT REEST ...
Start of Statistics at 2015-03-19 22:05:18.
Replicating from WEST.ACCOUNT to EAST.ACCOUNT:

*** Total statistics since 2015-03-19 20:44:21 ***
Total inserts 0.00
Total updates 1.00
Total deletes 0.00
Total discards 0.00
Total operations 1.00
Total CDR conflicts 1.00
CDR resolutions succeeded 1.00
CDR UPDATEROWEXISTS conflicts 1.00
```

The statistics also show that the Oracle GoldenGate instance that is connected to the east database detected a conflict and successfully applied a conflict resolution.

17. The Replicat groups that you defined include more mapping rules. You can test at will what happens when you introduce data conflicts. Stop the two Data Pumps, and arbitrarily introduce data conflicts and verify the results, after you restart the Data Pumps to achieve data convergence. In particular, evaluate what happens when you update a row in one of the three tables across the two databases, and you explicitly set the time stamp as part of the update, and what happens when you modify the row through an update command without setting the time stamp. (**Hint:** The UseMax conflict resolution policy relies on the highest time stamp. If your application updates rows without setting the update time stamp, what happens to the UseMax policy?)

This completes Practice 9-3.

This completes Practice 9. Stop here. Leave all the Extract, Data Pump, and Replicat groups running for the next practice.

# **Practices for Lesson 10: Conflict Detection and Resolution: Custom Techniques**

**Chapter 10**

## Practices for Lesson 10: Overview

---

### Practices Overview

In these practices, you will implement a data conflict avoidance technique based on non-overlapping sequences for insert statements and a much more sophisticated technique that is based on the powerful combination of Oracle GoldenGate MAP and SQLEXEC commands, which allows for quantitative resolution of data conflict.

**Helper scripts provided in `/home/oracle/labs/Practice10/configs`:**

Catch-up/Clean-up script	Purpose
<code>catch-up-prac-10_1.sh</code>	Executes steps for Practice 10-1
<code>catch-up-prac-10_2.sh</code>	Executes steps for Practice 10-2
<code>clean-up.sh</code>	Complete clean-up script. It resets the environment as it was before starting Practice 10.

**Note:** You should run the `clean-up.sh` script at the end of Practice 10-2, before moving on to Practice 11.

## Practice 10-1: Data Conflict Avoidance Techniques

### Overview

In this practice, you explore the data conflict avoidance technique for inserts, which is based on alternate and non-overlapping sequences.

### Assumptions

The active-active replication framework that was built during the practices for Lesson 9, “Bidirectional Replication: Two-Node Configuration” is operational. The Oracle GoldenGate instances are running, including the manager processes and the Extract, Data Pumps, and Replicat groups.

### Tasks

#### Preventing DDL replication

1. Click the OGG\_WEST window and edit the parameter file for the EXWEST Extract group, commenting out the DDL replication directives:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
exwest
```

```
Extract exwest
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "CDB12c")
ExtTrail ./dirdat/ew
UserIdAlias ogg_admin
--DDL INCLUDE MAPPED OBJNAME ogg1.west.*
--DDLOPTS ADDTRANDATA
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
StatOptions ResetReportStats
ReportCount Every 10 Records
GetUpdateBefores
TranLogOptions ExcludeUser C##OGG_ADMIN
SOURCECATALOG ogg1
Table ogg1.west.account GetBeforeCols(on update all);
Table ogg1.west.branch GetBeforeCols(on update all);
Table ogg1.west.customer GetBeforeCols(on update all);
```

2. Stop the EXWEST Extract group and restart it, to force the rereading of the configuration parameters:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop extract
exwest
Sending STOP request to EXTRACT EXWEST ...
Request processed.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start extract
exwest
Sending START request to MANAGER ...
EXTRACT EXWEST starting
```

3. Click the OGG\_EAST window and repeat the same steps for the second Oracle GoldenGate instance, commenting out the DDL replication directives:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > edit params
exeast
```

```
Extract exeast
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "DWNSTR")
ExtTrail ./dirdat/ee
UserIdAlias ogg_dwnadm
--DDL INCLUDE MAPPED OBJNAME oggdwn1.east.*
--DDLOPTIONS ADDTRANDATA
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
StatOptions ResetReportStats
ReportCount Every 10 Records
GetUpdateBefores
TranLogOptions ExcludeUser C##OGG_DWNADM
SOURCECATALOG oggdwn1
Table east.account GetBeforeCols(on update all);
Table east.branch GetBeforeCols(on update all);
Table east.customer GetBeforeCols(on update all);
```

4. Stop the EXEAST Extract group and restart it, to force the rereading of the configuration parameters. Then, exit GGSCI:

```
([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > stop extract exeast
Sending STOP request to EXTRACT EXEAST ...
Request processed.
([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > start extract exeast
Sending START request to MANAGER ...
EXTRACT EXEAST starting
```

5. Click the SQL\_WEST window and exit sqlplus. At the OS prompt, change the directory to ~/labs/Practice10/configs and relaunch sqlplus, connecting to the ogg1 PDB, west schema. Use the "set sqlprompt" sqlplus command to set the prompt according to the database you are connected to:

```
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
[OS prompt]$ cd ~/labs/Practice10/configs
[OS prompt]$ sqlplus west/Welcomel@ogg1
SQL*Plus: Release 12.1.0.2.0 Production on Sat Mar 21 11:16:28
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Wed Mar 18 2015 19:33:04 +11:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> set sqlprompt "WEST> "
WEST>
```

6. Click the SQL\_EAST window and exit sqlplus. At the OS prompt, change the directory to ~/labs/Practice10/configs and relaunch sqlplus, connecting to the oggdwn1 PDB, east schema. Use the "set sqlprompt" sqlplus command to set the prompt according to the database you are connected to:

```
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
[OS prompt]$ cd ~/labs/Practice10/configs
[OS prompt]$ sqlplus east/Welcomel@oggdwn1
SQL*Plus: Release 12.1.0.2.0 Production on Sat Mar 21 11:16:28
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Wed Mar 18 2015 19:33:04 +11:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> set sqlprompt "EAST> "
EAST>
```

7. Open an additional shell window and navigate to the ~/labs/Practice10/configs directory. Use the cat OS command to display the content of the two files sequences\_west.sql and sequences\_east.sql. The same sequences are created for the east and west databases, but the sequence generation algorithm is different. This arrangement ensures that the numbers assigned to the primary key fields are different and not overlapping across the two databases:

```
[OS prompt]$ cd ~/labs/Practice10/configs
[OS prompt]$ cat sequences_west.sql
create sequence accnt_seq start with 401 increment by 2 cache
100;
create sequence branch_seq start with 401 increment by 2 cache
100;
create sequence custmr_seq start with 401 increment by 2 cache
100;
create sequence accnt_num_seq start with 100001 increment by 2
cache 100;
create sequence item_seq start with 11 increment by 2 cache 100;

[OS prompt]$ cat sequences_east.sql
create sequence accnt_seq start with 400 increment by 2 cache
100;
create sequence branch_seq start with 400 increment by 2 cache
100;
create sequence custmr_seq start with 400 increment by 2 cache
100;
create sequence accnt_num_seq start with 100000 increment by 2
cache 100;
create sequence item_seq start with 10 increment by 2 cache 100;
[OS prompt]$
```

8. Click the SQL\_WEST window and execute the sequences\_west.sql script:

```
WEST> @sequences_west.sql
Sequence created.
Sequence created.
Sequence created.
Sequence created.
Sequence created.
```

9. Click the SQL\_EAST window and execute the sequences\_east.sql script:

```
EAST> @sequences_east.sql
Sequence created.
Sequence created.
Sequence created.
Sequence created.
Sequence created.
EAST>
```

10. In the SQL\_WEST window, where sqlplus is connected to the OGG1/west database, insert a new account row, using the sequence acct\_seq to obtain ACCT\_ID, which is part of the primary key. Use CUST\_ID 25, corresponding to customer Lee Barnard, as the customer for the account:

```
WEST> Insert into ACCOUNT
(ACCT_ID,CUST_ID,ACCOUNT_TYPE,ACCOUNT_NUMBER,ACCOUNT_BALANCE,
ACCOUNT_OPENED,ACCOUNT_BRANCH_ID,UPDATE_TS) values
(ACCNT_SEQ.nextval,25,'CHECKING',ACCNT_NUM_SEQ.nextval,2000,
SYSDATE , '2',SYSTIMESTAMP);
1 row created.
WEST> commit;
```

11. In the SQL\_EAST window, where sqlplus is connected to the OGGDWN1/east database, insert the same row:

```
EAST> Insert into ACCOUNT
(ACCT_ID,CUST_ID,ACCOUNT_TYPE,ACCOUNT_NUMBER,ACCOUNT_BALANCE,
ACCOUNT_OPENED,ACCOUNT_BRANCH_ID,UPDATE_TS) values
(ACCNT_SEQ.nextval,25,'CHECKING',ACCNT_NUM_SEQ.nextval,2000,
SYSDATE , '2',SYSTIMESTAMP);
1 row created.
EAST> commit;
```

12. In both windows connected to the east and the west database, submit the query that selects the ACCT\_ID assigned by the sequence:

```
WEST> set linesize 132
WEST> column ACCT_ID format 9999
WEST> select acct_id, update_ts from account where cust_id=25
order by 2;

ACCT_ID UPDATE_TS

25 19-MAR-15 11.46.25.208533 PM
225 19-MAR-15 11.46.42.536684 PM
401 21-MAR-15 11.30.58.946309 AM
400 21-MAR-15 11.32.05.853902 AM
```

13. Your values and time stamp may vary, but the point is that the last two insertions generated ACCT\_ID 400 and 401. The same values inserted from the two databases generated a logical duplicate, but there was no conflict. The sequences generate alternate and non-overlapping values for the columns used as primary key. This illustrates the basic principle of conflict avoidance for insertions, which is based on non-overlapping values generated by sequences.

To ensure insert data conflict avoidance, the applications using active-active configurations must always use the sequences created to support the primary key generation. This is why it is usually difficult to run applications that were not designed from the ground up for active-active replication, in multi-master replicated environments. Third-party packaged applications, which are sold under proprietary software licenses, that do not include the source code and are designed for a single instance deployment are in general difficult to port to multi-master environments.

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

## Practice 10-2: Resolving Data Conflicts by Using Quantitative Resolution Methods

### Overview

In this practice, you use MAP and SQLEXEC to implement a more advanced data conflict detection and resolution technique that is commonly used to ensure convergence where values or quantities can be updated on both databases that are participating in the active-active replication topology.

### Assumptions

- The active-active replication environment built during the practices for Lesson 9, “Bidirectional Replication: Two-Node Configuration” is running.
- You have successfully completed Practice 10-1.

### Tasks

1. Click the SQL\_WEST window, where sqlplus is connected to the OGG1/west schema. Run the item.sql and inventory.sql scripts to create and populate the ITEM and INVENTORY tables.

```
WEST> @item.sql
Table created.
1 row added.
... Many lines omitted for clarity ...
SQL> @inventory.sql
Table created.
1 row added.
... Many lines omitted for clarity ...
```

2. Click the SQL\_EAST window, where sqlplus is connected to the OGGDWN1/east schema, and create the same objects:

```
EAST> @item.sql
Table created.
1 row added.
... Many lines omitted for clarity ...
SQL> @inventory.sql
Table created.
1 row added.
... Many lines omitted for clarity ...
```

3. Click the OGG\_WEST window, where GGSCI is connected to the WEST Oracle GoldenGate instance. Add supplemental logging for the two tables that you just created in the west schema:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add trandata
ogg1.west.inventory, cols(ITEM_QUANTITY)
Logging of supplemental redo data enabled for table
OGG1.WEST.INVENTORY.

TRANDATA for scheduling columns has been added on table
'OGG1.WEST.INVENTORY'.

GGSCI ([Host]) 3> add trandata ogg1.west.item
Logging of supplemental redo data enabled for table
OGG1.WEST.ITEM.

TRANDATA for scheduling columns has been added on table
'OGG1.WEST.ITEM'.
```

4. Edit the parameter file for the exwest Extract group. Uncomment the DDL replication directives and add the Table command for the two newly created tables, ITEM and INVENTORY:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
exwest
```

```
Extract exwest
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "CDB12c")
ExtTrail ./dirdat/ew
UserIdAlias ogg_admin
DDL INCLUDE MAPPED OBJNAME ogg1.west.*
DDLOPTIONS ADDTRANDATA
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
StatOptions ResetReportStats
ReportCount Every 10 Records
GetUpdateBefores
TranLogOptions ExcludeUser C##OGG_ADMIN
SOURCECATALOG ogg1
Table ogg1.west.account GetBeforeCols(on update all);
Table ogg1.west.branch GetBeforeCols(on update all);
Table ogg1.west.customer GetBeforeCols(on update all);
Table ogg1.west.item getbeforecols (on update all);
Table ogg1.west.inventory getbeforecols (on update all);
```

5. Save the file. Stop the `exwest` Extract group and restart it, to force it to read the new parameters.

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop exwest
Sending STOP request to EXTRACT EXWEST ...
Request processed.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start exwest
Sending START request to MANAGER ...
EXTRACT EXWEST starting
```

6. You do not need to modify the `dpwest` Data Pump, because it uses a wildcard to allow replication for all tables residing in the `west` schema. Instead, edit the parameter file for the `rewest` Replicat group, and add the mapping rules for `ITEM` and `INVENTORY`. `ITEM` is a small and mostly static table for which you do not care to resolve data conflicts. `INVENTORY`, on the other hand, is a table that carries item quantities, whose conflicts could have serious business consequences. The quantitative conflict resolution formula is implemented by using `SQLEXEC` as a subcommand of `MAP`. `SQLEXEC` is used to query the before image and assess if the current data is generating a conflict.

```
GGSCI ([Host]) 4> edit params rewest
```

```
Replicat rewest
SETENV (ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV (ORACLE_SID='CDB12c')
AssumeTargetDefs
DiscardFile ./dirrpt/rewest.dsc, Purge
UserIdAlias ogg1_repl
DDLError DEFAULT IGNORE
getupdates
getinserts
getdeletes
Map oggdwn1.east.account, Target ogg1.west.account,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (delta_resolution_method, usedelta, cols(account_balance)),
 (max_resolution_method, usemax(update_ts),
 cols(account_type,account_number,account_opened,
 account_branch_id,update_ts)),
 (default overwrite));
Map oggdwn1.east.branch, Target ogg1.west.branch,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (max_resolution_method, usemax(update_ts),
 cols(branch_street_type,branch_street_name,
 branch_street_number,branch_level,branch_city,branch_state,
 branch_country,branch_size,branch_opened,zip_code,
 branch_name,update_ts)),
```

```

 (default overwrite));
Map oggdwn1.east.customer, Target oggl.west.customer,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (max_resolution_method, usemax(update_ts)),
 cols(name,middle_name,surname,full_name,dob,nationality,
 email,update_ts)),
 (default overwrite));
MAP oggdwn1.east.item, Target oggl.west.item;
MAP oggdwn1.east.inventory, Target oggl.west.inventory,
SQLLEXEC (id qty, query "SELECT item_quantity from
west.inventory where item_id = :v_item_id AND
item_size= :v_item_size AND item_color = :v_item_color",
params (v_item_id = item_id, v_item_size = item_size,
v_item_color = item_color),
COLMAP (item_id = item_id, item_size = item_size,
item_color = item_color, item_quantity =
@if(qty.item_quantity <> BEFORE.item_quantity,
@compute(qty.item_quantity - (before.item_quantity -
item_quantity)), item_quantity));

```

- Save the file. Stop and restart the Replicat group to force it to reread the parameters.

```

GGSCI ([Host]) 5> stop rewest
Sending STOP request to REPLICAT REWEST ...
Request processed.
GGSCI ([Host]) 6> start rewest
Sending START request to MANAGER ...
REPLICAT REWEST starting
GGSCI ([Host]) 7> info all

Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING DPWEST 00:00:00 00:00:04
EXTRACT RUNNING EXWEST 00:00:00 00:00:05
REPLICAT RUNNING REWEST 00:00:00 00:00:06
GGSCI ([Host]) 8>

```

8. Click the OGG\_EAST window and repeat the same steps for the second Oracle GoldenGate instance. Start by adding supplemental logging for the INVENTORY and ITEM tables in the east schema:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > add trandata
oggdwn1.east.inventory, cols(ITEM_QUANTITY)
Logging of supplemental redo data enabled for table
OGGDWN1.EAST.INVENTORY.

TRANDATA for scheduling columns has been added on table
'OGGDWN1.EAST.INVENTORY'.

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > add trandata
oggdwn1.east.item
Logging of supplemental redo data enabled for table
OGGDWN1.EAST.ITEM.

TRANDATA for scheduling columns has been added on table
'OGGDWN1.EAST.ITEM'.
```

9. Edit the parameter file for the exeast Extract group, remove the comments before the DDL commands, and add the item and inventory tables:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > edit params
exeast

Extract exeast
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "DWNSTR")
ExtTrail ./dirdat/ee
UserIdAlias ogg_dwnadm
DDL INCLUDE MAPPED OBJNAME oggdwn1.east.*
DDLOPTIONS ADDTRANDATA
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
StatOptions ResetReportStats
ReportCount Every 10 Records
GetUpdateBefores
TranLogOptions ExcludeUser C##OGG_DWNADM
SOURCECATALOG oggdwn1
Table east.account GetBeforeCols(on update all);
Table east.branch GetBeforeCols(on update all);
Table east.customer GetBeforeCols(on update all);
Table east.item GetBeforeCols(on update all);
Table east.inventory GetBeforeCols(on update all);
```

10. Save the file. Stop and restart the Extract group to force it to reread the parameters.

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > stop exeast
Sending STOP request to EXTRACT EXEAST ...
Request processed.
```

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > start exeast
Sending START request to MANAGER ...
EXTRACT EXEAST starting
```

11. Edit the parameter file for the `rewest` Replicat group and implement the quantitative conflict resolution formula using `MAP` and `SQLEXEC`:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > edit params
reeast
```

```
Replicat reeast
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "DWNSTR")
AssumeTargetDefs
DiscardFile ./dirrpt/reeast.dsc, Purge
UserIDAlias ogg_dwnrep1
DDLError DEFAULT IGNORE
getupdates
getinserts
getdeletes
Map ogg1.west.account, Target oggdwn1.east.account,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (delta_resolution_method, usedelta, cols(account_balance)),
 (max_resolution_method, usemax(update_ts),
 cols(account_type,account_number,account_opened,
 account_branch_id,update_ts)),
 (default overwrite));
Map ogg1.west.branch, Target oggdwn1.east.branch,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (max_resolution_method, usemax(update_ts),
 cols(branch_street_type,branch_street_name,
 branch_street_number,branch_level,branch_city,
 branch_state,branch_country,branch_size,branch_opened,
 zip_code,branch_name,update_ts)),
 (default overwrite));
Map ogg1.west.customer, Target oggdwn1.east.customer,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (max_resolution_method, usemax(update_ts),
 cols(name,middle_name,surname,full_name,dob,nationality,
 email,update_ts)),
 (default overwrite));
MAP ogg1.west.item, Target oggdwn1.east.item;
```

```

MAP ogg1.west.inventory, Target oggdwn1.east.inventory,
 SQLEXEC (id qty, query "SELECT item_quantity from
 east.inventory where item_id = :v_item_id AND item_size =
 :v_item_size AND item_color = :v_item_color",
 params (v_item_id = item_id, v_item_size = item_size,
 v_item_color = item_color)),
 COLMAP (item_id = item_id, item_size = item_size,
 item_color = item_color, item_quantity =
 @if(qty.item_quantity <> BEFORE.item_quantity,
 @compute(qty.item_quantity - (before.item_quantity -
 item_quantity)), item_quantity));

```

12. Save the file. Stop and restart the Replicat group to force it to reread the parameters:

```

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > stop reeast
Sending STOP request to REPLICAT REEST ...
Request processed.

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > start reeast
Sending START request to MANAGER ...
REPLICAT REEST starting

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING DPEAST 00:00:00 00:00:02
EXTRACT RUNNING EXEAST 00:00:00 00:00:06
REPLICAT RUNNING REEST 00:00:00 00:00:03

```

13. The infrastructure is now in place to observe SQLEXEC in action. The usual trick to simulate network latency is to stop both Data Pumps. Stop dpeast; then click the OGG\_WEST window and also stop dpwest:

```

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > stop dpeast
Sending STOP request to EXTRACT DPEAST ...
Request processed.

```

The OGG\_WEST window:

```

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop dpwest
Sending STOP request to EXTRACT DPWEST ...
Request processed.

```

14. Click the SQL\_WEST window and submit the query below:

```

WEST> select * from inventory where item_id = 1 and
 item_size='LARGE' and item_color = 'orange';
 ITEM_ID ITEM_SIZE ITEM_COLOR ITEM_QUANTITY

 1 LARGE orange 42

```

15. Select the window where sqlplus is connected to the east database, submit the same query, and verify that the output is the same:

```
EAST> select * from inventory where item_id = 1 and
item_size='LARGE' and item_color = 'orange';
 ITEM_ID ITEM_SIZE ITEM_COLOR ITEM_QUANTITY
----- ----- -----
 1 LARGE orange 42
```

16. Click the SQL\_WEST window, where sqlplus is connected to the west schema, and update the INVENTORY table:

```
WEST> update inventory set item_quantity = item_quantity - 12
where item_id = 1 and item_size='LARGE' and item_color =
'orange' and item_quantity = 42;
1 row updated.
WEST> commit;
```

17. Click the SQL\_EAST window, where sqlplus is connected to the east schema, and update the same row in the INVENTORY table; this time, set a different quantity:

```
EAST> update inventory set item_quantity = item_quantity - 8
where item_id = 1 and item_size='LARGE' and item_color =
'orange' and item_quantity = 42;
1 row updated.
EAST> commit;
```

18. Restart the two Data Pumps, dpwest and dpeast. Click the OGG\_WEST window and start dpwest:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start dpwest
Sending START request to MANAGER ...
EXTRACT DPWEST starting
```

19. Click the OGG\_EAST window and start the dpeast Data Pump:

```
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > start dpeast
Sending START request to MANAGER ...
EXTRACT DPEAST starting
```

20. Enter the GGSCI command view report reeast to verify that the mapping worked:

```
GGSCI ([Host]) 2> view report reeast
```

21. Toward the end of the report, Replicat shows a MAP resolved entry:

```
2015-03-21 12:44:25 INFO OGG-06505 MAP resolved (entry
ogg1.west.inventory) : MAP "OGG1"."WEST"
 ."INVENTORY", Target oggdwn1.east.inventory, SQLEXEC (id qty,
query "SELECT item_quantity from east
.inventory where item_id = :v_item_id AND item_size =
:v_item_size AND item_color = :v_item_color",
 params (v_item_id = item_id, v_item_size = item_size,
v_item_color = item_color)), COLMAP (item_id
 = item_id, item_size = item_size, item_color = item_color,
item_quantity = @if(qty.item_quantity <
```

```
> BEFORE.item_quantity, @compute(qty.item_quantity -
(before.item_quantity - item_quantity)), item_
quantity) .
2015-03-21 12:44:27 INFO OGG-06510 Using the following key
columns for target table OGGDWN1.EA
ST.INVENTORY: ITEM_ID, ITEM_SIZE, ITEM_COLOR.
```

22. Select the SQL\_WEST window. Enter the query below, and verify the quantity balance after the same row was updated concurrently while the Data Pumps were not operational:

WEST> select * from inventory where item_id = 1 and item_size='LARGE' and item_color = 'orange';
ITEM_ID ITEM_SIZE ITEM_COLOR ITEM_QUANTITY
-----
1 LARGE orange 22

23. Select the SQL\_EAST window, where sqlplus is running connected to the east database. Enter the same query and verify the results:

EAST>select * from inventory where item_id = 1 and item_size='LARGE' and item_color = 'orange';
ITEM_ID ITEM_SIZE ITEM_COLOR ITEM_QUANTITY
-----
1 LARGE orange 22

In both databases, the quantity column reflects the correct balance. The update performed on the east database subtracted 8 items and the update performed on the west database subtracted 12 items, for a total of 20 fewer items. The initial quantity was 42 items, and the current quantity is 22 items. The quantitative conflict resolution policy, which is implemented using SQLEXEC, worked.

24. To clean up your environment, you can either run the `clean-up.sh` script located in the `/home/oracle/labs/Practice10/configs` directory or click the OGG\_WEST window, stop and delete all replication groups, and then unregister the EXWEST Extract from the OGG1 database:

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB\$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt
MANAGER RUNNING
EXTRACT RUNNING DPWEST 00:00:00 00:00:03
EXTRACT RUNNING EXWEST 00:00:00 00:00:08
REPLICAT RUNNING REWEST 00:00:00 00:00:05
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB\$ROOT) >
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB\$ROOT) > stop er *
Sending STOP request to EXTRACT DPWEST ...
Request processed.
Sending STOP request to EXTRACT EXWEST ...

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

```

Request processed.
Sending STOP request to REPLICAT REWEST ...
Request processed.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT STOPPED DPWEST 00:00:00 00:00:08
EXTRACT STOPPED EXWEST 00:00:00 00:00:08
REPLICAT STOPPED REWEST 00:00:00 00:00:07

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > delete er *
Are you sure you want to delete all groups? y
Deleted EXTRACT DPWEST.
Deleted EXTRACT EXWEST.
Deleted REPLICAT REWEST.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > unregister
extract EXWEST database
Successfully unregistered EXTRACT EXWEST from database.

```

25. Select the OGG\_EAST window, and stop and delete all replication groups. Then unregister the EXEAST Extract from the OGGDWN1 database:

```

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING DPEAST 00:00:00 00:00:03
EXTRACT RUNNING EXEAST 00:00:00 00:00:08
REPLICAT RUNNING REEAST 00:00:00 00:00:05

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > stop er *
Sending STOP request to EXTRACT DPEAST ...
Request processed.

Sending STOP request to EXTRACT EXEAST ...
Request processed.

Sending STOP request to REPLICAT REEAST ...
Request processed.
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING

```

```
EXTRACT STOPPED DPEAST 00:00:00 00:00:08
EXTRACT STOPPED EXEAST 00:00:00 00:00:08
REPLICAT STOPPED REEAST 00:00:00 00:00:07

GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > delete er *
Are you sure you want to delete all groups? y
Deleted EXTRACT DPEAST.
Deleted EXTRACT EXEAST.
Deleted REPLICAT REEAST.
GGSCI ([HOST] as C##OGG_DWNADM@DWNSTR/CDB$ROOT) > unregister
extract EXEAST database
Successfully unregistered EXTRACT EXEAST from database.
```

This completes Practice 10-2.

You can exit from **GGSCI** and **sqlplus** on all windows and terminate them as you won't use any of these windows in Practice 11.

This completes Practice 10. Stop here.



## **Practices for Lesson 11: Multi-Master Replication Topology: Three-Node Configuration**

**Chapter 11**

## Practices for Lesson 11: Overview

---

### Practices Overview

In these practices, you will configure a multi-master replication environment based on three active nodes. It is important to be disciplined about the naming convention used for the various replication objects. Oracle GoldenGate best practices dictate that you do not use names for Extract or Replicat groups that end with numbers. You will use A, B, and C to denote the nodes `ogg_node1`, `OGG_NODE2`, and `OGG_NODE3`, respectively. The Extract group on the `ogg_node1` node will be called `exdba`, writing to the trail file `./dirdat/aa`. On the `OGG_NODE2` node, the Extract group will be called `exdbb`, writing to the trail file `./dirdat/bb`.

Each Oracle GoldenGate instance in these examples defines two Data Pumps. On the `ogg_node1` node, their names will be `dppbb` and `dppdbc`. The first two letters identify the Extract as a Data Pump, and the remaining three letters identify the destination (`dppbb` means the Extract pumps data to the `OGG_NODE2` instance; `dppdbc` means the Extract pumps data to the `OGG_NODE3` instance). The remote trails where the two Data Pumps write data use two letters, where the convention is source-destination. The remote trail where the `dppbb` Data Pump writes its data will be called `./dirdat/ab` ("a" means from node A, `ogg_node1` and "b" means node B, `OGG_NODE2`). The same convention applies to the Replicat groups. Node `ogg_node1` will define two Replicat groups—`reabb` and `reabc`—where the two initial letters indicate that they are Replicats, the third letter denotes the destination node ("a" for `OGG_NODE1`, "b" for `OGG_NODE2`, and "c" for `OGG_NODE3`), and the remaining two letters show the origin of the data (`reabb` means data coming from `OGG_NODE2` and `reabc` means data coming from `OGG_NODE3`).

The table below lists all names used for the multi-master replication topology that you will configure in the practice:

Node	Extract	Local Trail	Remote Nodes				
			Node Name	Data Pump	Remote Trail	Replicat	Replicat Trail
<code>OGG_NODE1</code>	<code>exdba</code>	<code>./dirdat/aa</code>	<code>OGG_NODE2</code>	<code>dppbb</code>	<code>./dirdat/ab</code>	<code>reabb</code>	<code>./dirdat/ba</code>
			<code>OGG_NODE3</code>	<code>dppdbc</code>	<code>./dirdat/ac</code>	<code>reabc</code>	<code>./dirdat/ca</code>
<code>OGG_NODE2</code>	<code>exdbb</code>	<code>./dirdat/bb</code>	<code>OGG_NODE1</code>	<code>dppba</code>	<code>./dirdat/ba</code>	<code>rebba</code>	<code>./dirdat/ab</code>
			<code>OGG_NODE3</code>	<code>dppbc</code>	<code>./dirdat/bc</code>	<code>rebcb</code>	<code>./dirdat/cb</code>
<code>OGG_NODE3</code>	<code>exdbc</code>	<code>./dirdat/cc</code>	<code>OGG_NODE1</code>	<code>dppca</code>	<code>./dirdat/ca</code>	<code>recba</code>	<code>./dirdat/ac</code>
			<code>OGG_NODE2</code>	<code>dppbb</code>	<code>./dirdat/cb</code>	<code>recbb</code>	<code>./dirdat/bc</code>

Helper scripts provided in `/home/oracle/labs/Practice11/configs`:

Catch-up/Clean-up script	Purpose
<code>prepare_db.sh</code>	Creates the <code>node1</code> , <code>node2</code> , and <code>node3</code> users in the CDB12c database; copies the credential stores to the OGG directories used by the practice
<code>catch-up-prac-11_1.sh</code>	Executes steps for Practice 11-1
<code>catch-up-prac-11_2.sh</code>	Executes steps for Practice 11-2

clean-up.sh	Wipes out all Extract and Replicat groups from Oracle GoldenGate and drops all objects from the database
-------------	----------------------------------------------------------------------------------------------------------

**Note:** You should run the `clean-up.sh` script at the end of Practice 11-2, so that you have a clean environment before starting Practice 12.

## Practice 11-1: Creating a Three-Way Active-Active Replication Environment

### Overview

In this practice, you create the necessary infrastructure for the implementation of a multi-master replication topology.

### Assumptions

The CDB12C database is up and running in ARCHIVELOG mode, supplemental logging has been enabled at the instance level, and the C##OGG\_ADMIN user exists and has the DBA, CDB\_DBA, and PDB\_DBA privileges. The admin user has also been granted Oracle GoldenGate privileges through the dbms\_goldengate\_auth.grant\_admin\_privilege procedure. Only one physical database is used to simulate a multi-master replication environment (CDB12C). Replication occurs between the three schemas mentioned above.

### Tasks

#### Preparing the database

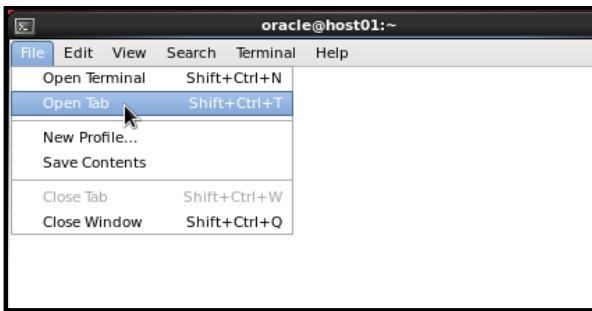
1. Open a terminal shell and navigate to the /home/oracle/labs/Practice11/configs directory. Run the `prepare_db.sh` script, which creates the database users/schemas node1, node2, and node3. These schemas require only connect, resource, and unlimited tablespace privileges. In addition, the script copies the Oracle GoldenGate credential files (`cwallet.sso`) from the /u03/ogg/ogg\_src subdirectories (dircrd and dirwlt) into the appropriate subdirectories under /u03/ogg/ogg\_node1, node2, and node3:

```
[OS prompt]$ cd ~/labs/Practice11/configs
[OS prompt]$ pwd
/home/oracle/labs/Practice11/configs
[OS prompt]$./prepare_db.sh
SQL*Plus: Release 12.1.0.2.0 Production on Tue May 5 21:02:04 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.

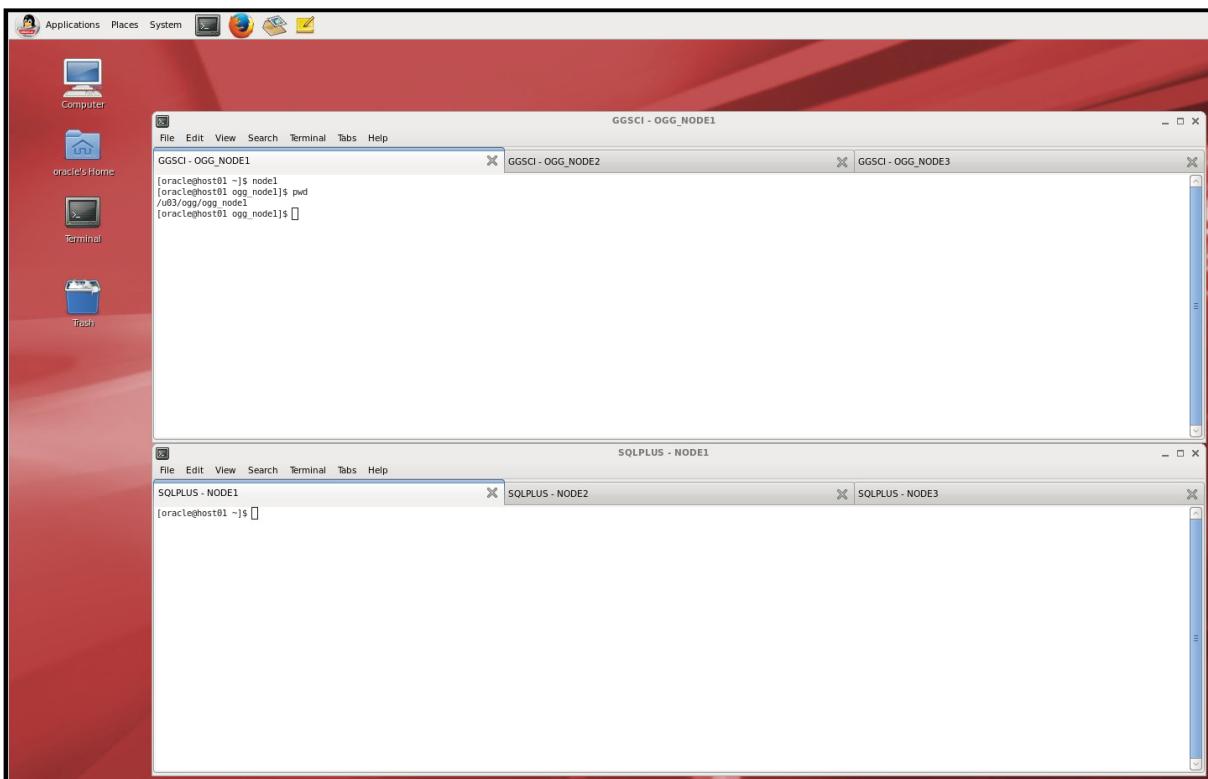
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL>
User created.
... Many lines omitted for clarity ...
SQL>
Grant succeeded.

SQL> Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
prepare_db.sh script completed
[OS prompt]$
```

- Configure your desktop in such a way that it displays two windows, one on top and one at the bottom. Stretch those windows to the right and open two tabs, in order to have three tabs per window:



- In the upper window, name the first tab "GGSCI - OGG\_NODE1," the second tab "GGSCI - OGG\_NODE2," and the third tab "GGSCI - OGG\_NODE3."
- In the lower window, name the first tab "SQLPLUS - NODE1," the second tab "SQLPLUS - NODE2," and the third tab "SQLPLUS - NODE3." Your environment should look like the screenshot below:



- Click the "SQLPLUS - NODE1" window, the lower window in your desktop. At the OS prompt, navigate to the ~/labs/Practice11/configs directory and launch sqlplus, connecting to the node1 schema (ask your instructor if you do not know or remember the password). Run the objects.sql script, which creates the database objects used in this practice, and the populate\_tables.sql script, which populates the tables with data. Also run sequences\_ogg\_node1.sql, which creates the non-overlapping sequences for this node:

```
[OS prompt]$ cd ~/labs/Practice11/configs
[OS prompt]$ sqlplus node1/<password>@ogg1
```

```
SQL*Plus: Release 12.1.0.2.0 Production on Sun Mar 15 10:28:35 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Sun Mar 15 2015 09:27:02 +11:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> @objects.sql
Table created.
... Many lines omitted for clarity ...
Table altered.
SQL> @populate_tables.sql
1 row created.
... Many lines omitted for clarity ...
Commit complete.
SQL> @sequences_ogg_node1.sql
Sequence created.
Sequence created.
Sequence created.
Sequence created.
```

6. Click the "SQLPLUS - NODE2" window. At the OS prompt, navigate to the `~/labs/Practice11/configs` directory and launch `sqlplus`, connecting to the `node2` schema. Run the `objects.sql` script, the `populate_tables.sql` script, and the `sequences_OGG_NODE2.sql` script, which creates the non-overlapping sequences for this node:

```
[OS prompt]$ cd ~/labs/Practice11/configs
[OS prompt]$ sqlplus node2/<password>@ogg1
SQL*Plus: Release 12.1.0.2.0 Production on Sun Mar 15 10:28:35 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Sun Mar 15 2015 09:27:02 +11:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> @objects.sql
Table created.
... Many lines omitted for clarity ...
Table altered.
SQL> @populate_tables.sql
1 row created.
... Many lines omitted for clarity ...
Commit complete.
SQL> @sequences_ogg_node2.sql
Sequence created.
```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

```
Sequence created.
Sequence created.
Sequence created.
```

- Click the "SQLPLUS - NODE3" window. At the OS prompt, navigate to the ~/labs/Practice11/configs directory and launch sqlplus, connecting to the node3 schema. Run the objects.sql script, the populate\_tables.sql script, and the sequences\_OGG\_NODE3.sql script, which creates the non-overlapping sequences for this node:

```
[OS prompt]$ cd ~/labs/Practice11/configs
[OS prompt]$ sqlplus node3/<password>@ogg1
SQL*Plus: Release 12.1.0.2.0 Production on Sun Mar 15 10:28:35 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Sun Mar 15 2015 09:27:02 +11:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real Application
Testing options
SQL> @objects.sql
Table created.
... Many lines omitted for clarity ...
Table altered.
SQL> @populate_tables.sql
1 row created.
... Many lines omitted for clarity ...
Commit complete.
SQL> @sequences_ogg_node3.sql
Sequence created.
Sequence created.
Sequence created.
Sequence created.
```

## Setting up Oracle GoldenGate Manager for all environments

- Click the "GGSCI - OGG\_NODE1" window and use the node1 alias to change the directory to /u03/ogg/ogg\_node1. Launch GGSCI and connect to the CDB12c database as C##OGG\_ADMIN by using the ogg\_admin alias. Select the "GGSCI - OGG\_NODE2" tab and use the node2 alias to change the directory to /u03/ogg/OGG\_NODE2. Launch GGSCI and connect to the CDB12c database as C##OGG\_ADMIN by using the ogg\_admin alias. Select the third tab ("GGSCI - OGG\_NODE3") and use the node3 alias. Change the directory to /u03/ogg/OGG\_NODE3. Launch GGSCI and connect to the CDB12c database as C##OGG\_ADMIN by using the ogg\_admin alias:

```
[OS prompt] $ node1
[OS prompt] $ pwd
/u03/ogg/ogg_node1
[OS prompt] $ ggsci
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
10:21:34
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
rights reserved.

GGSCI ([Host]) 1> DBLogin UserIdAlias ogg_admin
Successfully logged into database CDB$ROOT.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) >
```

Repeat the earlier steps for OGG\_NODE2 and OGG\_NODE3, making sure you select the tab corresponding to the environment.

- Select the OGG\_NODE1 tab and edit the parameter file for the manager:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params mgr
PORT 8100
DYNAMICPORTLIST 21000-21099
PurgeOldExtracts ./dirdat/*, UseCheckpoints
```

- Save the file and start the manager:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start mgr
Manager started.
```

- Select the OGG\_NODE2 tab and edit the parameter file for the manager:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params mgr
PORT 8200
DYNAMICPORTLIST 22000-22099
PurgeOldExtracts ./dirdat/*, UseCheckpoints
```

12. Save the file and start the manager:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start mgr
Manager started.
```

13. Select the OGG\_NODE3 tab and edit the parameter file for the manager:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params mgr
PORT 8300
DYNAMICPORTLIST 23000-23099
PurgeOldExtracts ./dirdat/*, UseCheckpoints
```

14. Save the parameter file and start the manager:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start mgr
Manager started.
```

### **Adding supplemental logging for the tables just created**

15. Select the OGG\_NODE1 tab and enter the TRANDATA statements for all tables created in the ogg\_node1 schema. You logged in to the database in step 3. If by any chance, you exited GGSCI and relaunched it, you would have to log in to the database again before issuing the TRANDATA command. You can either enter the commands or run the OBEY file ogg\_node1\_trandata.oby located in ~/labs/Practice11/configs.

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add trandata
ogg1.node1.account, cols
(ACCOUNT_TYPE, ACCOUNT_NUMBER, ACCOUNT_BALANCE, ACCOUNT_OPENED, ACCOUNT_BRANCH_ID, UPDATE_TS)
Logging of supplemental redo data enabled for table
OGG1.NODE1.ACCT.

TRANDATA for scheduling columns has been added on table
'OGG1.NODE1.ACCT'.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add trandata
ogg1.node1.branch,
cols(BRANCH_STREET_TYPE, BRANCH_STREET_NAME, BRANCH_STREET_NUMBER,
BRANCH_LEVEL, BRANCH_CITY, BRANCH_STATE, BRANCH_COUNTRY, BRANCH_SIZE
,BRANCH_OPENED, ZIP_CODE, BRANCH_NAME, UPDATE_TS)
Logging of supplemental redo data enabled for table
OGG1.NODE1.BRANCH.

TRANDATA for scheduling columns has been added on table
'OGG1.NODE1.BRANCH'.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add trandata
ogg1.node1.customer,
cols(NAME, MIDDLE_NAME, SURNAME, FULL_NAME, DOB, NATIONALITY, EMAIL, UPDATE_TS)
Logging of supplemental redo data enabled for table
OGG1.NODE1.CUSTOMER.

TRANDATA for scheduling columns has been added on table
'OGG1.NODE1.CUSTOMER'.
```

16. Select the OGG\_NODE2 tab and execute the TRANDATA statements for the tables created in the OGG\_NODE2 schema:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add trandata
ogg1.node2.account, cols
(ACCOUNT_TYPE, ACCOUNT_NUMBER, ACCOUNT_BALANCE, ACCOUNT_OPENED, ACCOUNT_BRANCH_ID, UPDATE_TS)
Logging of supplemental redo data enabled for table
OGG1.NODE2.ACCT.
TRANDATA for scheduling columns has been added on table
'OGG1.NODE2.ACCT'.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add trandata
ogg1.node2.branch,
cols(BRANCH_STREET_TYPE, BRANCH_STREET_NAME, BRANCH_STREET_NUMBER,
BRANCH_LEVEL, BRANCH_CITY, BRANCH_STATE, BRANCH_COUNTRY, BRANCH_SIZE
, BRANCH_OPENED, ZIP_CODE, BRANCH_NAME, UPDATE_TS)
Logging of supplemental redo data enabled for table
OGG1.NODE2.BRANCH.
TRANDATA for scheduling columns has been added on table
'OGG1.NODE2.BRANCH'.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add trandata
ogg1.node2.customer,
cols(NAME, MIDDLE_NAME, SURNAME, FULL_NAME, DOB, NATIONALITY, EMAIL, UPDATE_TS)
Logging of supplemental redo data enabled for table
OGG1.NODE2.CUSTOMER.
TRANDATA for scheduling columns has been added on table
'OGG1.NODE2.CUSTOMER'.
```

17. Select the OGG\_NODE3 tab and execute the TRANDATA statements for the tables created in the OGG\_NODE3 schema:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add trandata
ogg1.node3.account, cols
(ACCOUNT_TYPE, ACCOUNT_NUMBER, ACCOUNT_BALANCE, ACCOUNT_OPENED, ACCOUNT_BRANCH_ID, UPDATE_TS)
Logging of supplemental redo data enabled for table
OGG1.NODE3.ACCT.
TRANDATA for scheduling columns has been added on table
'OGG1.NODE3.ACCT'.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add trandata
ogg1.node3.branch,
cols(BRANCH_STREET_TYPE, BRANCH_STREET_NAME, BRANCH_STREET_NUMBER,
BRANCH_LEVEL, BRANCH_CITY, BRANCH_STATE, BRANCH_COUNTRY, BRANCH_SIZE
, BRANCH_OPENED, ZIP_CODE, BRANCH_NAME, UPDATE_TS)
Logging of supplemental redo data enabled for table
OGG1.NODE3.BRANCH.
TRANDATA for scheduling columns has been added on table
'OGG1.NODE3.BRANCH'.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add trandata
ogg1.node3.customer,
```

```

cols (NAME, MIDDLE_NAME, SURNAME, FULL_NAME, DOB, NATIONALITY, EMAIL, UP
DATE_TS)
Logging of supplemental redo data enabled for table
OGG1.NODE3.CUSTOMER.
TRANSDATA for scheduling columns has been added on table
'OGG1.NODE3.CUSTOMER'.

```

### Creating the Extract groups

18. Select the "GGSCI - OGG\_NODE1" window. Edit the parameter file for the EXDBA Extract group:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
exdba
```

```

Extract exdba
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "CDB12c")
ExtTrail ./dirdat/aa
UserIdAlias ogg_admin
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
StatOptions ResetReportStats
ReportCount every 10 Records
TranLogOptions ExcludeUser C##OGG_ADMIN
SOURCECATALOG ogg1
Table node1.* , GetBeforeCols (on update all);

```

The Extract group definition is straightforward. You enable the capturing of the before image data by using the GetBeforeUpdates keyword, and you specify that the before image of all columns that belong to all tables should be captured for updates - GetBeforeCols (on update all).

19. Save the file and add the Extract. Register it with the database (ogg1). You will be using the local trail ./dirdat/aa to store the RDBMS rows locally:

```

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > register
EXTRACT EXDBA DATABASE CONTAINER (ogg1)
Extract EXDBA successfully registered with database at SCN
3459126.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
exdba, Integrated tranlog, begin now
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add exttrail
./dirdat/aa, extract exdba, megabytes 10

```

20. Select the "GGSCI - OGG\_NODE2" window. Edit the parameter file for the EXDBB Extract group:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
exdbb
```

```
Extract exdbb
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "CDB12c")
ExtTrail ./dirdat/bb
UserIdAlias ogg_admin
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
StatOptions ResetReportStats
ReportCount every 10 Records
TranLogOptions ExcludeUser C##OGG_ADMIN
SOURCECATALOG ogg1
Table node2.* , GetBeforeCols (on update all);
```

21. Save the file and add the Extract. Register it with the database (ogg1). You will be using the local trail ./dirdat/bb to store the RDBMS rows locally:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > register
Extract EXDBB database container (ogg1)
Extract EXDBB successfully registered with database at SCN
1752694.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
exdbb, Integrated tranlog, begin now
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add exttrail
.dirdat/bb, extract exdbb, megabytes 10
```

22. Select the "GGSCI - OGG\_NODE3" window. Edit the parameter file for the EXDBC Extract group:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
exdbc
```

```
Extract exdbc
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "CDB12c")
ExtTrail ./dirdat/cc
UserIdAlias ogg_admin
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
StatOptions ResetReportStats
ReportCount every 10 Records
TranLogOptions ExcludeUser C##OGG_ADMIN
SOURCECATALOG ogg1
Table node3.* , GetBeforeCols (on update all);
```

23. Save the file and add the Extract. Register it with the database (ogg1). You will be using the local trail ./dirdat/cc to store the RDBMS rows locally:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > register
Extract EXDBC database container (ogg1)
Extract EXDBC successfully registered with database at SCN
1804659.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
exdbc, Integrated tranlog, begin now
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add exttrail
./dirdat/cc, extract exdbc, megabytes 10
```

### Creating the Data Pumps

24. Select the "GGSCI - OGG\_NODE1" window. Edit the parameter file for the DPDBB Data Pump. DPDBB pumps the data from node1 to node2:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
dpdbb
```

```
Extract dpdbb
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV (ORACLE_SID='CDB12c')
UserIdAlias ogg_admin
RmtHost OGG_NODE2, Mgrport 8200
RmtTrail ./dirdat/ab
Passthru
SOURCECATALOG ogg1
Table node1.*;
```

The Data Pump parameter file is also straightforward. Because the table definition is identical across the three schemas, the pump is defined as PASSTHRU. The wildcard sign (\*) specifies that data belonging to all tables that are defined in the `ogg_node1` schema should be transferred to the remote trail.

- Save the parameter file for DPDBB. Add the `dpdbb` Data Pump to the Oracle GoldenGate instance and define the RMTTRAIL for it:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
dpdbb, exttrailsource ./dirdat/aa
EXTRACT added.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add rmttrail
./dirdat/ab, extract dpdbb, megabytes 10
RMTTRAIL added.
```

- Edit the parameter file for the DPDBC Data Pump. DPDBC pumps the data from node1 to node3:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
dpdbc
Extract dpdbc
SETENV (ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV (ORACLE_SID='CDB12c')
UserIdAlias ogg_admin
RmtHost OGG_NODE3, Mgrport 8300
RmtTrail ./dirdat/ac
Passthru
SOURCECATALOG oggl
Table node1.*;
```

- Save the parameter file for DPDBC. Add the `dpdbc` Data Pump to the Oracle GoldenGate instance and define the RMTTRAIL for it:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
dpdbc, exttrailsource ./dirdat/aa
EXTRACT added.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add rmttrail
./dirdat/ac, extract dpdbc, megabytes 10
RMTTRAIL added.
```

- Select the "GGSCI - OGG\_NODE2" window. Edit the parameter file for the DPDBA Data Pump. DPDBA pumps the data from node2 to node1:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
dpdba
Extract dpdba
SETENV (ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV (ORACLE_SID='CDB12c')
UserIdAlias ogg_admin
RmtHost ogg_node1, Mgrport 8100
RmtTrail ./dirdat/ba
Passthru
```

```
SOURCECATALOG ogg1
Table node2.*;
```

29. Save the parameter file for DPDBA. Add the dpdba Data Pump to the Oracle GoldenGate instance and define the RMTTRAIL for it:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
dpdba, exttrailsource ./dirdat/bb
EXTRACT added.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add rmttrail
./dirdat/ba, extract dpdba, megabytes 10
RMTTRAIL added.
```

30. Edit the parameter file for the DPDBC Data Pump. DPDBC pumps the data from node2 to node3:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
dpdbc
```

```
Extract dpdbc
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV (ORACLE_SID='CDB12c')
UserIdAlias ogg_admin
RmtHost OGG_NODE3, Mgrport 8300
RmtTrail ./dirdat/bc
Passthru
SOURCECATALOG ogg1
Table node2.*;
```

31. Save the parameter file for DPDBC. Add the dpdbc Data Pump to the Oracle GoldenGate instance and define the RMTTRAIL for it:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
dpdbc, exttrailsource ./dirdat/bb
EXTRACT added.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add rmttrail
./dirdat/bc, extract dpdbc, megabytes 10
RMTTRAIL added.
```

32. Select the "GGSCI - OGG\_NODE3" window. Edit the parameter file for the DPDBA Data Pump. DPDBA pumps the data from node3 to node1:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
dpdba
```

```
Extract dpdba
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV (ORACLE_SID='CDB12c')
UserIdAlias ogg_admin
RmtHost ogg_node1, Mgrport 8100
RmtTrail ./dirdat/ca
Passthru
SOURCECATALOG ogg1
Table node3.*;
```

33. Save the parameter file for DPDBA. Add the dpdba Data Pump to the Oracle GoldenGate instance and define the RMTTRAIL for it:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
dpdba, exttrailsource ./dirdat/cc
EXTRACT added.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add rmttrail
./dirdat/ca, extract dpdba, megabytes 10
RMTTRAIL added.
```

34. Edit the parameter file for the DPDBB Data Pump. DPDBB pumps the data from node3 to node2:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
dpdbb
Extract dpdbb
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV (ORACLE_SID='CDB12c')
UserIdAlias ogg_admin
RmtHost OGG_NODE2, Mgrport 8200
RmtTrail ./dirdat/cb
Passthru
SOURCECATALOG ogg1
Table node3.*;
```

35. Save the parameter file for DPDBB. Add the DPDBB Data Pump to the Oracle GoldenGate instance and define the RMTTRAIL for it:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add extract
dpdbb, exttrailsource ./dirdat/cc
EXTRACT added.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add rmttrail
./dirdat/cb, extract dpdbb, megabytes 10
RMTTRAIL added.
```

## Creating the Replicats

36. Select the "GGSCI - OGG\_NODE1" window. Edit the parameter file for the REABB Replicat group. The REABB Replicat stores into the node1 schema the rows pumped into the ./dirdat/ba trail by the DPDBA Data Pump defined on node2:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
reabb
```

```
Replicat reabb
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "CDB12c")
DBOPTIONS INTEGRATEDPARAMS (parallelism 4)
AssumeTargetDefs
Discardfile ./dirrpt/reabb.dsc, Purge
UserIDAlias ogg1_repl
Map ogg1.node2.* , Target ogg1.node1.*;
```

The Replicat parameters used above ensure basic replication. You will add more advanced features, such as conflict detection and resolution, in the following practices.

37. Save the parameter file. Add the Replicat to the Oracle GoldenGate instance:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add replicat
 reabb, Integrated exttrail ./dirdat/ba
REPLICAT (Integrated) added.
```

38. Edit the parameter file for the REABC Replicat group. The rows transmitted into the ./dirdat/ca trail file by the DPDBA Data Pump, which is defined on the OGG\_NODE3 node, are stored into the ogg\_node1 schema by the REABC Replicat:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
 reabc
```

```
Replicat reabc
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "CDB12c")
DBOPTIONS INTEGRATEDPARAMS(parallelism 4)
AssumeTargetDefs
Discardfile ./dirrpt/reabc.dsc, Purge
UserIDAlias ogg1_repl
Map ogg1.node3.* ,Target ogg1.node1.*;
```

39. Save the parameter file. Add the Replicat to the Oracle GoldenGate instance:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add replicat
 reabc, Integrated exttrail ./dirdat/ca
REPLICAT (Integrated) added.
```

40. Select the "GGSCI - OGG\_NODE2" window. Edit the parameter file for the REBBA Replicat group. The rows transmitted into the ./dirdat/ca trail file by the DPDBA Data Pump, which is defined on the ogg\_node1 node, are stored in the node2 schema by the REBBA Replicat:

```
GGSCI GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit
 params rebba
```

```
Replicat rebba
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "CDB12c")
DBOPTIONS INTEGRATEDPARAMS(parallelism 4)
AssumeTargetDefs
Discardfile ./dirrpt/rebba.dsc, Purge
UserIDAlias ogg1_repl
Map ogg1.node1.* ,Target ogg1.node2.*;
```

41. Save the parameter file. Add the Replicat to the Oracle GoldenGate instance:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add replicat
rebba, Integrated exttrail ./dirdat/ab
REPLICAT (Integrated) added.
```

42. Edit the parameter file for the REBBC Replicat group. The rows transmitted into the ./dirdat/ca trail file by the DPDBA Data Pump, which is defined on the OGG\_NODE3 node, are stored in the node2 schema by the REBBC Replicat:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
rebba
```

```
Replicat rebba
SETENV (ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV (ORACLE_SID = "CDB12c")
DBOPTIONS INTEGRATEDPARAMS (parallelism 4)
AssumeTargetDefs
Discardfile ./dirrpt/rebbc.dsc, Purge
UserIDAlias ogg1_repl
Map ogg1.node3.* ,Target ogg1.node2.*;
```

43. Save the parameter file. Add the Replicat to the Oracle GoldenGate instance:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add replicat
rebba, Integrated exttrail ./dirdat/cb
REPLICAT (Integrated) added.
```

44. Select the "GGSCI - OGG\_NODE3" window. Edit the parameter file for the RECBA Replicat group. The rows transmitted into the ./dirdat/ca trail file by the DPDBA Data Pump, which is defined on the ogg\_node1 node, are stored in the node3 schema by the RECBA Replicat:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
recba
```

```
Replicat recba
SETENV (ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV (ORACLE_SID = "CDB12c")
DBOPTIONS INTEGRATEDPARAMS (parallelism 4)
AssumeTargetDefs
Discardfile ./dirrpt/recba.dsc, Purge
UserIDAlias ogg1_repl
Map ogg1.node1.* ,Target ogg1.node3.*;
```

45. Save the parameter file. Add the Replicat to the Oracle GoldenGate instance:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add replicat
recba, Integrated exttrail ./dirdat/ac
REPLICAT (Integrated) added.
```

46. Edit the parameter file for the RECBB Replicat group. The rows transmitted into the ./dirdat/ca trail file by the DPDBA Data Pump, which is defined on the OGG\_NODE2 node, are stored in the node3 schema by the RECBB Replicat:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
recbb

Replicat recbb
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "CDB12c")
DBOPTIONS INTEGRATEDPARAMS (parallelism 4)
AssumeTargetDefs
Discardfile ./dirrpt/recbb.dsc, Purge
UserIDAlias ogg1_repl
Map ogg1.node2.* ,Target ogg1.node3.*;
```

47. Save the parameter file. Add the Replicat to the Oracle GoldenGate instance:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > add replicat
recbb, Integrated extrail ./dirdat/bc
REPLICAT (Integrated) added.
```

### Testing the basic three-way replication

48. You have created the necessary infrastructure to implement basic three-way replication. You can now start all Extract/Replicat groups and generate some database activity to verify that replication is working across the three instances. Select the "GGSCI - OGG\_NODE1" window and start all groups defined for that Oracle GoldenGate instance:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start er *

Sending START request to MANAGER ...
EXTRACT DPDBB starting
Sending START request to MANAGER ...
EXTRACT DPDBC starting
Sending START request to MANAGER ...
EXTRACT EXDBA starting
Sending START request to MANAGER ...
REPLICAT REABB starting
Sending START request to MANAGER ...
REPLICAT REABC starting

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING DPDBB 00:00:00 00:00:06
EXTRACT RUNNING DPDBC 00:00:00 00:00:06
```

EXTRACT	RUNNING	EXDBA	00:00:03	00:00:05
REPLICAT	RUNNING	REABB	00:00:00	00:00:08
REPLICAT	RUNNING	REABC	00:00:00	00:00:09

49. Select the "GGSCI - OGG\_NODE2" window and start all groups defined for that Oracle GoldenGate instance:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start er *
Sending START request to MANAGER ...
EXTRACT DPDBA starting
Sending START request to MANAGER ...
EXTRACT DPDBC starting
Sending START request to MANAGER ...
EXTRACT EXDBB starting
Sending START request to MANAGER ...
REPLICAT REBBA starting
Sending START request to MANAGER ...
REPLICAT REBBC starting

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all

Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING DPDBA 00:00:00 00:00:04
EXTRACT RUNNING DPDBC 00:00:00 00:00:04
EXTRACT RUNNING EXDBB 00:00:06 00:00:04
REPLICAT RUNNING REBBA 00:00:00 00:00:07
REPLICAT RUNNING REBBC 00:00:00 00:00:06
```

50. Select the "GGSCI - OGG\_NODE3" window and start all groups defined for that Oracle GoldenGate instance:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start er *
Sending START request to MANAGER ...
EXTRACT DPDBA starting
Sending START request to MANAGER ...
EXTRACT DPDBB starting
Sending START request to MANAGER ...
EXTRACT EXDBC starting
Sending START request to MANAGER ...
REPLICAT RECBA starting
Sending START request to MANAGER ...
REPLICAT RECBB starting

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING DPDBA 00:00:00 00:00:04
EXTRACT RUNNING DPDBB 00:00:00 00:00:04
EXTRACT RUNNING EXDBC 00:00:06 00:00:04
REPLICAT RUNNING RECBA 00:00:00 00:00:07
REPLICAT RUNNING RECBB 00:00:00 00:00:05
```

51. Generate some database activity and verify that replication works. Click the "SQLPLUS - NODE3" window, and add one row to the CUSTOMER table (note that you do not need to provide a value for the UPDATE\_TS column as that column defaults to SYSTIMESTAMP)

```
SQL> set lines 132
SQL> desc customer
Name Null? Type

CUST_ID NOT NULL NUMBER(12)
NAME NOT NULL VARCHAR2(50)
MIDDLE_NAME
SURNAME
FULL_NAME
DOB
NATIONALITY
EMAIL
UPDATE_TS NOT NULL TIMESTAMP(6)

SQL> insert into CUSTOMER (CUST_ID, NAME, MIDDLE_NAME, SURNAME, FULL_NAME, DOB, NATIONALITY, EMAIL) Values
(320, 'Toby', 'Rafael', 'Rechner', 'Toby Rafael Rechner',
```

```

TO_DATE('11/10/1972 00:00:00', 'MM/DD/YYYY HH24:MI:SS') ,
'AUSTRALIAN', 'tfrechner@dumginec.com');

1 row created.

SQL> commit;

SQL> select name, surname from customer where cust_id=320;

NAME SURNAME

Toby Rechner

SQL>

```

52. Select the "SQLPLUS-NODE2" window and verify that the new row that was inserted into the CUSTOMER table on the node3 schema has been replicated into the node2 schema:

```

SQL> set lines 132
SQL> select name, surname from customer where cust_id=320;

NAME SURNAME

Toby Rechner

SQL>

```

53. Select the "SQLPLUS-NODE1" tab and verify that the new row that was inserted into the CUSTOMER table on the node3 schema has also been replicated into the node1 schema:

```

SQL> set lines 132
SQL> select name, surname from customer where cust_id=320;

NAME SURNAME

Toby Rechner

SQL>

```

54. Select the window that hosts the three tabs connected to the Oracle GoldenGate instances. Click the "GGSCI - OGG\_NODE3" tab, where GGSCI is connected to the OGG\_NODE3 instance. Enter a stats exdbc command to display the statistics for the Extract group:

```

GGSCI ([Host]) 19> stats exdbc
Sending STATS request to EXTRACT EXDBC ...
Start of Statistics at 2015-03-22 09:26:11.
Output to ./dirdat/cc:
Extracting from OGG1.NODE3.CUSTOMER to OGG1.NODE3.CUSTOMER:
*** Total statistics since 2015-03-22 09:26:05 ***
 Total inserts 1.00
 Total updates 0.00
 Total deletes 0.00
 Total discards 0.00

```

```

 Total operations 1.00
*** Daily statistics since 2015-03-22 09:26:05 ***
 Total inserts 1.00
 Total updates 0.00
 Total deletes 0.00
 Total discards 0.00
 Total operations 1.00
*** Hourly statistics since 2015-03-22 09:26:05 ***
 Total inserts 1.00
 Total updates 0.00
 Total deletes 0.00
 Total discards 0.00
 Total operations 1.00
*** Latest statistics since 2015-03-22 09:26:05 ***
 Total inserts 1.00
 Total updates 0.00
 Total deletes 0.00
 Total discards 0.00
 Total operations 1.00
End of Statistics.

```

55. Because you have added a row in the node3 schema, you should also see statistics for the two Data Pumps (DPDBA and DPDBB) in the OGG\_NODE3 instance, the REBBC Replicat in the OGG\_NODE2 instance, and the REABC Replicat in the ogg\_node1 node. Select the tabs corresponding to the various instances and display the statistics for all Data Pumps and Replicats involved in the replication.
56. Select the window with the three tabs where sqlplus is connected to the three schemas. Click the first tab ("SQLPLUS - NODE1") and update the CUSTOMER table to verify that not only inserts, but also updates, are correctly replicated:

```

SQL> update CUSTOMER set MIDDLE_NAME = 'Antonio' where CUST_ID =
320;
1 row updated.
SQL> commit;

```

57. Select the "SQLPLUS - NODE2" tab, where the sqlplus session is running, and select the MIDDLE\_NAME column from the CUSTOMER table for the customer with CUST\_ID equals to 320 to verify that the update has been replicated:

```
SQL> select middle_name from customer where cust_id=320;
MIDDLE_NAME

Antonio
SQL>
```

58. The last step verified that you have basic replication working across the three nodes. The following practices will deal with data conflict and resolution. Leave the windows and the tabs inside the windows open, because you will reuse them in the next practice.

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

## Practice 11-2: Configuring CDR for a Multi-Master Replication Environment

### Overview

In this practice, you build on what you accomplished so far, and use Oracle GoldenGate CDR to add conflict detection and resolution to the multi-master topology that is configured in Practice 11-1.

### Assumptions

Practice 11-1 has been completed successfully and a three-way replication environment is running.

### Tasks

#### Stopping and editing all Replicats to add CDR directives

1. Select the window with the three tabs connected to the Oracle GoldenGate instances. Click the tab where GGSCI is connected to the OGG\_NODE1 instance and stop the two Replicat groups:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop re*
Sending STOP request to REPLICAT REABB ...
Request processed.
Sending STOP request to REPLICAT REABC ...
Request processed.
```

2. Edit the Replicat parameter files for the two Replicats REABB and REABC and add the CDR directives. For the ACCOUNT table, you want a USEDELTA resolution method when an update conflict is detected. The ACCOUNT\_BALANCE column stores a numeric value that could be updated concurrently by more than one session, generating erroneous results if the before image is not checked. Conflicts affecting the other two tables BRANCH and CUSTOMER can be resolved by using the USEMAX method applied to the UPDATE\_TS column, which stores the time stamp of the moment the change occurred. USEMAX chooses the most recent time stamp. You must replace the line that performs a wildcard mapping (Map ogg1.node2.\* ,Target ogg1.node3.\* ; ) with three individual Map statements, each indicating the specific CDR type chosen to resolve potential conflicts. The lines in bold show the directives that you must add to the Replicat parameter file:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
REABB
```

```
Replicat reabb
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "CDB12c")
DBOPTIONS INTEGRATEDPARAMS(parallelism 4)
AssumeTargetDefs
Discardfile ./dirrpt/reabb.dsc, Purge
UserIDAlias ogg1_repl
AllowDupTargetMap
getupdates
```

```

getinserts
getdeletes
Map ogg1.node2.account, Target ogg1.node1.account,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (delta_resolution_method, usedelta, cols(account_balance)),
 (max_resolution_method, usemax(update_ts),
 cols(account_type,account_number,account_opened,
 account_branch_id,update_ts)),
 (default overwrite));
Map ogg1.node2.branch, Target ogg1.node1.branch,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (max_resolution_method, usemax(update_ts),
 cols(branch_street_type,branch_street_name,
 branch_street_number,branch_level,branch_city,branch_state,
 branch_country,branch_size,branch_opened,zip_code,
 branch_name, update_ts)),
 (default overwrite));
Map ogg1.node2.customer, Target ogg1.node1.customer,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (max_resolution_method, usemax(update_ts),
 cols(name,middle_name,surname,full_name,dob,nationality,
 email,update_ts)),
 (default overwrite));

```

Note that the original Map statement gets replaced.

- Save the parameter file for REABB. Edit the parameter file for the REABC Replicat group:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
REABC
```

```

Replicat reabc
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "CDB12c")
DBOPTIONS INTEGRATEDPARAMS (parallelism 4)
AssumeTargetDefs
Discardfile ./dirrpt/reabc.dsc, Purge
UserIDAlias ogg1_repl
AllowDupTargetMap
getupdates
getinserts
getdeletes
Map ogg1.node3.account, Target ogg1.node1.account,
```

```

comparecols(on update all)
resolveconflict(updaterowexists,
(delta_resolution_method, usedelta, cols(account_balance)),
(max_resolution_method, usemax(update_ts),
cols(account_type,account_number,account_opened,
account_branch_id,update_ts)),
(default overwrite));
Map ogg1.node3.branch, Target ogg1.node1.branch,
comparecols(on update all)
resolveconflict(updaterowexists,
(max_resolution_method, usemax(update_ts),
cols(branch_street_type,branch_street_name,
branch_street_number,branch_level,branch_city,branch_state,
branch_country,branch_size,branch_opened,zip_code,
branch_name, update_ts)),
(default overwrite));
Map ogg1.node3.customer, Target ogg1.node1.customer,
comparecols(on update all)
resolveconflict(updaterowexists,
(max_resolution_method, usemax(update_ts),
cols(name,middle_name,surname,full_name,dob,nationality,
email,update_ts)),
(default overwrite));

```

- Save the parameter file. When you are back in GGSCI, start the Replicat groups you just modified:

```

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start re*
Sending START request to MANAGER ...
REPLICAT REABB starting
Sending START request to MANAGER ...
REPLICAT REABC starting

```

- Select the OGG\_NODE2 tab and stop the Replicat groups:

```

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop re*
Sending STOP request to REPLICAT REBBA ...
Request processed.
Sending STOP request to REPLICAT REBBC ...
Request processed.

```

- Edit the REBBA parameter file and add the CDR directives after removing the wildcard mapping (Map ogg1.node2.\* ,Target ogg1.node2.\* ;). The CDR directives are the same for all Replicat groups; only the mapping schema differs. In the case of the REBBA Replicat that is running on the OGG\_NODE2 node, you are mapping tables from the node1 schema (the replication source) to the node2 schema (the replication target):

```

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
REBBA

Replicat rebba
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "CDB12c")
DBOPTIONS INTEGRATEDPARAMS(parallelism 4)
AssumeTargetDefs
Discardfile ./dirrpt/rebba.dsc, Purge
UserIDAlias ogg1_repl
AllowDupTargetMap
getupdates
getinserts
getdeletes
Map ogg1.node1.account, Target ogg1.node2.account,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (delta_resolution_method, usedelta, cols(account_balance)),
 (max_resolution_method, usemax(update_ts),
 cols(account_type,account_number,account_opened,
 account_branch_id,update_ts)),
 (default overwrite));
Map ogg1.node1.branch, Target ogg1.node2.branch,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (max_resolution_method, usemax(update_ts),
 cols(branch_street_type,branch_street_name,
 branch_street_number,branch_level,branch_city,branch_state,
 branch_country,branch_size,branch_opened,zip_code,
 branch_name, update_ts)),
 (default overwrite));
Map ogg1.node1.customer, Target ogg1.node2.customer,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (max_resolution_method, usemax(update_ts),
 cols(name,middle_name,surname,full_name,dob,nationality,
 email,update_ts)),
 (default overwrite));

```

7. Save the parameter file and edit the parameter file for the REBBC Replicat:

```

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
REBBC

```

```

Replicat rebbc
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")

```

```

SETENV(ORACLE_SID = "CDB12c")
DBOPTIONS INTEGRATEDPARAMS (parallelism 4)
AssumeTargetDefs
Discardfile ./dir rpt/rebbc.dsc, Purge
UserIDAlias ogg1_repl
AllowDupTargetMap
getupdates
getinserts
getdeletes
Map ogg1.node3.account, Target ogg1.node2.account,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (delta_resolution_method, usedelta, cols(account_balance)),
 (max_resolution_method, usemax(update_ts),
 cols(account_type,account_number,account_opened,
 account_branch_id,update_ts)),
 (default overwrite));
Map ogg1.node3.branch, Target ogg1.node2.branch,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (max_resolution_method, usemax(update_ts),
 cols(branch_street_type,branch_street_name,
 branch_street_number,branch_level,branch_city,branch_state,
 branch_country,branch_size,branch_opened,zip_code,
 branch_name,update_ts)),
 (default overwrite));
Map ogg1.node3.customer, Target ogg1.node2.customer,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (max_resolution_method, usemax(update_ts),
 cols(name,middle_name,surname,full_name,dob,nationality,
 email,update_ts)),
 (default overwrite));

```

- Save the parameter file and restart all Replicat groups for the OGG\_NODE2 Oracle GoldenGate instance:

```

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start re*
Sending START request to MANAGER ...
REPLICAT REBBA starting
Sending START request to MANAGER ...
REPLICAT REBBC starting

```

- Select the OGG\_NODE3 tab and stop the Replicat groups:

```

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop re*

```

```
Sending STOP request to REPLICAT RECBA ...
Request processed.
Sending STOP request to REPLICAT RECBB ...
Request processed.
```

10. Edit the parameter file for the RECBA Replicat group, adding the CDR directives for the ACCOUNT, BRANCH, and CUSTOMER tables:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
RECBA
```

```
Replicat recba
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "CDB12c")
DBOPTIONS INTEGRATEDPARAMS(parallelism 4)
AssumeTargetDefs
Discardfile ./dirrpt/recba.dsc, Purge
UserIDAlias ogg1_repl
AllowDupTargetMap
getupdates
getinserts
getdeletes
Map ogg1.node1.account, Target ogg1.node3.account,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (delta_resolution_method, usedelta, cols(account_balance)),
 (max_resolution_method, usemax(update_ts),
 cols(account_type,account_number,account_opened,
 account_branch_id,update_ts)),
 (default overwrite));
Map ogg1.node1.branch, Target ogg1.node3.branch,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (max_resolution_method, usemax(update_ts),
 cols(branch_street_type,branch_street_name,
 branch_street_number,branch_level,branch_city,branch_state,
 branch_country,branch_size,branch_opened,zip_code,
 branch_name,update_ts)),
 (default overwrite));
Map ogg1.node1.customer, Target ogg1.node3.customer,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (max_resolution_method, usemax(update_ts),
 cols(name,middle_name,surname,full_name,dob,nationality,
 email,update_ts)),
 (default overwrite));
```

11. Save the parameter file. Edit the parameter file for the RECBB Replicat group:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > edit params
RECBB
```

```
Replicat recbb
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "CDB12c")
DBOPTIONS INTEGRATEDPARAMS (parallelism 4)
AssumeTargetDefs
Discardfile ./dirrpt/recbb.dsc, Purge
UserIDAlias ogg1_repl
AllowDupTargetMap
getupdates
getinserts
getdeletes
Map ogg1.node2.account, Target ogg1.node3.account,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (delta_resolution_method, usedelta, cols(account_balance)),
 (max_resolution_method, usemax(update_ts),
 cols(account_type,account_number,account_opened,
 account_branch_id,update_ts)),
 (default overwrite));
Map ogg1.node2.branch, Target ogg1.node3.branch,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (max_resolution_method, usemax(update_ts),
 cols(branch_street_type,branch_street_name,
 branch_street_number,branch_level,branch_city,branch_state,
 branch_country,branch_size,branch_opened,zip_code,
 branch_name, update_ts)),
 (default overwrite));
Map ogg1.node2.customer, Target ogg1.node3.customer,
 comparecols(on update all)
 resolveconflict(updaterowexists,
 (max_resolution_method, usemax(update_ts),
 cols(name,middle_name,surname,full_name,dob,nationality,
 email,update_ts)),
 (default overwrite));
```

12. Save the parameter file. When you are back in GGSCI, restart the Replicat groups running in the OGG\_NODE3 Oracle GoldenGate instance:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start re*
Sending START request to MANAGER ...
```

```
REPLICAT RECBA starting
Sending START request to MANAGER ...
REPLICAT RECBB starting
```

### Stopping all Data Pump Extracts to simulate network latency

13. To see CDR in action, you must use the usual trick of simulating network latency by stopping the Data Pumps and modifying the same row while logged in to different databases. Click the window with the three tabs connected to the Oracle GoldenGate instances, select the OGG\_NODE1 tab, and stop the two Data Pumps (DPDBB and DPDBC):

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop dp*
Sending STOP request to EXTRACT DPDBB ...
Request processed.
Sending STOP request to EXTRACT DPDBC ...
Request processed.
```

14. Select the OGG\_NODE2 tab and stop the Data Pumps DPDBA and DPDBC:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop dp*
Sending STOP request to EXTRACT DPDBA ...
Request processed.
Sending STOP request to EXTRACT DPDBC ...
Request processed.
```

15. Select the OGG\_NODE3 tab and stop the Data Pumps DPDBA and DPDBB:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > stop dp*
Sending STOP request to EXTRACT DPDBA ...
Request processed.
Sending STOP request to EXTRACT DPDBB ...
Request processed.
```

### Updating the same row in all databases to induce data conflict

16. While all Data Pumps are stopped, select the window with the three sqlplus sessions connected to the node1, node2, and node3 schemas. Click the SQLPLUS-NODE1 tab and update one row in the ACCOUNT table, for example:

```
SQL> select acct_id,cust_id,account_balance from account where
acct_id = 283 and cust_id = 83;
 ACCT_ID CUST_ID ACCOUNT_BALANCE

 283 83 9381
SQL> update account set account_balance = account_balance - 31
where acct_id = 283 and cust_id = 83 and account_balance = 9381;
1 row updated.
SQL> commit;
Commit complete.
```

17. Select the SQLPLUS-NODE2 tab and update the same row, this time subtracting \$20 from the account balance:

```
SQL> select acct_id,cust_id,account_balance from account where
acct_id = 283 and cust_id = 83;
ACCT_ID CUST_ID ACCOUNT_BALANCE

283 83 9381
SQL> update account set account_balance = account_balance - 20
where acct_id = 283 and cust_id = 83 and account_balance = 9381;
1 row updated.
SQL> commit;
Commit complete.
```

18. Select the SQLPLUS-NODE3 tab and update the same row in the ACCOUNT table, this time subtracting \$40 from the account balance:

```
SQL> select acct_id,cust_id,account_balance from account where
acct_id = 283 and cust_id = 83;
ACCT_ID CUST_ID ACCOUNT_BALANCE

283 83 9381
SQL> update account set account_balance = account_balance - 40
where acct_id = 283 and cust_id = 83 and account_balance = 9381;
1 row updated.
SQL> commit;
Commit complete.
```

### Restarting all Data Pump Extracts to Test Conflict Resolution

19. Select the window with the three tabs connected to the Oracle GoldenGate instances. Click the tab where GGSCI is connected to the OGG\_NODE1 instance and restart all Data Pumps:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start dp*
Sending START request to MANAGER ...
EXTRACT DPDBB starting
Sending START request to MANAGER ...
EXTRACT DPDBC starting
```

20. Click the tab where GGSCI is connected to the OGG\_NODE2 instance and restart all Data Pumps:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start dp*
Sending START request to MANAGER ...
EXTRACT DPDBA starting
Sending START request to MANAGER ...
EXTRACT DPDBC starting
```

21. Click the tab where GGSCI is connected to the OGG\_NODE3 instance and restart all Data Pumps:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/CDB$ROOT) > start dp*
Sending START request to MANAGER ...
EXTRACT DPDBA starting
Sending START request to MANAGER ...
EXTRACT DPDBB starting
```

22. Select the SQLPLUS-NODE1 tab in the window that hosts the three tabs connected to the Oracle GoldenGate instances and display the account balance for the ACCOUNT row where ACCT\_ID is 283 and CUST\_ID is 83. The expected balance is \$9,290. The first session subtracted \$31 from the balance, the second session subtracted \$20 from the balance, and the third session subtracted \$40 from the balance. The three schemas OGG\_NODE1, OGG\_NODE2, and OGG\_NODE3 should converge on the value of 9,290 for the ACCOUNT\_BALANCE field:

```
SQL> select acct_id,cust_id,account_balance from account where acct_id
= 283 and cust_id = 83;

ACCT_ID CUST_ID ACCOUNT_BALANCE

283 83 9290
```

23. Select the SQL\_NODE2 tab and verify that the OGG\_NODE2 schema has the same value for ACCOUNT\_BALANCE:

```
SQL> select acct_id,cust_id,account_balance from account where acct_id
= 283 and cust_id = 83;

ACCT_ID CUST_ID ACCOUNT_BALANCE

283 83 9290
```

24. Perform the same verification on the third schema, OGG\_NODE3. Select the OGG\_NODE3 tab and select the row in the ACCOUNT table where ACCT\_ID is 283 and CUST\_ID is 83:

```
SQL> select acct_id,cust_id,account_balance from account where
acct_id = 283 and cust_id = 83;

ACCT_ID CUST_ID ACCOUNT_BALANCE

283 83 9290
```

25. The three schemas NODE1, NODE2, and NODE3 show the same (correct) value for the ACCOUNT\_BALANCE field. The USEDELTA resolution method correctly computed the actual balance when a data conflict was detected. You can verify that CDR actually performed the required conflict resolution by displaying the statistics for the Replicat groups, specifying REPORTCDR as a STATS parameter. Select the window with the three tabs connected to the Oracle GoldenGate instances. Click the tab where GGSCI is connected to the OGG\_NODE1 instance and request the display of statistics that affect the REABB and REABC Replicat groups:

```
GGSCI ([Host]) 25> stats reabb reportcdr
Sending STATS request to REPLICAT REABB ...
Start of Statistics at 2015-03-22 15:03:36.
Integrated Replicat Statistics:
 Total transactions 1.00
 Redirected 0.00
 DDL operations 0.00
 Stored procedures 0.00
 Datatype functionality 0.00
 Event actions 0.00
 Direct transactions ratio 0.00%
Replicating from OGG1.NODE2.ACCT to OGG1.NODE1.ACCT:
*** Total statistics since 2015-03-22 14:22:16 ***
 Total inserts 0.00
 Total updates 1.00
 Total deletes 0.00
 Total discards 0.00
 Total operations 1.00
 Total CDR conflicts 1.00
 CDR resolutions succeeded 1.00
 CDR UPDATEROWEXISTS conflicts 1.00
*** Daily statistics since 2015-03-22 14:22:16 ***
 Total inserts 0.00
 Total updates 1.00
 Total deletes 0.00
 Total discards 0.00
 Total operations 1.00
 Total CDR conflicts 1.00
 CDR resolutions succeeded 1.00
 CDR UPDATEROWEXISTS conflicts 1.00
*** Hourly statistics since 2015-03-22 15:00:00 ***
 No database operations have been performed.
*** Latest statistics since 2015-03-22 14:22:16 ***
 Total inserts 0.00
 Total updates 1.00
```

```
Total deletes 0.00
Total discards 0.00
Total operations 1.00
Total CDR conflicts 1.00
CDR resolutions succeeded 1.00
CDR UPDATEROWEXISTS conflicts 1.00
End of Statistics.

GGSCI ([Host]) 26> stats reabc reportcdr
Sending STATS request to REPLICAT REABC ...
Start of Statistics at 2015-03-22 15:04:41.
Integrated Replicat Statistics:
 Total transactions 1.00
 Redirected 0.00
 DDL operations 0.00
 Stored procedures 0.00
 Datatype functionality 0.00
 Event actions 0.00
 Direct transactions ratio 0.00%
Replicating from OGG1.NODE3.ACCOUNT to OGG1.NODE1.ACCOUNT:
*** Total statistics since 2015-03-22 14:22:18 ***
 Total inserts 0.00
 Total updates 1.00
 Total deletes 0.00
 Total discards 0.00
 Total operations 1.00
 Total CDR conflicts 1.00
 CDR resolutions succeeded 1.00
 CDR UPDATEROWEXISTS conflicts 1.00
*** Daily statistics since 2015-03-22 14:22:18 ***
 Total inserts 0.00
 Total updates 1.00
 Total deletes 0.00
 Total discards 0.00
 Total operations 1.00
 Total CDR conflicts 1.00
 CDR resolutions succeeded 1.00
 CDR UPDATEROWEXISTS conflicts 1.00
*** Hourly statistics since 2015-03-22 15:00:00 ***
 No database operations have been performed.
*** Latest statistics since 2015-03-22 14:22:18 ***
 Total inserts 0.00
 Total updates 1.00
```

Total deletes	0.00
Total discards	0.00
Total operations	1.00
Total CDR conflicts	1.00
CDR resolutions succeeded	1.00
CDR UPDATEROWEXISTS conflicts	1.00
End of Statistics.	

26. Select the "GGSCI - OGG\_NODE2" tab and request the statistics for the Replicat groups defined for the OGG\_NODE2 Oracle GoldenGate instance (REBBA and REBBC). Do the same for the OGG\_NODE3 instance, displaying the statistics for the RECBA and RECBB Replicat groups.
27. Stop all Extract and Replicat groups in all Oracle GoldenGate instances. Delete all groups and unregister the Extract groups from the database. Alternatively, you can run the clean-up script `clean-up.sh` found in the `~/labs/Practice11/configs` directory, which does all that for you. Terminate all open windows as you will not need those windows open in the next practice.

### What Have You Achieved After Completing Practice 11?

In Practice 11, you have set up a complete multi-master configuration. Initially you configured simple replication with no data conflict detection and resolution (Practice 11-1). Subsequently, you refined your multi-master topology by introducing automatic conflict detection and resolution by using the CDR facility provided by Oracle GoldenGate (Practice 11-2).

This completes Practice 11-2.

This completes Practice 11. Stop here.



## **Practices for Lesson 12: Active Data Guard and Oracle GoldenGate: How to Achieve Maximum Availability**

**Chapter 12**

## Practices for Lesson 12: Overview

---

### Practices Overview

In these practices, you will install the Oracle Bundled Agent (XAG) software and configure Oracle GoldenGate to transparently survive the loss of a RAC node while replicating data from the Oracle RAC environment to the CDB12c database that is running on the host computer.

There are no catch-up or clean-up scripts for this practice.

## Practice 12-1: Prepare the Environment

### Overview

In this practice, you shut down the Oracle RDBMS instance DWNSTR (if it is running) to free up some memory for the Oracle RAC instance OGGRAC. You then start the two virtual machines o16-121-rac1 and o16-121-rac2 and install the Oracle Bundled Agent software on both nodes.

### Assumptions

The Oracle DBFS file system has been correctly installed and it is configured as an Oracle Clusterware resource. Practice 4 has completed successfully.

### Tasks

1. On the host computer, open a terminal shell window and check if the DWNSTR Oracle RDBMS instance is running. Use the UNIX/Linux `ps -ef` command to show if the `pmon` process for the DWNSTR instance is running. You should see a `pmon` process for the `CDB12C` instance. If you also determine that the DWNSTR instance is running, you must shut it down because that instance is not used and it takes up valuable memory space, which reduces the memory available for VirtualBox. If you must stop DWNSTR, set the Oracle environment variable to point to DWNSTR and use `sqlplus`, connecting as `sysdba`, to shut down the instance. After the instance has been shut down, `exit sqlplus` and terminate the window:

```
[OS prompt]$ ps -ef | grep pmon
oracle 4223 1 0 12:54 ? 00:00:00 ora_pmon_CDB12C
oracle 4575 1 0 12:56 ? 00:00:00 ora_pmon_DWNSTR
oracle 4908 4194 0 12:58 pts/0 00:00:00 grep pmon
[OS prompt]$ export ORACLE_SID=DWNSTR
[OS prompt]$ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Sun Mar 29 13:10:51
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
```

```
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
[OS prompt] $ exit
```

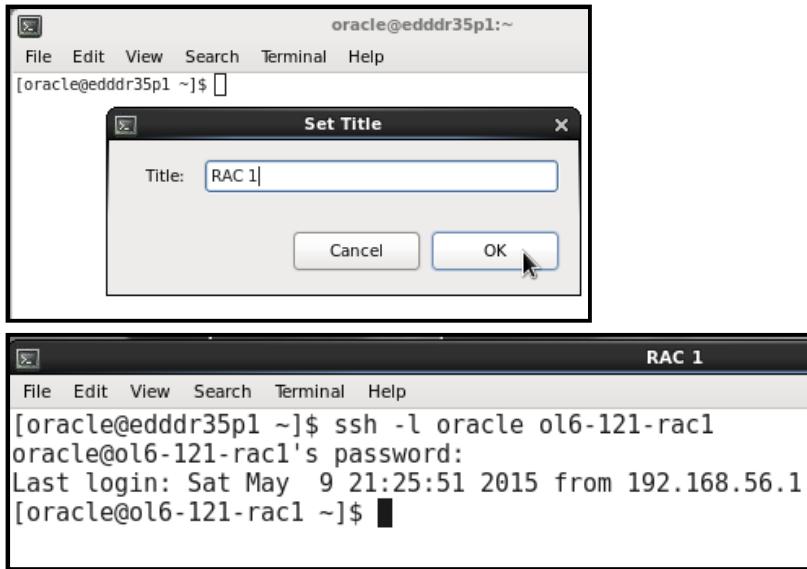
2. Open or activate the window where the Oracle VirtualBox Manager is running. Click the `ol6-121-rac1` icon in the left pane, and then click the Start icon (the green arrow on the top panel) to start the `ol6-121-rac1` virtual machine:



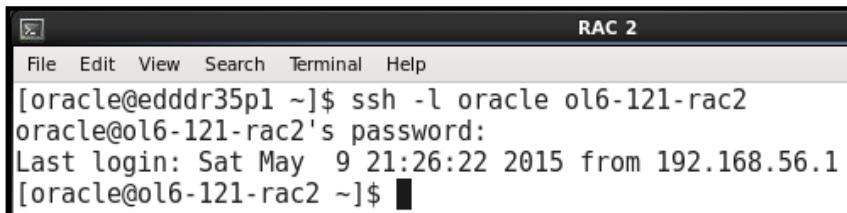
3. When the control returns to the VirtualBox Manager window after starting `ol6-121-rac1`, click the `ol6-121-rac2` icon in the left pane, and then click the Start icon (the green arrow) on the top panel to also start the `ol6-121-rac2` virtual machine:



4. On the host computer, create a terminal shell, set its title to "RAC 1," and use `ssh` to connect to the `ol6-121-rac1` virtual machine:



- On the host computer, create another terminal shell and use ssh, this time to connect to the ol6-121-rac2 virtual machine. Set the terminal shell title to "RAC 2."



### Installing the Oracle Bundled Agent (XAG) software

- Click the "RAC 1" terminal window. Expand (unzip) the zip file containing the Oracle Grid infrastructure Bundled Agents into the /tmp directory. You find the file xagpack\_6.zip in the /media/sf\_install directory:

```
[OS prompt] $ cd /tmp
[OS prompt] $ unzip /media/sf_install/xagpack_6.zip
 creating: xag/
 inflating: xag/xagsetup.sh
 creating: xag/template/
[many rows omitted...]
 inflating: xag/agapacheas.pm
 inflating: xag/agpsoft.pm
[OS prompt] $
```

- Change the directory to /tmp/xag and run the xagsetup.sh script with the '--install', '--directory,' and '--all\_nodes' arguments. The directory argument needs to point to a directory outside of the grid infrastructure home. Choose /u01/app/oracle/product/xag as the XAG install directory:

```
[OS prompt] $ cd xag
[OS prompt] $./xagsetup.sh --install --directory
/u01/app/oracle/product/xag --all_nodes
Installing Oracle Grid Infrastructure Agents on: ol6-121-rac1
```

```
Preparing to install Oracle Grid Infrastructure Agents on nodes:
ol6-121-rac2
Installing Oracle Grid Infrastructure Agents on ol6-121-rac2
Done.
[OS prompt] $
```

8. Make sure the agctl has been correctly installed:

```
[OS prompt] $ /u01/app/oracle/product/xag/bin/agctl query
releaseversion
The Oracle Grid Infrastructure Agents release version is 6.1.1
[OS prompt] $
```

### Enabling data replication for the OGGRAC Oracle RAC instance

9. The OGGRAC instance should be in the shutdown state (you shut the instance down at the end of practice 4-3 [step 4.3.8]). Click the "RAC 1" terminal window and invoke the db\_env alias to set the required environment variables and use the svrctl utility. Verify that the OGGRAC instance is not running:

```
[OS prompt] $ db_env
[OS prompt] $ svrctl status database -d OGGRAC
Instance OGGRAC1 is not running on node ol6-121-rac1
Instance OGGRAC2 is not running on node ol6-121-rac2
```

If the output is different (i.e. the OGGRAC instance is in fact running) you should shut it down using the command:

```
[OS prompt] $ svrctl stop database -d OGGRAC -f
```

10. Startup mount the OGGRAC1 instance (the instance local to the ol6-121-rac1 node.) Use the svrctl command to startup the local instance, and use sqlplus to set the database in ARCHIVELOG mode:

```
[OS prompt] $ svrctl start instance -d OGGRAC -i OGGRAC1 -o
mount
[OS prompt] $ sqlplus / as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Sat May 9 23:39:02
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options

SQL> alter database archivelog;
Database altered.
SQL> alter database open;
Database altered.
```

```

SQL> archive log list
Database log mode Archive Mode
Automatic archival Enabled
Archive destination USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence 101
Next log sequence to archive 102
Current log sequence 102
SQL> exit
[OS prompt] $

```

11. In your environment, the log sequence number will be different, but the point is that you should see log sequence number as proof that the database is running in ARCHIVELOG mode. Use the `srvctl` utility to start `OGGRAC2`, the second instance of the `OGGRAC` database. Connect to the `OGGRAC` database by using `sqlplus` as the `system` user and make sure the database is in ARCHIVELOG mode:

```

[OS prompt] $ srvctl start instance -d OGGRAC -i OGGRAC2
[OS prompt] $ sqlplus sys/oracle@OGGRAC as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Sat May 9 23:51:26
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Sat May 09 2015 21:31:59 -04:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options

SQL> select log_mode from v$database;

LOG_MODE

ARCHIVELOG
SQL>

```

12. Add supplemental logging and force logging for the `OGGRAC` database; force a log file switch:

```

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

SQL> ALTER DATABASE FORCE LOGGING;
Database altered.

SQL> ALTER SYSTEM SWITCH LOGFILE;
System altered.

```

```

SQL> alter system set streams_pool_size= '400M' scope= spfile;
System altered.

SQL> alter system set enable_goldengate_replication = TRUE scope
= both;
System altered.
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options

```

13. Use the `srvctl` utility to stop and restart the database after parameter changes and to verify that the database is running::

```

[OS prompt] $ srvctl stop database -d OGGRAC -f
[OS prompt] $ srvctl start database -d OGGRAC
[OS prompt] $ srvctl status database -d OGGRAC
Instance OGGRAC1 is running on node ol6-121-rac1
Instance OGGRAC2 is running on node ol6-121-rac2

```

14. Use `sqlplus` to connect to OGGRAC as sysdba and verify that `STREAMS_POOL_SIZE` has been increased, and that `ENABLE_GOLDENGATE_REPLICATION` has been set to true:

```

[OS prompt] $ sqlplus sys/oracle@OGGRAC as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on Sat May 9 23:51:26
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Sat May 09 2015 21:31:59 -04:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
SQL> select * from V$SGAINFO;
NAME BYTES RES CON_ID

Fixed SGA Size 2923680 No 0
Redo Buffers 13852672 No 0
Buffer Cache Size 318767104 Yes 0
In-Memory Area Size 0 No 0
Shared Pool Size 251658240 Yes 0
Large Pool Size 150994944 Yes 0
Java Pool Size 16777216 Yes 0
Streams Pool Size 419430400 Yes 0
Shared IO Pool Size 33554432 Yes 0

```

```

Data Transfer Cache Size 0 Yes 0
Granule Size 16777216 No 0
Maximum SGA Size 1174405120 No 0
Startup overhead
 in Shared Pool 186537848 No 0
Free SGA Memory Available 0 0
14 rows selected.

SQL> show parameter enable_goldengate
NAME TYPE VALUE

enable_goldengate_replication boolean TRUE
SQL>

```

15. Create the OGGDATA tablespace and set it as the default database tablespace:

```

SQL> create tablespace oggdata datafile size 500M;
Tablespace created.
SQL> alter database default tablespace oggdata;
Database altered.
SQL>

```

16. Create the Oracle GoldenGate administrative user (OGG\_ADMIN) and give it all necessary privileges:

```

SQL> create user OGG_ADMIN identified by oracle default
tablespace oggdata temporary tablespace temp;
User created.
SQL> grant dba to ogg_admin;
Grant succeeded.
SQL> exec
dbms_goldengate_auth.grant_admin_privilege('ogg_admin',
'CAPTURE', TRUE);
PL/SQL procedure successfully completed.
SQL>

```

17. Create the unprivileged replication source user (src\_user).

```

SQL> create user src_user identified by oracle default
tablespace oggdata temporary tablespace temp;
User created.
SQL> grant connect,resource,unlimited tablespace to src_user;
Grant succeeded.
SQL>

```

18. Exit sqlplus:

```

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,

```

```
Advanced Analytics and Real Application Testing options
[OS prompt] $
```

### Determining the IP address of the host computer

19. On the host computer (and NOT in any of the guest virtual machines), open a terminal shell window and enter the `ifconfig` command to display the network interface configuration. You will most likely find entries for `vboxnet0` and `vboxnet1` (the VirtualBox virtual network stacks). You should also find the `lo` (loopback) interface (with IP address 127.0.0.1). You are interested in determining the IPv4 address of your host computer. Its interface is either `em1` or `eth0` (if you are in doubt, ask your instructor). For example:

```
[OS prompt] $ ifconfig
em1 Link encap:Ethernet HWaddr 34:17:EB:99:95:11
 inet addr:10.150.30.98 Bcast:10.150.30.255
 Mask:255.255.255.0
 [...several omitted lines...]
 Interrupt:20 Memory:f7c00000-f7c20000

lo Link encap:Local Loopback
 inet addr:127.0.0.1 Mask:255.0.0.0
 [...several omitted lines...]

vboxnet0 Link encap:Ethernet HWaddr 0A:00:27:00:00:00
 inet addr:192.168.56.1 Bcast:192.168.56.255
 Mask:255.255.255.0
 [...several omitted lines...]

vboxnet1 Link encap:Ethernet HWaddr 0A:00:27:00:00:01
 inet addr:192.168.57.1 Bcast:192.168.57.255
 Mask:255.255.255.0
 [...several omitted lines...]
[OS prompt] $
```

In the example above the IP address is 10.150.30.98, which is the value after the `inet addr` label for the `em1` interface. Usually, but not always, the IP address that you are looking for is listed in the first interface, like in the example above. Again, if you have any doubts, ask your instructor.

Write down your IP address. You must create entries in the `/etc/hosts` files on both virtual machines to set up Oracle GoldenGate replication between the OGGRAC database (the replication source) and the `CDB12C` database (the replication target).

20. Click the "RAC 1" terminal window and use the `su` command to connect as root. Edit the `/etc/hosts` file and insert an entry for `ogg_host` (the host computer) with its associated IP address, which you determined in step 19:

```
[OS prompt] $ su -
Password: *****
[OS prompt] # vi /etc/hosts
127.0.0.1 localhost localhost.localdomain
#:1 localhost localhost.localdomain localhost6
localhost6.localdomain6
192.168.56.101 ol6-121-rac1.localdomain ol6-121-rac1
192.168.56.102 ol6-121-rac2.localdomain ol6-121-rac2
Private
192.168.57.101 ol6-121-rac1-priv.localdomain ol6-121-rac1-priv
192.168.57.102 ol6-121-rac2-priv.localdomain ol6-121-rac2-priv
Virtual
192.168.56.103 ol6-121-rac1-vip.localdomain ol6-121-rac1-vip
192.168.56.104 ol6-121-rac2-vip.localdomain ol6-121-rac2-vip
10.150.30.98 ogg_host.localdomain ogg_host
```

21. Obviously, replace `10.150.30.98` with the IP address you found in step 19. Save the `hosts` file and exit the editor. Try the `ping` command to test the newly added entry in the `hosts` file; use the `CTRL + C` combination to stop the output:

```
[OS prompt] # ping ogg_host
PING ogg_host.localdomain (10.150.30.98) 56(84) bytes of data.
64 bytes from ogg_host.localdomain (10.150.30.98): icmp_seq=1
ttl=64 time=0.120 ms
64 bytes from ogg_host.localdomain (10.150.30.98): icmp_seq=2
ttl=64 time=0.381 ms
^C
```

22. Enter `exit` to disconnect from the root user:

```
[OS prompt] # exit
[OS prompt] $
```

23. Repeat the steps for the second virtual machine (`ol6-121-rac2`). Click the "RAC 2" terminal window and use the `su` command to connect as root. Edit `/etc/hosts` and add the entry for `ogg_host` (as you did in step 20). Save the `hosts` file and exit the editor. When at the OS prompt, use the `ping` command to test the `ogg_host` entry. Exit the root connection also when operating in the "RAC 2" terminal shell.

This completes Practice 12-1. Continue with Practice 12-2.

## Practice 12-2: Configuring Oracle GoldenGate Replication

### Overview

In this practice, you perform the usual steps necessary to configure data extraction and data delivery in order to replicate from the OGGRAC database (the source) to the CDB12C database (the target).

### Assumptions

Practice 12-1 has been successfully completed.

### Tasks

#### Configuring the OGGRAC database for replication

1. Click the "RAC 1" terminal window. Connect to the OGGRAC database by using `sqlplus` as the `src_user` user. Create a basic infrastructure that is able to generate data by storing rows selected from the `ALL_OBJECTS` system view into a local table. This entails creating an empty table modeled after `ALL_OBJECTS`, adding a numeric column to it, and setting it as the primary key; you also create a sequence and a `BEFORE INSERT` trigger, which populates the primary key column from sequence-generated values.

```
[OS prompt] $ sqlplus src_user/oracle@OGGRAC
SQL*Plus: Release 12.1.0.2.0 Production on Sun May 10 07:08:00 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, Real Application Clusters, Automatic Storage
Management, OLAP,
Advanced Analytics and Real Application Testing options

SQL> create table ora_all_obj as select * from all_objects where
 rownum < 1;
Table created.
SQL> alter table ora_all_obj add obj_seq_id number not null ;
Table altered.
SQL> ALTER TABLE ora_all_obj ADD CONSTRAINT pk_ora_all_obj PRIMARY
KEY(obj_seq_id) ;
Table altered.
SQL> create sequence obj_seq start with 1 increment by 1 nomaxvalue;
Sequence created.
SQL>
```

2. Create the BEFORE INSERT trigger and exit sqlplus:

```
SQL> CREATE TRIGGER ora_obj_trg
 BEFORE INSERT ON ora_all_obj
 FOR EACH ROW
BEGIN
 :new.obj_seq_id := obj_seq.nextval;
END;
/

Trigger created.
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
[OS prompt] $
```

With this minimal infrastructure, you can now generate multiple rows simply by inserting into the ORA\_ALL\_OBJ table all rows from ALL\_OBJECTS.

3. On the host computer (NOT on the VM), you configure the replication target database. Open a terminal shell window on the host and use sqlplus to connect to the trg\_user on the ogg2 pluggable database hosted by CDB12C.

```
[OS prompt] $ sqlplus trg_user/oracle@ogg2
SQL*Plus: Release 12.1.0.2.0 Production on Sun May 10 12:47:21
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options
SQL>
```

4. Create the mirror table to the table just created on OGGRAC, and then exit sqlplus:

```

CREATE TABLE TRG_USER.ORA_ALL_OBJ
 ("OWNER" VARCHAR2(128 BYTE) NOT NULL ENABLE,
 "OBJECT_NAME" VARCHAR2(128 BYTE) NOT NULL ENABLE,
 "SUBOBJECT_NAME" VARCHAR2(128 BYTE),
 "OBJECT_ID" NUMBER NOT NULL ENABLE,
 "DATA_OBJECT_ID" NUMBER,
 "OBJECT_TYPE" VARCHAR2(23 BYTE),
 "CREATED" DATE NOT NULL ENABLE,
 "LAST_DDL_TIME" DATE NOT NULL ENABLE,
 "TIMESTAMP" VARCHAR2(19 BYTE),
 "STATUS" VARCHAR2(7 BYTE),
 "TEMPORARY" VARCHAR2(1 BYTE),
 "GENERATED" VARCHAR2(1 BYTE),
 "SECONDARY" VARCHAR2(1 BYTE),
 "NAMESPACE" NUMBER NOT NULL ENABLE,
 "EDITION_NAME" VARCHAR2(128 BYTE),
 "SHARING" VARCHAR2(13 BYTE),
 "EDITIONABLE" VARCHAR2(1 BYTE),
 "ORACLE_MAINTAINED" VARCHAR2(1 BYTE),
 "OBJ_SEQ_ID" NUMBER NOT NULL ENABLE,
 PRIMARY KEY ("OBJ_SEQ_ID")
 USING INDEX PCTFREE 10 INITTRANS 2 MAXTRANS 255
 TABLESPACE "OGGDATA" ENABLE
) PCTFREE 10 PCTUSED 40 INITTRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
 TABLESPACE OGGDATA ;
SQL> exit
[OS prompt] $

```

5. Click the "RAC 1" terminal window. If sqlplus is still running, exit from it. The dbfs\_mount resource you defined in Practice 4 is defined as "local\_resource." You must change that and define dbfs\_mount as "cluster\_resource." The reason for using cluster\_resource is so the resource can only be running on a single node, preventing the accidental mounting of DBFS from concurrent nodes. Use the grid\_env alias to set the environment variables that are needed for the grid infrastructure, and check the status of the dbfs\_mount resource. If it is "ONLINE," first stop the dbfs\_mount resource. If it is already "OFFLINE," simply delete it by using the crsctl utility:

```

[OS prompt] $ grid_env
[OS prompt] $ crsctl stat resource dbfs_mount
NAME=dbfs_mount
TYPE=local_resource
TARGET=ONLINE , ONLINE
STATE=ONLINE on ol6-121-rac1, ONLINE on ol6-121-rac2

```

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

```
[OS prompt] $ crsctl stop resource dbfs_mount
CRS-2673: Attempting to stop 'dbfs_mount' on 'ol6-121-rac2'
CRS-2673: Attempting to stop 'dbfs_mount' on 'ol6-121-rac1'
CRS-2677: Stop of 'dbfs_mount' on 'ol6-121-rac2' succeeded
CRS-2677: Stop of 'dbfs_mount' on 'ol6-121-rac1' succeeded
[OS prompt] $ crsctl delete resource dbfs_mount
[OS prompt] $
```

6. Navigate to the /home/oracle directory where you have the add-dbfs-resource.sh file. Edit that file and change the crsctl -type to cluster\_resource. Save the shell file and execute it, to re-define dbfs\_mount as cluster\_resource:

```
[OS prompt] $ cd ~
[OS prompt] $ vi add-dbfs-resource.sh
start script add-dbfs-resource.sh
#!/bin/bash
ACTION_SCRIPT=/u01/app/12.1.0.2/grid/crs/script/mount-dbfs.sh
RESNAME=dbfs_mount
DBNAME=OGGRAC
DBNAMEL=`echo $DBNAME | tr A-Z a-z`
ORACLE_HOME=/u01/app/12.1.0.2/grid
PATH=$ORACLE_HOME/bin:$PATH
export PATH ORACLE_HOME
crsctl add resource $RESNAME \
 -type cluster_resource \
 -attr "ACTION_SCRIPT=$ACTION_SCRIPT, \
 CHECK_INTERVAL=30,RESTART_ATTEMPTS=10, \
 START_DEPENDENCIES='hard(ora.$DBNAMEL.db) pullup(ora.$DBNAMEL.db)', \
 STOP_DEPENDENCIES='hard(ora.$DBNAMEL.db)', \
 SCRIPT_TIMEOUT=300"
end script add-dbfs-resource.sh
```

7. Save the file and exit the editor. Then, execute the file. Use crsctl status to verify the dbfs\_mount resource definition:

```
[OS prompt] $./add-dbfs-resource.sh
[OS prompt] $ crsctl status resource dbfs_mount
NAME=dbfs_mount
TYPE=cluster_resource
TARGET=OFFLINE
STATE=OFFLINE
[OS prompt] $
```

8. Start the dbfs\_mount resource by using crsctl:

```
[OS prompt] $ crsctl start resource dbfs_mount
CRS-2672: Attempting to start 'dbfs_mount' on 'ol6-121-rac2'
CRS-2676: Start of 'dbfs_mount' on 'ol6-121-rac2' succeeded
```

**Important:** This step determines the mount point and the node where to configure Oracle GoldenGate next. In this case, the Oracle Clusterware infrastructure chose to mount the DBFS file system on the rac2 node. So, the Oracle GoldenGate software setup will be performed on the ol6-121-rac2 node. Had Oracle Clusterware chosen, on the other hand, to mount the DBFS file system on rac1, you would now proceed to set up Oracle GoldenGate on the ol6-121-rac1 node.

9. Choose either the "RAC 1" or "RAC 2" terminal window, according to where Oracle Clusterware mounted the DBFS file system. If the DBFS file system was mounted on the ol6-121-rac2 node, use the "RAC 2" terminal window. Navigate to the /u03/ogg/ogg\_src directory and move several Oracle GoldenGate directories to the DBFS file system. Create symbolic links to the moved directories so that Oracle GoldenGate can still find those directories:

```
[OS prompt] $ cd /u03/ogg/ogg_src
[OS prompt] $ mv dirprm /oracle/dbfs_direct/FS1/
[OS prompt] $ mv dirchk /oracle/dbfs_direct/FS1/
[OS prompt] $ mv dirpcs /oracle/dbfs_direct/FS1/
[OS prompt] $ mv dirtmp /oracle/dbfs_direct/FS1/
[OS prompt] $ mv dircrd /oracle/dbfs_direct/FS1/
[OS prompt] $ mv dirwlt /oracle/dbfs_direct/FS1/
[OS prompt] $ ln -s /oracle/dbfs_direct/FS1/dirprm dirprm
[OS prompt] $ ln -s /oracle/dbfs_direct/FS1/dirchk dirchk
[OS prompt] $ ln -s /oracle/dbfs_direct/FS1/dirpcs dirpcs
[OS prompt] $ ln -s /oracle/dbfs_direct/FS1/dirtmp dirtmp
[OS prompt] $ ln -s /oracle/dbfs_direct/FS1/dircrd dircrd
[OS prompt] $ ln -s /oracle/dbfs_direct/FS1/dirwlt dirwlt
```

10. Create the dirdat and BR directories under /oracle/dbfs\_direct/FS1/:

```
[OS prompt] $ mkdir /oracle/dbfs_direct/FS1/dirdat
[OS prompt] $ mkdir /oracle/dbfs_direct/FS1/BR
```

11. Use the db\_env alias to set the required environment variables and launch GGSCI. Create a Wallet and a credential store, and then add the ogg\_admin alias for the OGG\_ADMIN@OGGRAC user to the credential store; log in to OGGRAC by using the newly created alias:

```
[OS prompt] $ db_env
[OS prompt] $ ggsci
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
10:21:34
Operating system character set identified as UTF-8.
```

```

Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
rights reserved.

GGSCI ([HOST]) > create wallet
Created wallet at location 'dirwlt'.
Opened wallet at location 'dirwlt'.
GGSCI ([HOST]) > Add CredentialStore
Credential store created in ./dircrd/.
GGSCI ([HOST]) > Alter CredentialStore Add User OGG_ADMIN@OGGRAC
password oracle alias ogg_admin
Credential store in ./dircrd/ altered.
GGSCI ([HOST]) > dblogin useridalias ogg_admin
Successfully logged into database.
GGSCI ([HOST] as OGG_ADMIN@OGGRAC2) >

```

12. Edit the parameter file for the Oracle GoldenGate manager and make sure to set the auto start and auto restart settings for all replication groups:

```

GGSCI ([HOST] as OGG_ADMIN@OGGRAC2) > edit param mgr
PORT 7809
DynamicPortList 20000-20099
PurgeOldExtracts /oracle/dbfs_direct/FS1/dirdat/*,MinKeepHours 2
AUTOSTART ER *
AUTORESTART ER *

```

13. Stop and restart the manager so that it loads the new parameters. If the manager was already stopped, GGSCI will display a message ("Manager already started") which you can disregard:

```

GGSCI ([HOST] as OGG_ADMIN@OGGRAC2) > stop mgr
Manager stopped.
GGSCI ([HOST] as OGG_ADMIN@OGGRAC2) > start mgr
Manager started.

```

14. Add the SCHEMATRANLATA definition for the source schema (`src_user`):

```

GGSCI ([HOST] as OGG_ADMIN@OGGRAC2) > add schematrandata
src_user ALLCOLS
2015-05-11 11:10:39 INFO OGG-01788 SCHEMATRANLATA has been
added on schema src_user.
2015-05-11 11:10:39 INFO OGG-01976 SCHEMATRANLATA for
scheduling columns has been added on schema src_user.
2015-05-11 11:10:40 INFO OGG-01977 SCHEMATRANLATA for all
columns has been added on schema src_user.

```

15. You must now prepare the GGSCI environment on the other cluster node. In the earlier steps, if you configured the data extraction environment on the "RAC 2" node because Oracle Clusterware started the dbfs\_mount file system on ol6-121-rac2, you must now click the "RAC 1" terminal window and configure the file system links on that node. Click the "RAC 1" terminal window and navigate to the /u03/ogg/ogg\_src directory. Remove the dircrd, dirchk, dirpcs, dirprm, dirtmp, and dirwlt directories and set up logical links to the dbfs\_mount file system instead:

```
[OS prompt] $ cd /u03/ogg/ogg_src
[OS prompt] $ rm -rf dircrd dirchk dirpcs dirprm dirtmp dirwlt
[OS prompt] $ ln -s /oracle/dbfs_direct/FS1/dirprm dirprm
[OS prompt] $ ln -s /oracle/dbfs_direct/FS1/dirchk dirchk
[OS prompt] $ ln -s /oracle/dbfs_direct/FS1/dirpcs dirpcs
[OS prompt] $ ln -s /oracle/dbfs_direct/FS1/dirtmp dirtmp
[OS prompt] $ ln -s /oracle/dbfs_direct/FS1/dircrd dircrd
[OS prompt] $ ln -s /oracle/dbfs_direct/FS1/dirwlt dirwlt
```

**Note:** The links that you just created appear as broken links if you enter ls -l, but that is expected, because the dbfs\_mount resource did not mount the /oracle/dbfs\_direct file system on the node where you are creating the symbolic links. The links will not be broken when Oracle Clusterware relocates the /oracle/dbfs\_direct file system after either a role transition (in the case of a Data Guard configuration) or a node failure (in the case of a RAC environment).

drwxr-x---	2 oracle oinstall	4096 Apr 2 20:25	dirbdb	
lrwxrwxrwx	1 oracle oinstall	30 May 14 00:00	dirchk	-> /oracle/dbfs_direct/FS1/dirchk
lrwxrwxrwx	1 oracle oinstall	30 May 14 00:00	dircrd	-> /oracle/dbfs_direct/FS1/dircrd
drwxr-x---	2 oracle oinstall	4096 Apr 2 20:25	dirdat	
drwxr-x---	2 oracle oinstall	4096 Apr 2 20:25	dirdef	
drwxr-x---	2 oracle oinstall	4096 Apr 2 20:25	dirdmp	
drwxr-xr-x	2 oracle oinstall	4096 Apr 2 20:24	dirjar	
drwxr-xr-x	2 oracle oinstall	4096 Apr 2 20:24	dirout	
lrwxrwxrwx	1 oracle oinstall	30 May 14 00:00	dirpcs	-> /oracle/dbfs_direct/FS1/dirpcs
lrwxrwxrwx	1 oracle oinstall	30 May 14 00:00	dirprm	-> /oracle/dbfs_direct/FS1/dirprm
drwxr-x---	2 oracle oinstall	4096 Apr 2 20:25	dirprt	
drwxr-x---	2 oracle oinstall	4096 Apr 2 20:25	dirsrl	
lrwxrwxrwx	1 oracle oinstall	30 May 14 00:01	dirtmp	-> /oracle/dbfs_direct/FS1/dirtmp
lrwxrwxrwx	1 oracle oinstall	30 May 14 00:01	dirwlt	-> /oracle/dbfs_direct/FS1/dirwlt
drwxr-xr-x	5 oracle oinstall	4096 Apr 2 20:23	dirwww	

The links are shown in red because they appear as broken links. Do not worry about it.

16. Click the "RAC 2" terminal window (where GGSCI is running), register the EXRAC Extract group with the OGGRAC database, and edit the EXRAC parameters:

```
GGSCI ([HOST] as OGG_ADMIN@OGGRAC2) > register extract exrac
database
```

```
Extract EXRAC successfully registered with database at SCN
5187685.
```

```
GGSCI ([HOST] as OGG_ADMIN@OGGRAC2) > edit param exrac
```

```

Extract exrac
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV(ORACLE_SID = "OGGRAC")
ExtTrail /oracle/dbfs_direct/FS1/dirdat/ea
UserIdAlias ogg_admin
BR BRDIR /oracle/dbfs_direct/FS1/BR
LOGALLSUPCOLS
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 128)
UPDATERECORDFORMAT COMPACT
Table src_user.*;

```

**Note:** You specify the Bounded Recovery directory by using the BR BRDIR parameter.

- Save the EXRAC parameter file and exit the editor. Add the Extract group and its EXTTRAIL:

```

GGSCI ([HOST] as OGG_ADMIN@OGGRAC2) > Add Extract exrac,
Integrated TranLog, Begin Now
EXTRACT added.

GGSCI ([HOST] as OGG_ADMIN@OGGRAC2) > Add ExtTrail
/oracle/dbfs_direct/FS1/dirdat/ea, Extract exrac, Megabytes 10
EXTTRAIL added.

```

- Create the Data Pump Extract DPRAC:

```
GGSCI ([HOST] as OGG_ADMIN@OGGRAC2) > edit param dprac
```

```

Extract dprac
SETENV(ORACLE_HOME="/u01/app/oracle/product/12.1.0.2/db_1")
SETENV (ORACLE_SID="OGGRAC")
UserIdAlias ogg_admin
RMTHOST ogg_host, MGRPORT 7909
RmtTrail ./dirdat/pe
Passthru
Table src_user.*;

```

- Save the parameter file and exit the editor. Before adding the Data Pump Extract, make sure that the Oracle GoldenGate replication target environment is running on the host. On the host computer (not in any of the guest VMs), start a terminal shell window and use the oggtrg alias to navigate to the /u03/ogg/ogg\_trg directory. Erase the trail files stored in the dirdat directory. This is an important step. You do not want to leave around trail files which could interfere with the trail files you are going to create in this practice. Launch GGSCI and enter the "info all" command to verify that the manager is running. If it is not running, start the manager:

```

[OS prompt] $ oggtrg
[OS prompt] $ pwd
/u03/ogg/ogg_trg
[OS prompt] $ rm -f ./dirdat/*
[OS prompt] $ ggsci
Oracle GoldenGate Command Interpreter for Oracle

```

```

Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
10:21:34
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
rights reserved.

GGSCI ([HOST]) > info all

```

Start the manager if it is not running, but leave the terminal shell window open.

- Click the "RAC 2" window and add the Data Pump Extract and the remote trail:

```

GGSCI ([HOST] as OGG_ADMIN@OGGRAC2) > Add Extract dprac,
ExtTrailSource /oracle/dbfs_direct/FS1/dirdat/ea
EXTRACT added.

GGSCI ([HOST] as OGG_ADMIN@OGGRAC2) > Add RmtTrail ./dirdat/pe,
Extract dprac, Megabytes 10

```

- Because the Oracle GoldenGate manager is configured to both AUTOSTART and AUTORESTART Extract and Replicat groups, you do not need to explicitly start the Extract groups. Simply enter the "info all" command and verify that EXRAC and DPRAC are running:

```

GGSCI ([HOST] as OGG_ADMIN@OGGRAC2) 31> info all

Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING DPRAC 00:00:00 00:00:08
EXTRACT RUNNING EXRAC 00:00:06 00:00:02

```

If any of the Extract groups was not automatically started by the manager, and its status is STOPPED, simply start it manually using the START <group name> command. If none of the Extract group is in the RUNNING state, use "start ER \*" to start them both.

- Enable the terminal shell window you opened in step 18 (connected on the host computer), where GGSCI is connected to the target replication system. Log in to the target database (alias ogg\_repl) and edit the parameter file for the RERAC replicat:

```

GGSCI ([HOST]) > dblogin useridalias ogg_repl
Successfully logged into database OGG2.

GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > edit param rerac

Replicat rerac
AssumeTargetDefs
DiscardFile ./dirrpt/rerac.dsc, Purge
UserIDAlias ogg_repl
Map src_user.* , target ogg2.trg_user.*;

```

23. Save the parameter file and exit the editor. Add the Replicat, connecting it to the exttrail pe and start the Replicat:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > add replicat rerac,
integrated exttrail ./dirdat/pe
REPLICAT (Integrated) added.
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > start rerac
Sending START request to MANAGER ...
REPLICAT RERAC starting
```

24. Make sure the Replicat group started without issues. Enter the "info all" command:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > info all

Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
REPLICAT RUNNING RERAC 00:00:00 00:00:10
```

25. Generate some activity on the OGGRAC database and verify that data is replicated. Click the "RAC 2" window and exit GGSCI. Connect to the OGGRAC database as src\_user and populate the ORA\_ALL\_OBJ table:

```
GGSCI ([HOST] as OGG_ADMIN@OGGRAC2) > exit
[OS prompt] $ sqlplus src_user/oracle@OGGRAC
SQL*Plus: Release 12.1.0.2.0 Production on Mon May 11 17:51:10
2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Mon May 11 2015 09:49:26 -04:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 -
64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
SQL> INSERT INTO ORA_ALL_OBJ
(OWNER,OBJECT_NAME,SUBOBJECT_NAME,OBJECT_ID,DATA_OBJECT_ID,OBJEC
T_TYPE,CREATED,LAST_DDL_TIME,TIMESTAMP,STATUS,
TEMPORARY,GENERATED,SECONDARY,NAMESPACE,EDITION_NAME,SHARING,
EDITIONABLE,ORACLE_MAINTAINED) SELECT
OWNER,OBJECT_NAME,SUBOBJECT_NAME, OBJECT_ID, DATA_OBJECT_ID,
OBJECT_TYPE,CREATED,LAST_DDL_TIME,TIMESTAMP,STATUS,
TEMPORARY,GENERATED,SECONDARY,NAMESPACE,EDITION_NAME,SHARING,
EDITIONABLE, ORACLE_MAINTAINED FROM ALL_OBJECTS
WHERE ROWNUM < 10000;
9999 rows created.
SQL> commit;
```

26. Click the terminal shell windows where **GGSCI** is connected to the replication target environment and request statistics for the **RERAC** Replicat. Verify that the rows created on OGGRAC have been replicated:

```
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) > stats rerac
Sending STATS request to REPLICAT RERAC ...
Start of Statistics at 2015-05-11 21:55:04.

Integrated Replicat Statistics:
 Total transactions 1.00
 Redirected 0.00
 DDL operations 0.00
 Stored procedures 0.00
 Datatype functionality 0.00
 Event actions 0.00
 Direct transactions ratio 0.00%
Replicating from SRC_USER.ORA_ALL_OBJ to OGG2.TRG_USER.ORA_ALL_OBJ:
*** Total statistics since 2015-05-11 11:05:07 ***
 Total inserts 9999.00
 Total updates 0.00
 Total deletes 0.00
 Total discards 0.00
 Total operations 9999.00
[...many lines omitted...]
GGSCI ([HOST] as C##OGG_ADMIN@CDB12c/OGG2) >
```

This completes Practice 12-2. Continue with Practice 12-3.

## Practice 12-3: Configuring the Oracle Bundled Agent (XAG)

### Overview

In this practice, you use the `agctl` utility to configure the Oracle Bundled Agent so that Oracle GoldenGate becomes an Oracle Clusterware resource and, as such, it can survive role transitions and RAC node failures.

### Assumptions

Practices 12-1 and 12-2 have been successfully completed.

### Tasks

#### Configuring Oracle GoldenGate Virtual IP (VIP)

- Click the "RAC 2" terminal window and exit `sqlplus`. Set the Oracle Grid environment variables by executing the alias "grid\_env." You must configure a virtual IP for the Oracle GoldenGate resource, which must be in the subnet used by the Oracle Clusterware infrastructure for the RAC database. Use the `crsctl` utility to find out what the network segment is:

```
SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release
12.1.0.2.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic
Storage Management, OLAP,
Advanced Analytics and Real Application Testing options
[OS prompt] $ grid_env
[OS prompt] $ crsctl stat res -p | grep -ie .network -ie subnet
| grep -ie name -ie ora_subnet
NAME=ora.net1.network
USR_ORA_SUBNET=192.168.56.0
```

- The subnet is 192.168.56.0. Let us assign the IP address 192.168.56.200 to the Oracle GoldenGate resource. To accomplish this, you need to connect as the `root` user and use the `appvipcfg` utility to create a new VIP. Use the `su` command (but this time, do not type the dash character after `su`). You want to keep the environment set up as the `oracle` user):

```
[OS prompt] $ su
Password: *****
[OS prompt] #
```

- Source the `grid_env` file located in the `/home/oracle` directory to set up the required environment variables. Use the `appvipcfg` utility to configure the new VIP:

```
[OS prompt] # appvipcfg create -network=1 -ip=192.168.56.200 -
vipname=rac-ggate-vip -user=root -group=oinstall
[OS prompt] #
```

4. Use the `crsctl` utility to set permission for the `oracle` user to interact with the newly created VIP:

```
[OS prompt] # crsctl setperm resource rac-ggate-vip -u
user:oracle:r-x
```

5. Log out from root, and as the `oracle` user, use `crsctl` to start the VIP resource:

```
[OS prompt] # exit
[OS prompt] $ crsctl start resource rac-ggate-vip
CRS-2672: Attempting to start 'rac-ggate-vip' on 'ol6-121-rac2'
CRS-2676: Start of 'rac-ggate-vip' on 'ol6-121-rac2' succeeded
```

### Configuring the bundled agent with agctl

6. To use `agctl`, you must find out how several Clusterware components are named. You need to know:

- RAC node names
- File system name
- Database name
- VIP name
- Oracle GoldenGate home directory
- Oracle Home
- Oracle GoldenGate Extract group names

Use the `crsctl` utility to find out the database name:

```
[OS prompt] $ crsctl stat res -w "TYPE = ora.database.type"
NAME=ora.oggrac.db
TYPE=ora.database.type
TARGET=ONLINE , ONLINE
STATE=ONLINE on ol6-121-rac1, ONLINE on ol6-121-rac2
```

7. Find out the RAC node name by using the `olsnodes` utility:

```
[OS prompt] $ olsnodes
ol6-121-rac1
ol6-121-rac2
```

You now have all the required information:

Clusterware Component	Values
RAC nodes	ol6-121-rac1, ol6-121-rac2
File system name	dbfs_mount
Database name	ora.oggrac.db
Oracle GoldenGate home directory	/u03/ogg/ogg_src
Oracle Home	/u01/app/oracle/product/12.1.0.2/db_1
VIP name	rac-ggate-vip
Oracle GoldenGate Extract group names	EXRAC, DPRAC

8. You can now use the agctl utility to configure the bundled agents:

```
[OS prompt] $ /u01/app/oracle/product/xag/bin/agctl add
goldengate rac_ggate --gg_home /u03/ogg/ogg_src --instance_type
source --nodes ol6-121-rac1,ol6-121-rac2 --vip_name rac-ggate-
vip --filesystems dbfs_mount --databases ora.oggrac.db --
oracle_home /u01/app/oracle/product/12.1.0.2/db_1 --
monitor_extracts EXRAC,DPRAC --critical_extracts EXRAC,DPRAC
[OS prompt] $
```

**Note:** There are no spaces around the comma that separates node names and Extract group names. Leaving a space provokes a syntax error.

9. The agctl utility does not give any feedback. Use agctl again with the "config" parameter to assess the configuration entered in the previous step:

```
[OS prompt] $ /u01/app/oracle/product/xag/bin/agctl config
goldengate rac_ggate
GoldenGate location is: /u03/ogg/ogg_src
GoldenGate instance type is: source
Configured to run on Nodes: ol6-121-rac1 ol6-121-rac2
ORACLE_HOME location is: /u01/app/oracle/product/12.1.0.2/db_1
Databases needed: ora.oggrac.db
File System resources needed: dbfs_mount
Extracts to monitor: EXRAC,DPRAC
Replicats to monitor:
Critical extracts: EXRAC,DPRAC
Critical replicats:
Autostart on DataGuard role transition to PRIMARY: no
Autostart JAgent: no
```

**Note:** The Data Guard role transition is not enabled as you are not using Data Guard. To enable role transition, there is an additional parameter to agctl, dataguard\_autostart, which must be set to "yes."

10. Before starting the goldengate resource, you must stop GoldenGate. Navigate to the /u03/ogg/ogg\_src directory (if you are not already in it) and launch GGSCI. Stop both Extract groups and stop the manager. Then exit GGSCI:

```
[OS prompt] $ cd /u03/ogg/ogg_src
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
10:21:34
Operating system character set identified as UTF-8.
Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
rights reserved.
```

```
GGSCI ([HOST]) > stop ER * !
Sending STOP request to EXTRACT EXRAC ...
Request processed.
Sending STOP request to EXTRACT DPRAC ...
Request processed.
GGSCI ([HOST]) > stop mgr !
Sending STOP request to MANAGER ...
Request processed.
Manager stopped.
GGSCI ([HOST]) > exit
[OS prompt] $
```

- Start the goldengate resource by using the agctl utility:

```
[OS prompt] $ /u01/app/oracle/product/xag/bin/agctl start
goldengate rac_ggate
[OS prompt] $
```

- It takes some time for Oracle Clusterware to perform all the duties to start the goldengate resource. If you invoke agctl with the "status" parameter, it is likely that the answer will be "INTERMEDIATE" until it displays "is running":

```
[OS prompt] $ /u01/app/oracle/product/xag/bin/agctl status
goldengate rac_ggate
Goldengate instance 'rac_ggate' INTERMEDIATE on ol6-121-rac2
[OS prompt] $ /u01/app/oracle/product/xag/bin/agctl status
goldengate rac_ggate
Goldengate instance 'rac_ggate' INTERMEDIATE on ol6-121-rac2
```

- After a while, you will get the "is running" status:

```
[OS prompt] $ /u01/app/oracle/product/xag/bin/agctl status
goldengate rac_ggate
Goldengate instance 'rac_ggate' is running on ol6-121-rac2
```

**Note:** The Oracle Clusterware services could start the Oracle GoldenGate resource on the ol6-121-rac1 node, rather than ol6-121-rac2. That is normal behavior. If that happens, simply replace all occurrences of rac2 with rac1 in the next steps.

- Use the crsctl utility to visualize the status of your cluster:

[OS prompt]\$ crsctl stat res -t				
Name	Target	State	Server	State details
<hr/> Local Resources <hr/>				
ora.DATA.dg	ONLINE	ONLINE	ol6-121-rac1	STABLE
	ONLINE	ONLINE	ol6-121-rac2	STABLE
ora.LISTENER.				
lsnr	ONLINE	ONLINE	ol6-121-rac1	STABLE
	ONLINE	ONLINE	ol6-121-rac2	STABLE

ora.asm	ONLINE	ONLINE	ol6-121-rac1	Started, STABLE
	ONLINE	ONLINE	ol6-121-rac2	Started, STABLE
ora.net1.				
network	ONLINE	ONLINE	ol6-121-rac1	STABLE
	ONLINE	ONLINE	ol6-121-rac2	STABLE
ora.ons	ONLINE	ONLINE	ol6-121-rac1	STABLE
	ONLINE	ONLINE	ol6-121-rac2	STABLE
<hr/>				
Cluster Resources				
<hr/>				
dbfs_mount 1	ONLINE	ONLINE	ol6-121-rac2	STABLE
ora.LISTENER_				
SCAN1.lsnr 1	ONLINE	ONLINE	ol6-121-rac2	STABLE
ora.LISTENER_				
SCAN2.lsnr 1	ONLINE	ONLINE	ol6-121-rac1	STABLE
ora.LISTENER_				
SCAN3.lsnr 1	ONLINE	ONLINE	ol6-121-rac1	STABLE
ora.MGMTLSNR 1	ONLINE	ONLINE	ol6-121-rac 169.254.109.232 192.168.57.101	, STABLE
ora.cvu 1	ONLINE	ONLINE	ol6-121-rac1	STABLE
ora.mgmtdb 1	ONLINE	ONLINE	ol6-121-rac1	Open, STABLE
ora.oc4j 1	ONLINE	ONLINE	ol6-121-rac1	STABLE
ora.oggrac.db1	ONLINE	ONLINE	ol6-121-rac1	Open, STABLE
	2	ONLINE	ONLINE	ol6-121-rac2 Open, STABLE
ora.ol6-121-				
rac1.vip 1	ONLINE	ONLINE	ol6-121-rac1	STABLE
ora.ol6-121-				
rac2.vip 1	ONLINE	ONLINE	ol6-121-rac2	STABLE
ora.scan1				
.vip 1	ONLINE	ONLINE	ol6-121-rac2	STABLE
ora.scan2				
.vip 1	ONLINE	ONLINE	ol6-121-rac1	STABLE
ora.scan3				
.vip 1	ONLINE	ONLINE	ol6-121-rac1	STABLE
rac-ggate-vip				
	1	ONLINE	ONLINE	ol6-121-rac2 STABLE
xag.rac_ggate.goldengate				
	1	ONLINE	ONLINE	ol6-121-rac2 STABLE
<hr/>				
[OS prompt] \$				

15. In this example, the `xag.rac_ggate.goldengate` resource is running on the `ol6-121-rac2` node. You can use the "relocate" subcommand of `agctl` to stop Oracle GoldenGate from running on `ol6-121-rac2` and to move it to the `ol6-121-rac1` node. If the Oracle Clusterware services started the Oracle GoldenGate resource on the `ol6-121-rac1` node, simply relocate the Oracle GoldenGate service to the `ol6-121-rac2` node:

```
[OS prompt] $ /u01/app/oracle/product/xag/bin/agctl relocate
goldengate rac_ggate --node ol6-121-rac1
[OS prompt] $
```

**Note:** `agctl` does not provide feedback if everything is okay. If, however, something is wrong, `agctl` prints messages to the screen, for example:

```
CRS-2676: Start of 'rac-ggate-vip' on 'ol6-121-rac1' succeeded
CRS-2674: Start of 'dbfs_mount' on 'ol6-121-rac1' failed
CRS-2679: Attempting to clean 'dbfs_mount' on 'ol6-121-rac1'
```

If you receive error messages from `agctl`, try to resolve them. Ask your instructor if the problem persists.

16. After the `xag.rac_ggate.goldengate` resource has been relocated to the other node, verify the status of your cluster by using `crsctl` again:

<code>[OS prompt] \$ crsctl stat res -t</code>				
-----				
Name	Target	State	Server	State details
-----				
Local Resources				
-----				
ora.DATA.dg	ONLINE	ONLINE	ol6-121-rac1	STABLE
	ONLINE	ONLINE	ol6-121-rac2	STABLE
[... Many omitted lines...]				
rac-ggate-vip	1	ONLINE	ol6-121-rac1	STABLE
<b>xag.rac_ggate.goldengate</b>	1	ONLINE	ol6-121-rac1	STABLE

The output shows that the `goldengate` resource is now running on `ol6-121-rac1`.

17. You can also use GGSCI on the "RAC 1" node to assess the status of your data extraction. Click the "RAC 1" terminal window and navigate to the `/u03/ogg/ogg_src` directory. Launch GGSCI and enter the "info all" command. Then, exit GGSCI:

```
[OS prompt] $ cd /u03/ogg/ogg_src
[OS prompt] $ ggsci
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
10:21:34
Operating system character set identified as UTF-8.
```

```
Copyright (C) 1995, 2014, Oracle and/or its affiliates. All rights reserved.
GGSCI ([HOST]) > info all
Program Status Group Lag at Chkpt Time Since
Chkpt

MANAGER RUNNING
EXTRACT RUNNING DPRAC 00:00:00 00:00:01
EXTRACT RUNNING EXRAC 00:00:04 00:00:06
GGSCI ([HOST]) > exit
[OS prompt]$
```

This completes Practice 12-3. Continue with Practice 12-4.

## Practice 12-4: Failing a RAC Node to Test Oracle GoldenGate Survival

### Overview

In this practice, you intentionally provoke a node failure to test the survivability of the `xag.rac_ggate.goldengate` resource.

### Assumptions

Practices 12-1 through 12-3 have been successfully completed.

### Tasks

1. While testing the effectiveness of the Oracle Bundled Agent, you successfully relocated the `xag.rac_ggate.goldengate` resource from `ol6-121-rac2` to `ol6-121-rac1`. Your data extraction is now running on the `ol6-121-rac1` node. Generate some database activity to test replication. Click the "RAC 1" terminal window, use the `db_env` alias to set the database-related environment variables, and launch `sqlplus` to connect to the `OGGRAC` database as the `src_user` user. Insert 200 rows in the `ORA_ALL_OBJ` table and exit `sqlplus`:

```
[OS prompt]$ db_env
[OS prompt]$ sqlplus src_user/oracle@OGGRAC
SQL*Plus: Release 12.1.0.2.0 Production on Tue May 12 09:24:38 2015
Copyright (c) 1982, 2014, Oracle. All rights reserved.
Last Successful login time: Tue May 12 2015 05:06:57 -04:00
Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, Real Application Clusters, Automatic Storage
Management, OLAP,
Advanced Analytics and Real Application Testing options
SQL> INSERT INTO ORA_ALL_OBJ
(OWNER,OBJECT_NAME,SUBOBJECT_NAME,OBJECT_ID,DATA_OBJECT_ID,OBJEC
T_TYPE,CREATED,LAST_DDL_TIME,TIMESTAMP,STATUS,
TEMPORARY,GENERATED,SECONDARY,NAMESPACE,EDITION_NAME,SHARING,EDI
TIONABLE,ORACLE_MAINTAINED) SELECT
OWNER,OBJECT_NAME,SUBOBJECT_NAME,OBJECT_ID,DATA_OBJECT_ID,OBJECT
_TYPE,CREATED,LAST_DDL_TIME,TIMESTAMP,STATUS,
TEMPORARY,GENERATED,SECONDARY,NAMESPACE,EDITION_NAME,SHARING,EDI
TIONABLE,ORACLE_MAINTAINED FROM ALL_OBJECTS WHERE ROWNUM < 201;
200 rows created.
SQL> commit;
Commit complete.
SQL> exit
```

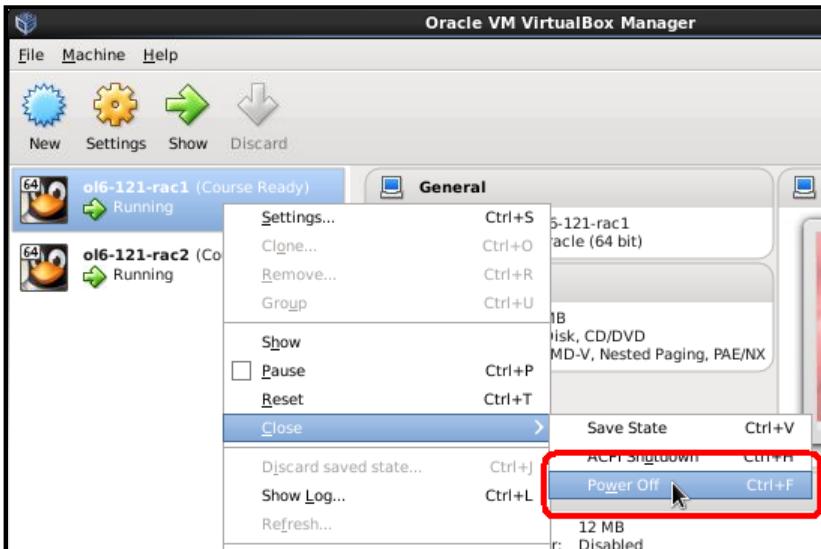
2. Make sure your default directory is still /u03/ogg/ogg\_src. If it is not, navigate to /u03/ogg/ogg\_src and launch GGSCI. Request statistics for the EXRAC Extract and exit GGSCI:

```
[OS prompt]$ pwd
/u03/ogg/ogg_src
[OS prompt]$ ggsci
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.1.0
OGGCORE_12.1.2.1.0_PLATFORMS_140727.2135.1_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Aug 7 2014
10:21:34
Operating system character set identified as UTF-8.
Copyright (C) 1995, 2014, Oracle and/or its affiliates. All
rights reserved.

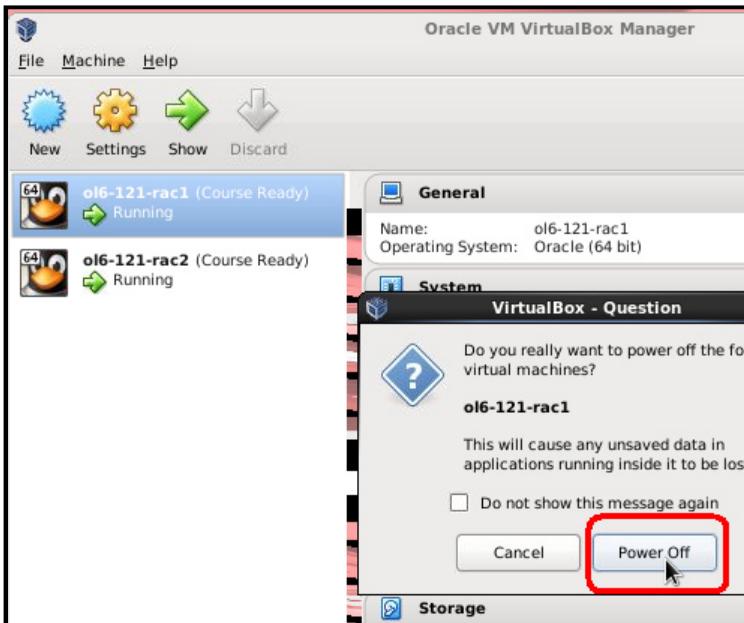
GGSCI ([HOST]) > stats exrac
Sending STATS request to EXTRACT EXRAC ...
Start of Statistics at 2015-05-12 09:29:21.
Output to /oracle/dbfs_direct/FS1/dirdat/ea:
Extracting from SRC_USER.ORA_ALL_OBJ to SRC_USER.ORA_ALL_OBJ:
*** Total statistics since 2015-05-12 09:28:57 ***
 Total inserts 200.00
 Total updates 0.00
 Total deletes 0.00
 Total discards 0.00
 Total operations 200.00
*** Daily statistics since 2015-05-12 09:28:57 ***
 Total inserts 200.00
 Total updates 0.00
 Total deletes 0.00
 Total discards 0.00
 Total operations 200.00
*** Hourly statistics since 2015-05-12 09:28:57 ***
 Total inserts 200.00
 Total updates 0.00
 Total deletes 0.00
 Total discards 0.00
 Total operations 200.00
*** Latest statistics since 2015-05-12 09:28:57 ***
 Total inserts 200.00
 Total updates 0.00
 Total deletes 0.00
 Total discards 0.00
 Total operations 200.00

End of Statistics.
GGSCI ([HOST]) > exit
[OS prompt]$
```

3. Click the terminal shell window running on the host computer that is connected to the replication target environment. Request statistics for the RERAC Replicat group ("stats rerac") and make sure that the 200 rows inserted on OGGRAC have been replicated.
4. Abruptly stop the ol6-121-rac1 virtual machine. You can use Oracle VirtualBox Manager to abruptly kill the ol6-121-rac1 node. Right-click the ol6-121-rac1 node in the left pane of the VirtualBox Manager and select "Close" > "Power Off":



5. In the confirmation dialog box that is displayed, click the "Power Off" button:



6. After a few seconds, the ol6-121-rac1 virtual machine disappears. Click the "RAC 2" terminal window and use the crsctl utility to check the status of the cluster. In particular, you should see the xag.rac\_ggate.goldengate resource ONLINE on the ol6-121-rac2 node. It is also likely that before reaching the ONLINE status, the xag.rac\_ggate.goldengate resource will go through the INTERMEDIATE state.
7. After you have verified that Oracle GoldenGate is running on the ol6-121-rac2 node, activate the "RAC 2" window, and insert another 200 rows in the ORA\_ALL\_OBJ table by

- using sqlplus. Exit sqlplus and use GGSCI on o16-121-rac2 (/u03/ogg/ogg\_src directory) to request statistics on the EXRAC and DPRAC Extract groups.
8. Finally, verify that data delivery worked. Click the terminal shell window running on the host computer that is connected to the replication target environment. Request statistics for the RERAC Replicat group ("stats rerac") and make sure that the additional 200 rows inserted on OGGRAC have been replicated.
  9. When you are satisfied that replication worked after surviving node failure, shut down the two virtual machines the same way you did in step 12-4.5.

This completes Practice 12-4.

This completes Practice 12 for *Oracle GoldenGate 12c: Advanced Configuration for Oracle*.

