Oracle® GoldenGate

Tutorial

for Oracle to Oracle

Version 10.4

November 2009

Oracle GoldenGate, Tutorial for Oracle to Oracle, version 10.4

Contents

Oracle to Oracle Replication

Extract, Replicat and associated utilities enable you to create, load and refresh one Oracle database to another Oracle database.

This tutorial provides a quick overview of Oracle to Oracle database replication using Extract and Replicat. For more detailed information, please consult the Oracle GoldenGate Administration Guide.

This tutorial may be read to get a general overview of how Extract and Replicat operate. Alternatively, you can follow along each step of the way.

Prerequisites

If you plan to execute the instructions in the tutorial, make sure all software is already installed.

The following table describes items that are referred to throughout the tutorial. You will need to identify your installation-specific values and substitute them as you go.

Item	Tutorial Reference	Description
Unix Programs	/ggs	Directory of Unix GoldenGate installation.
Unix Parameter Files	/ggs/dirprm	Directory for GoldenGate parameter files.
Unix Report Files	/ggs/dirrpt	Directory for output from GoldenGate programs.
Unix Definitions Files	/ggs/dirdef	Directory for generated Oracle DDL and definition files.
GGS temporary storage	/ggs/dirdat	Directory to hold temporary Extract trails
Oracle Logon	userid, password	User ID and password for the target database.
Unix System Network Address	unixserver2	IP address of the target Unix system in network.

The source Oracle database tables used in these examples can be created and loaded with sample data using the following commands.

```
% sqlplus userid/password
SQLPLUS> @/ggs/demo_ora_create
SQLPLUS> @/ggs/demo_ora_insert
SQLPLUS> exit
```

The target Oracle database tables used in these examples can be created using the following commands.

```
% sqlplus userid/password
SQLPLUS> @/ggs/demo_ora_create
SQLPLUS> exit
```

Overview of Tutorial Tasks

This section outlines the steps required in each phase of database load and replication.

Once the target database is created, it can be loaded with data from the Oracle database. To load the target database

- Start the Manager program on both systems, with a TCP/IP listening port configured.
- Run the Extract program on Unix to retrieve, convert and output data from the Oracle tables into a flat file on Unix.
- Run the Replicat program on Unix to insert the initial data into the target database.

After initial synchronization, Extract and Replicat work together to keep the databases in sync near real-time via incremental transaction replication. Perform this function by

- Adding supplemental transaction log data for update operations.
- Running the Extract program to retrieve and store the incremental changed data from the Oracle tables into trail files on the target Unix system.
- Running the Replicat program to replicate extracted data.

Once Extract and Replicat are running, changes are replicated perpetually. At this point, we will also demonstrate the following functions.

- How to retrieve information on Extract and Replicat status.
- How to gracefully stop replication.
- How to restart replication with transaction integrity.

Notes on Command Syntax

Commands throughout the tutorial make specific references to directories, file names, checkpoint group names, begin times, etc. Unless otherwise noted, these items do not have to correspond exactly in your environment; they are used to illustrated concrete examples. Where the prompt is written GGSCI {unixserver1}> the command should be executed on the source system. {unixserver2} indicates the target system.

For exact syntax, consult the Oracle GoldenGate Reference Guide.

Loading and Refreshing Oracle Tables

This section demonstrates the basic features of GoldenGate's Oracle to Oracle replication facilities. The following functions are illustrated.

- Loading the target Oracle database with initial data from the source Oracle database.
- Replicating Oracle database changes to another Oracle database using near real-time incremental replication.

Prepare the Environment to Map and Collect Data

Prior to executing data extraction, which moves data from Oracle to Oracle in the Unix environment, perform the following task.

- Add the MGRPORT parameter to the Manager Parameter file. This ensures that server collector processes can be dynamically created on the remote system to receive and log data created by Extract.

Preparing Manager to Start Dynamic Server Collectors

The Server Collector program receives incoming data over a TCP/IP connection from Extract on the source Unix system, and then outputs the incremental changes to temporary storage on the target Unix system. The Server Collector is automatically started by the Manager process at the request of the Extract program whenever moving data between systems.

The Manager program provides a number of important functions, including monitoring critical system components and starting GoldenGate processes. Before running any other GoldenGate programs on Unix, you must start Manager.

Before starting Manager, you must edit Manager's startup parameter file (called /ggs/dirprm/mgr.prm) and add the PORT parameter. You can do this manually with with the a Unix editor, or you can use GGSCI to start the **vi** program for you with these commands:

```
GGSCI (unixserver1) > EDIT PARAMS MGR
```

In either case, add the following text to the MGR.PRM file, then save the file and quit.

```
PORT 7809
```

Manager is started via GGSCI using the following commands.

```
GGSCI (unixserver1) > START MANAGER
```

If your target Oracle database is on another system, repeat the above steps on the target system. Start the target Manager process with the START MANAGER command.

```
GGSCI (unixserver2) > EDIT PARAMS MGR
```

```
PORT 7809
```

```
GGSCI (unixserver2) > START MANAGER
```

Initial Data Extract, Conversion and Load

At this point, data can be extracted directly from the Oracle tables, converted, then loaded into Oracle tables. Initial load from Oracle to Oracle involves the following tasks.

- Extract is configured as a batch task to retrieve data directly from the source tables and send the data directly to a Replicat batch task.
- Replicat is configured as a batch task to populate target tables.

Specifying Extract Parameters

Extract parameters are entered into an Extract parameter file you create via any text editor. The following file (named /ggs/dirprm/initext.prm) is an example of a file created for this purpose. Note that the ordering of the parameters is critical.

```
GGSCI (unixserver1) > EDIT PARAMS INITEXT
```

```
--
-- Extract parameter file to capture TCUSTORD
-- and TCUSTMER initial data for Replicat
--
EXTRACT INITEXT

USERID userid, PASSWORD password

RMTHOST unixserver2, MGRPORT 7809

RMTTASK REPLICAT, GROUP INITREP
TABLE schema.TCUSTMER;
TABLE schema.TCUSTORD;
```

Parameters explained:

Two dashes (--) at the beginning of a line indicates a comment.

EXTRACT is the group name used for this batch task, which retrieves data directly from the tables rather than the redo log and sends the data to the RMTTASK Replicat group.

USERID and PASSWORD must match an existing account in the Oracle database. The active Oracle database is indicated by the user's ORACLE_SID environment variable.

The RMTHOST parameter specifies the TCP/IP address of the target Unix system to which the data is moved. MGRPORT specifies the well known port number on which Manager has been configured to listen for requests for Extract Server Collectors. RMTHOST can also be specified as a standard TCP/IP host name.

The RMTTASK entry determines where the extracted data is output on the target, in this case, a Replicat group named INITREP.

Each TABLE entry specifies a table from which to extract data, and a TARGET structure for the data. If the source and target column names are the same, no other parameters are required. If your columns have different names, refer to the COLMAP statement on how to explicitly map each column. Make sure you change the schema to match the owner of the table.

Note also that a semi-colon (;) is required at the end of the TABLE entry.

Extracting a Subset of Columns and Rows from a Table

Frequently, a subset of the source table's columns are required on the target platform, rather than each and every column.

For example, the target application may use only 10 out of the 70 columns in the source table. In this case, the COLMAP clause can limit the mapping to those 10 columns.

As another alternative, the set of rows to transfer can be limited by specifying WHERE clauses. For example, the parameter entry TABLE TCUSTORD, WHERE (CUST_CODE = "ABC" OR CUST_CODE = "XYZ") would exclude any rows not meeting the criteria.

Configuring the Replicat Parameter File

As with Extract, you must create a Replicat parameter file. Launch GGSCI on the target system and issue the following GGSCI command:

```
GGSCI (unixserver2) > EDIT PARAMS INITREP
```

```
--
-- REPLICAT parameter file to replicate initial changes
-- for TCUSTMER.
--
REPLICAT INITREP

ASSUMETARGETDEFS
DISCARDFILE /ggs/dirrpt/tcustmer.dsc, PURGE

USERID userid, PASSWORD password

MAP schema.*, TARGET schema.*;
```

Parameters explained:

REPLICAT is the batch task group named and must match the name specified in the source Extract parameter file.

DISCARDFILE determines where operations that fail during replication are output. The discard file is useful for debugging problems during the replication process. It will contain any rejected rows and the associated causes. When PURGE is specified existing contents are purged at startup, or APPEND could be specified instead.

ASSUMETARGETDEFS tells Replicat to assume that the source Oracle tables are structured like the target tables. This eliminates the need to retrieve table definitions from the source system. Source definitions must be generated using DEFGEN if a source table is structured differently than the target.

USERID and PASSWORD must match an existing account in the Oracle database. The active Oracle database is indicated by the user's ORACLE_SID environment variable.

Each MAP entry establishes a relationship between a source Oracle table and a target Oracle table. In this example, we used a wildcard map.

In addition, any replication error will cause Replicat to abort (for example, a duplicate record condition). See the documentation on the following Replicat parameters to customize error response: HANDLECOLLISIONS, OVERRIDEDUPS, INSERTMISSINGUPDATES and REPERROR. Note that restart issues are discussed later in this tutorial.

Adding Extract Batch Task Group

Extract must be defined as a batch task within GGSCI before it may be started. Use GGSCI to configure the Extract Task with the following command.

```
GGSCI (unixserver1) > ADD EXTRACT INITEXT, SOURCEISTABLE
```

Adding Replicat Batch Task Group

Replicat must be defined as a batch task within GGSCI before it may be started. Use GGSCI to configure the Replicat Task with the following command.

```
GGSCI (unixserver2) > ADD REPLICAT INITREP, SPECIALRUN
```

Running Extract

The Extract program is initiated either in online mode with GGSCI or in batch mode from the Unix shell with the following commands. When Extract is run for the initial load, run it in batch mode as shown below.

```
GGSCI (unixserver1) > START EXTRACT INITEXT
```

The process may be monitored with the INFO or STATUS commands.

```
GGSCI (unixserver1) > INFO EXTRACT INITEXT
```

Note that the output will be sent to the Extract's report file. To view the file, use the following GGSCI command.

```
GGSCI (unixserver1) > VIEW REPORT INITEXT
```

When Extract completes, you will see statistics indicating how many records were output into the target tables.

Keeping the Source and Target Databases in Sync

After the initial load is complete, databases can be kept in sync using Extract and Replicat. All changes occurring against source tables are automatically detected by Extract, then formatted and transferred near real-time to temporary files on Unix. Once there, the data is read from these files and replicated to the target database by the Replicat program.

Perform the following tasks to implement extraction and replication.

On the source system

- Add supplemental log data for update operations.
- Set up an initial Extract checkpoint on source Unix system.
- Create an Extract parameter file on source Unix system.
- Start Extract.

On the target system

- Create a checkpoint table in the target database.
- Set up an initial Replicat checkpoint on target Unix system.
- Create a Replicat parameter file on target Unix system.
- Start Replicat.

Adding supplemental log data

By default, Oracle only logs changed columns for update operations. Normally, this means that primary key columns are not logged during an update operation. However, Replicat requires the primary key columns in order to apply the update on the target system. The ADD TRANDATA command in GGSCI is used to cause Oracle to log primary key columns for all updates.

To add supplemental log data, issue the following commands on the source Unix system.

```
$ cd /ggs
$ ggsci

GGSCI (unixserver1) > DBLOGIN USERID userid
GGSCI (unixserver1) > ADD TRANDATA schema.TCUSTMER
GGSCI (unixserver1) > ADD TRANDATA schema.TCUSTORD
```

The DBLOGIN command establishes a database connection for the specified user. The user is prompted for a password.

The ADD TRANDATA command causes Oracle to log primary key columns for all update operations on the specified table.

Setting up Extract Checkpoints

Checkpoints enable both Extract and Replicat to process data continuously from one run to another. Checkpoints enable Extract and Replicat to be restarted while ensuring that all records are replicated once and only once.

Extract requires two checkpoints: one into the Oracle redo log, which is the source of all database changes, and one into the remote extract trails. Remote extract trails are a series of temporary files created on the target Unix system that contain extracted changes.

To set up these checkpoints on your source system, issue the following commands on Unix.

```
$ cd /ggs
$ ggsci

GGSCI (unixserver1) > ADD EXTRACT EXTORA, TRANLOG, BEGIN NOW
GGSCI (unixserver1) > ADD RMTTRAIL /ggs/dirdat/rt, EXTRACT EXTORA,
MEGABYTES 10
```

The ADD EXTRACT command establishes an Extract checkpoint group name and also identifies the Oracle redo log as the data source. The BEGIN NOW clause causes Extract to process database operations occurring at or after the time the Extract group was added. Alternatively, you can specify a date and time instead of the keyword NOW.

The ADD RMTTRAIL command establishes a checkpoint into a remote extract trail. After each file in this trail reaches approximately 10 megabytes, Extract creates the next file in the sequence. Files will be named RT000000, RT000001, RT000002 and so on. These files are the source of input to the Replicat program. Note that instead of RT you could specify any two-character prefix.

Choose group names, destination files and sizes appropriate for your environment.

Notes on Remote Extract Trails

Remote extract trail files are only temporary. Manager can be configured (using PURGEOLDEXTRACTS) to delete the trail file when Extract and Replicat are both finished processing it. Therefore, intermediate storage requirements are actually quite small.

Intermediate storage allows extraction and replication activities to occur independently of each other. Frequently, extraction will run perpetually, while replication will be applied once per day in batch due to considerations in the target application.

Rollover from one file to the next can also be controlled by time of day rather than file size, via the EXTRACT ROLLOVER parameter.

Configuring the Extract Parameter File

Most Extract parameters are entered into a parameter file. You may create your parameter file manually using **vi** or another editor on Unix. The parameter file name is the same name as the name of the group, in this case EXTORA. Issue the following command to launch **vi** from GGSCI:

```
GGSCI (unixserver1) > EDIT PARAMS EXTORA
```

You can also edit the file /ggs/dirprm/extora.prm directly from any other text editor.

Enter the following parameters into EXTORA. Note that the ordering of the parameters is important.

```
--
-- Extract parameter file to capture TCUSTORD
-- and TCUSTMER changes
--
EXTRACT EXTORA

USERID userid, PASSWORD password

RMTHOST unixserver2, MGRPORT 7809

RMTTRAIL /ggs/dirdat/rt

TABLE schema.TCUSTMER;
TABLE schema.TCUSTORD;
```

Parameters explained:

EXTRACT EXTORA identifies the particular checkpoint group with which this parameter file is associated.

USERID and PASSWORD must match an existing account in the Oracle database. The active Oracle database is indicated by the user's ORACLE_SID environment variable.

RMTHOST identifies the system to which to output extracted database changes and must be specified before RMTTRAIL. MGRPORT specifies the port number on which Manager has been configured to listen for requests for Server Collectors. RMTHOST can also be specified as a standard TCP/IP host name.

RMTTRAIL specifies the file set to which database changes will be output. Changes detected on any TABLE specified below this entry will be output to the remote trail.

Each TABLE entry specifies a table from which to extract data.

Creating a GLOBALS Parameter File and Adding CHECKPOINTTABLE

Parameters that affect all GoldenGate processes are defined in the GLOBALS parameter file. This file must be named GLOBALS (uppercase, without an extension) and located in your installation directory. You may create your parameter file manually from the command shell using **vi** or another editor on Unix.

Edit this file using GGSCI on the target system:

```
$ cd /ggs
$ ggsci
GGSCI (unixserver2) > EDIT PARAMS ./GLOBALS
```

And add the following parameter to establish the name of the check point table.

```
CHECKPOINTTABLE schema.ggchkpt
```

Adding the Database Checkpoint Table

Exit the GGSCI session to activate the new GLOBALS parameter.

```
GGSCI (unixserver2) > exit
```

Now start a new GGSCI session and create the checkpoint table on the target.

```
$ ggsci

GGSCI (unixserver2) DBLOGIN USERID userid, PASSWORD password
GGSCI {unixserver2) ADD CHECKPOINTTABLE
```

The database checkpoint table will be created on the target system. In this example, the name is ggchkpt.

Setting up a Replicat Checkpoint

The Replicat checkpoint establishes an initial position into the extract trail created by Extract. By default, this will always be the first record extracted. The checkpoint is updated after each transaction, ensuring that all data is processed from run to run.

To set up the Replicat checkpoint, issue the following commands on the target Unix system.

```
GGSCI (unixserver2)> ADD REPLICAT REPORA, EXTTRAIL /ggs/dirdat/rt
```

The ADD REPLICAT command establishes the extract trail (EXTTRAIL) created by Extract as the source of information to replicate. REPORA is the name given to this checkpoint group.

Configuring the Replicat Parameter File

As with Extract, you must create a Replicat parameter file. Here you can manually use **vi** or another editor on Unix. In this example, we create the following file (called /ggs/dirprm/repora.prm) for this purpose. Make sure the file is saved in a text format.

Alternatively, you can launch the **vi** program within GGSCI on the target Unix system.

```
$ cd /ggs
$ ggsci

GGSCI (unixserver2) > EDIT PARAMS REPORA
```

```
--
-- REPLICAT parameter file to replicate changes
-- for TCUSTORD and TCUSTMER.
--
REPLICAT REPORA
Deleted PURGEOLDEXTRACTS parameter
ASSUMETARGETDEFS

DISCARDFILE /ggs/dirrpt/repora.dsc, PURGE

USERID userid, PASSWORD password

MAP schema.TCUSTORD, TARGET schema.TCUSTORD;
MAP schema.TCUSTMER, TARGET schema.TCUSTMER;
```

Parameters explained:

REPLICAT REPORA associates this parameter file with the checkpoint established via GGSCI. This also implicitly establishes the extract trail /ggs/dirdat/rt as the source of data to replicate.

ASSUMETARGETDEFS allows Replicat to assume that the source Oracle tables are structured like the target tables. This eliminates the need to retrieve table definitions from the source system. Source definitions must be generated using DEFGEN if a source table is structured differently than the target.

DISCARDFILE determines where records from operations that fail during replication are output. The discard file is useful for debugging problems during the replication process. This file will contain any rejected rows and the associated causes. Any existing contents are purged at startup when PURGE is specified (APPEND could be specified instead).

USERID and PASSWORD must match an existing account in the Oracle database. The active Oracle database is indicated by the user's ORACLE_SID environment variable.

Each MAP entry establishes a relationship between a source Oracle table and a target Oracle table.

Note that no END parameter is specified. The omission of END means that Replicat will continue to run until explicitly stopped by the user (or a fatal error occurs). END is specified when Replicat is run in batch mode versus online mode.

In addition, any replication error will cause Replicat to abort (for example, a duplicate record condition). See the documentation on the following Replicat parameters to customize error response: HANDLECOLLISIONS, OVERRIDEDUPS, INSERTMISSINGUPDATES and REPERROR. Note that restart issues are discussed later in this tutorial.

Adding Demo Transactions

At this point, you can add some demo transactions to the source Oracle database with the following Unix commands on the source system.

```
$ sqlplus userid/password
```

```
SQLPLUS> @/ggs/demo_ora_misc
```

These transactions will be extracted and replicated in the following steps.

Running Extract

The Extract program for capturing database changes is initiated from GGSCI on the source Unix system.

```
$ /ggs/ggsci

GGSCI (unixserver1) > START EXTRACT EXTORA
```

Extract continues to run until explicitly stopped via GGSCI.

Running Replicat

Replicat is initiated from the target Unix command prompt as follows.

```
$ cd /ggs
$ ggsci

$GGSCI (unixserver2) > START REPLICAT REPORA
```

Replicat will continue to run until explicitly stopped (usually via GGSCI) or a fatal error occurs.

Initializing the Target While the Source Database Remains On-line

If the source database is being updated during initial load activities, the order of synchronization activities must be changed in the following manner.

- Change capture must be started (using Extract)
- Initial capture and load must be executed (Extract, Replicat).
- Change replication must be initiated (Replicat). When this step is performed, the Replicat HANDLECOLLISIONS parameter must also be set to account for possible duplicate and missing record conditions, which are normal in this situation.

Assessing Replication Status, Diagnosing Problems

Using the REPORT and DISCARD Files

Both Extract and Replicat echo parameters, as well as diagnostic messages, to their respective report files.

In order to view the report file while Extract is running, issue the following command from GGSCI on Unix (EXTORA is the name of the EXTRACT group).

```
GGSCI (unixserver1) > VIEW REPORT EXTORA
```

The Replicat report can be viewed on the target system in a similar manner.

```
GGSCI (unixserver2) > VIEW REPORT REPORA
```

Any errors in the parameter file are output to the report and out files. Once the message “Run-Time Warnings” appears in the report, all parameters have been validated and data processing has begun.

Discard files also serve as a source for debugging replication problems. Frequently, looking in the discard file at specific record values and error numbers is the fastest path to problem resolution.

Obtaining Extract and Replicat Process Status through GGSCI

To obtain the status and history of an Extract process, issue the following commands.

From GGSCI on Unix

```
GGSCI (unixserver1) > INFO EXTRACT EXTORA, DETAIL
```

The output of the INFO command will look something like this:

```
GGSCI (unixserver1) > info extract extora, detail

EXTRACT    EXTORA            Last Started 2009-01-03 17:20   Status
RUNNING
Checkpoint Lag             00:00:00 (updated 00:00:05 ago)
Log Read Checkpoint        Oracle Redo Logs
                           2009-01-03 09:46:36  Seqno 212, RBA 195584

Target Extract Trails:

  Remote Trail Name                           Seqno        RBA     Max MB

  /ggs/dirdat/rt                                  0      14960         10

  Extract Source                    Begin             End

  /oracle/data/redo02.log           2009-01-03 17:20  2002-01-03 17:20
  Not Available                     * Initialized *   2002-01-03 17:15

Current directory              /ggs

Report file                    /ggs/dirrpt/extora.rpt
Parameter file                 /ggs/dirprm/extora.prm
Checkpoint file                /ggs/dirchk/extora.cke
Process file                   /ggs/dirpcs/extora.pce
Error log                      /ggs/ggserr.log
```

The statistics show the history of Extract runs for the EXTORA checkpoint group and the current file in the remote extract trail. Also indicated is the current status of the process, in this case RUNNING.

Similarly, to obtain the status and history of a Replicat process, issue the following commands.

From GGSCI on Unix

```
GGSCI (unixserver2) > INFO REPLICAT REPORA, DETAIL

REPLICAT   REPORA          Last Started 2009-01-03 09:47   Status RUNNING
Checkpoint Lag             00:00:00 (updated 00:00:02 ago)
Log Read Checkpoint        File /ggs/dirdat/rt000000
                           2009-01-03 17:39:28  RBA 14960
```

```
  Extract Source                       Begin             End

  /ggs/dirdat/rt000000         2009-01-03 17:39  2009-01-03 17:40
  /ggs/dirdat/rt000000         * Initialized *   2009-01-03 17:39
  /ggs/dirdat/rt000000         * Initialized *   First Record


Current directory               /ggs

Report file                     /ggs/dirrpt/repora.rpt
Parameter file                  /ggs/dirprm/repora.prm
Checkpoint file                 /ggs/dirchk/repora.chk
Checkpoint table                TARGET.GGCHKPT
Process file                    /ggs/dirpcs/reporaT.pcr
Stdout file                     /ggs/dirout/repora.out
Error log                       /ggs/ggserr.log
```

Stopping and Restarting Extract and Replicat

The GGSCI STOP Command

Both Extract and Replicat can be stopped via the GGSCI utility. To stop Extract, run GGSCI from the source system on which Extract is running and issue the following command.

```
GGSCI (unixserver1) > STOP EXTRACT EXTORA
```

This command may take a few seconds to execute.

On the target system, Replicat is stopped with the following command.

```
GGSCI (unixserver2) > STOP REPLICAT REPORA
```

Viewing Replication Statistics

Extract outputs how many inserts, updates and deletes it captured after normal stoppage into the REPORT file. View the REPORT file with the VIEW REPORT command described earlier.

Replicat outputs how many inserts, updates and deletes were applied to the target database in the standard out file and the report file after the process is stopped normally.

Generating interim reports is also possible without stopping Extract and Replicat. See the Extract and Replicat REPORT parameter for more details, and the GGSCI SEND…REPORT commands.

Restarting Extract and Replicat

Both Extract and Replicat are restarted in the same manner as they were originally started, whether or not the previous run ended gracefully or abnormally.

The extraction and replication processes automatically start where they left off, while ensuring no transactions are missed or duplicated. This integrity is guaranteed by the checkpoints maintained by each process.

Where to Go for More Information

Hopefully, this tutorial has provided a quick overview of what you need to do in order to replicate data from Oracle to Oracle. Undoubtedly, you will eventually fine-tune this process in your own environment.

Reference the Oracle GoldenGate Reference Guide and the Oracle GoldenGate Administration Guide for additional information on:

- Extract Parameters for Windows and Unix
- Replicat Parameters for Windows and Unix
- Extract Management Considerations
- Replicat Management Considerations