Configuring Oracle GoldenGate with NetApp Storage

Padmanabhan Sadagopan, NetApp
Mike Papio, Oracle
September 2011 | TR-3960

EXECUTIVE SUMMARY

This technical report describes the installation and configuration of Oracle® GoldenGate with NetApp® storage systems. Oracle GoldenGate—with real-time data integration and replication across diverse IT systems—addresses the enterprise business needs of zero-downtime migration and continuous availability of transactional data. NetApp storage systems provide the shared storage infrastructure needed to implement Oracle GoldenGate, specifically for the Oracle GoldenGate server configuration files and database-related files.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Transactional data and business applications that depend on such data are mission critical and are often expected to provide 24/7 uptime. With the ever-expanding use of online applications, the importance of transactional data is steadily increasing. Oracle GoldenGate software facilitates the management and movement of transactional data across the enterprise by providing data availability, reduced latency, flexibility, and data integrity. Oracle GoldenGate—together with NetApp storage technologies—helps enterprises to easily migrate data in real time among heterogeneous systems. This report summarizes the Oracle GoldenGate technology architecture and processes and describes how to implement this software platform on NetApp storage.

2 AUDIENCE

This document is intended for system and storage administrators, product management, and IT infrastructure managers who are familiar with concepts pertaining to Oracle GoldenGate and NetApp Data ONTAP® technologies.

3 ORACLE GOLDENGATE

Oracle GoldenGate provides a real-time, log-based, change data capture and replication software platform that enables high volumes of transactional data to be moved between heterogeneous platforms and databases—even bidirectionally—at subsecond speed, with minimal impact on source and target systems. With this unique technology, Oracle provides continuous availability and concurrent data integration solutions to deliver real-time access to real-time information. Oracle GoldenGate is designed for easy, low-impact implementation, operation, modification, and extension to support the evolving needs of enterprise information.

4 NETAPP STORAGE SOLUTIONS

The NetApp Data ONTAP operating system is scalable and flexible. This system provides seamless access to data using block- and file-level (SAN/NAS) protocols on the storage system. The following standard features of Data ONTAP can be leveraged when you deploy your enterprise applications on NetApp storage:

- **NetApp FlexVol®**. Enables you to create volumes that you can manage and move independently from physical storage. FlexVol volumes can also be increased or decreased in size on-the-fly.
- **NetApp Snapshot™**. Enables system administrators to rapidly create point-in-time copies of file systems, which can be used to protect data.
- **NetApp SnapRestore®**. Recovers entire file systems or data volumes in seconds, regardless of capacity or number of files.
- **NetApp FlexClone®**. Provides instant replication of data files, LUNs, and volumes—without requiring additional storage space at the time of creation.

For complete information on using these features, see the [Data ONTAP Storage Management Guide](#) on the [NetApp Support Site](#) (formerly known as NOW™).

5 ORACLE GOLDENGATE ON NETAPP ARCHITECTURE OVERVIEW

By configuring GoldenGate on NetApp storage, you can take advantage of all the previously mentioned standard features of Data ONTAP.

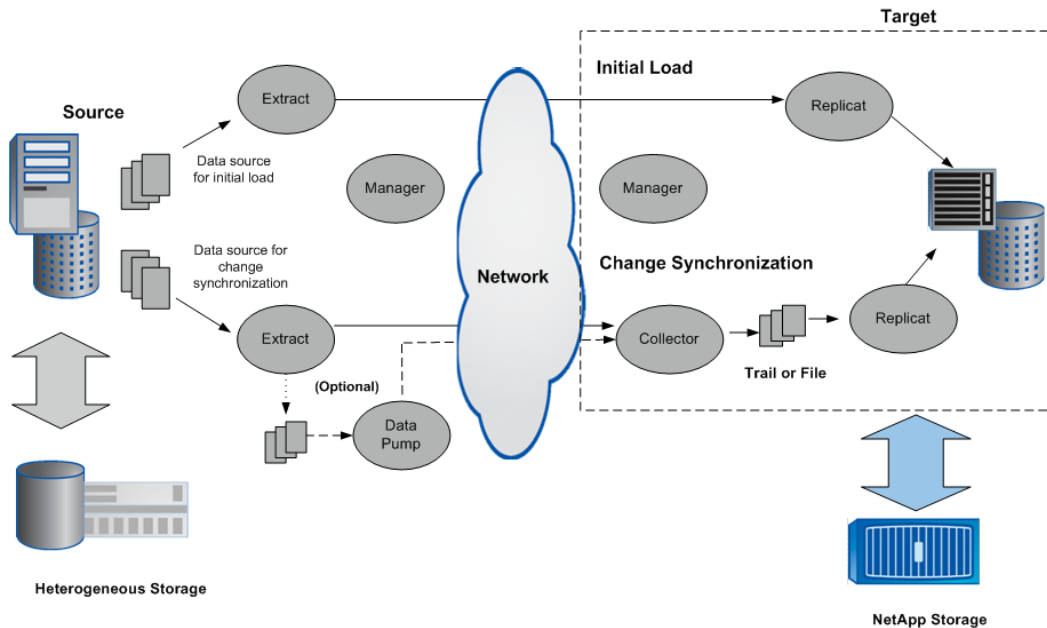**Figure 1) Oracle GoldenGate on NetApp logical architecture.**

Figure 1 illustrates the logical architecture of Oracle GoldenGate for initial data loads and for the replication of ongoing database changes. This is the basic configuration; variations of this model are recommended depending on business needs. The target GoldenGate server software binary/configuration files and target database files are stored on NetApp storage.

Oracle GoldenGate is composed of the following components:

- Extract
- Trail Files
- Data Pump
- Collector
- Replicat
- Manager
- Checkpoints

5.1 OVERVIEW OF ORACLE GOLDENGATE EXTRACT

The Extract process runs on the source system and is the extraction (capture) mechanism of Oracle GoldenGate. Extract captures all changes made to objects that you configure for synchronization. Extract stores the changes until it receives commit records or rollbacks for the transactions that contain them. When a rollback is received, Extract discards the data for that transaction. When a commit is received, Extract sends the data for that transaction to the trail for propagation to the target system. All of the log records for a transaction are written to the trail as a sequentially organized transaction unit. This design promotes both speed and data integrity.

Extract may be configured to read directly from database tables. This is typically done when Oracle GoldenGate is used to instantiate a target. Extract may also be configured to read from Oracle Archive logs only. This mode of operation is typically used when there is no need for real-time data capture and delivery.

5.2   OVERVIEW OF ORACLE GOLDENGATE TRAIL FILES

To support the continuous extraction and replication of database changes, Oracle GoldenGate stores the captured changes temporarily on disk in a series of files called a trail. A trail can exist on the source or target system, or on an intermediary system, depending on how you configure Oracle GoldenGate. On the local system it is known as an extract trail (or local trail). On a remote system it is known as a remote trail.

By using a trail for storage, Oracle GoldenGate can support data accuracy and fault tolerance. The use of a trail also allows extraction and replication activities to occur independently of each other. With these processes separated, you have more choices for how data is delivered. For example, instead of extracting and replicating changes continuously, you could extract changes continuously but store them in the trail for replication to the target later, whenever the target application needs them.

5.3   OVERVIEW OF ORACLE GOLDENGATE DATA PUMPS

A data pump is a secondary Extract group within the source Oracle GoldenGate configuration. If a data pump is not used, Extract must send data to a remote trail on the target. In a typical configuration that includes a data pump, however, the primary Extract group writes to a trail on the source system. The data pump reads this trail and sends the data over the network to a remote trail on the target. The data pump adds storage flexibility and also serves to isolate the primary Extract process from TCP/IP activity.

Like a primary Extract group, a data pump can be configured for either online or batch processing. It can perform data filtering, mapping, and conversion, or it can be configured in pass-through mode, in which data is passively transferred as is, without manipulation. Pass-through mode increases the throughput of the data pump, because all functionality that looks up object definitions is bypassed.

Data pump configuration enhances the fault tolerance and reliability of the overall Oracle GoldenGate environment. In the event of a network failure (between the source and the target systems), Oracle GoldenGate can continue to capture transactions because the data can be queued up locally in the Trail Files on the source, enhancing its recoverability in case of database failures.

5.4   OVERVIEW OF ORACLE GOLDENGATE COLLECTOR

Collector is a process that runs in the background on the target system. Collector receives extracted database changes that are sent across the TCP/IP network and writes them to a trail or extract file. Typically, Manager starts Collector automatically when a network connection is required. When Manager starts Collector, the process is known as a dynamic Collector, and Oracle GoldenGate users generally do not interact with it. However, you can run Collector manually. This is known as a static Collector. Not all Oracle GoldenGate configurations use a Collector process.

5.5   OVERVIEW OF ORACLE GOLDENGATE REPLICAT

The Replicat process runs on the target system. Replicat reads extracted data changes and Data Definition Language (DDL) changes (if supported) that are specified in the Replicat configuration; it then replicates them to the target database. Replicat applies extracted data changes to target objects using a native database interface or ODBC, depending on the database type. Replicated DDL and sequences are also applied, if supported for the type of database that is being used. To preserve data integrity, Replicat applies the replicated changes in the same order as they were committed to the source database.

You can use multiple Replicat processes with multiple Extract processes in parallel to increase throughput. Each set of processes handles different objects. To differentiate among processes, you assign each one a group name. You can delay Replicat so that it waits a specific amount of time before applying data to the target database. A delay might be desirable, for example, to prevent the propagation of errant SQL, to control data arrival across different time zones, or to allow time for other planned events to occur.

5.6   OVERVIEW OF ORACLE GOLDENGATE MANAGER

Manager is the control process of Oracle GoldenGate. Manager must be running on each system in the Oracle GoldenGate configuration before Extract or Replicat can be started. Manager must remain running while those processes are running so that resource management functions are performed. Manager performs the following functions:

- Monitors and restarts Oracle GoldenGate processes
- Issues threshold reports, for example, when throughput slows down or when synchronization latency increases
- Maintains trail files and logs
- Allocates data storage space
- Reports errors and events
- Receives and routes requests from the user interface

5.7   OVERVIEW OF ORACLE GOLDENGATE CHECKPOINTS

Checkpoints store the current read and write positions of a process to disk for recovery purposes. These checkpoints provide that data changes marked for synchronization actually are extracted by Extract and replicated by Replicat, and they prevent redundant processing. They provide fault tolerance by preventing the loss of data should the system, the network, or an Oracle GoldenGate process need to be restarted. For complex synchronization configurations, checkpoints enable multiple Extract or Replicat processes to read from the same set of trails.

Checkpoints work with interprocess acknowledgments to prevent messages from being lost in the network. Oracle GoldenGate has a proprietary guaranteed-message delivery technology.

6 ORACLE GOLDENGATE SERVER AND STORAGE ENVIRONMENT PREPARATION

To prepare the environment to install Oracle GoldenGate on NetApp storage, you must do the following:

1. Configure the NetApp storage system.
2. Configure the Oracle GoldenGate server.

Note: Typically, the source and target server platform and storage system will be different, but in this example we used the following components for both:

- Oracle Database 11g™ Enterprise Edition Release 11.2.0.1.0–64-bit
- Oracle GoldenGate Version 11.1.1.1 OGCORE_11.1.1_PLATFORMS_110421.2040
- Red Hat Linux® 4.1.2-44
- NetApp FAS2050 storage controllers
- NetApp Data ONTAP 7.3.5.1

6.1   NETAPP STORAGE CONFIGURATION

You create FlexVol volumes to provide resizable, flexible file systems that can be mounted and accessed from Oracle GoldenGate servers. FlexVol volumes allow you to manage the logical layer of the file system independently of the physical layer of storage.

FlexVol volumes can grow automatically to make sure that space in your aggregates is used efficiently and to reduce the likelihood that your volumes will run out of space.

To create and export the storage volumes for Oracle GoldenGate, you must complete the following steps:

1. Update the `/etc/hosts` file in the storage system to include GoldenGate source and target servers (name or ip address) so that they can communicate with each other.

    For example:

    ```
    Storage system>>  wrfile  /etc/hosts

    10.61.161.90  riker9.rtp.netapp.com  riker9

    10.61.161.91  riker10.rtp.netapp.com  riker10
    ```

2. Enable `rsh`/`ssh` access to the GoldenGate source and target servers.
    For example:

    ```
    > options  rsh enable  on

    > options  ssh  enable on
    ```

3. Create the aggregate and volumes in the storage system.
    For example:

    ```
    > aggr create  aggr1

    > vol create filer1_gg1_vol1  -s volume aggr1  30g

    Creation of volume 'filer1_gg1_vol1l' with size 30g on containing
    aggregate 'aggr1' has completed
    ```

4. NFS export the volumes. Edit the file `/etc/exports` to NFS export the volumes to the GoldenGate source and target servers.
    For example:

    ```
    Storage system>>  wrfile  /etc/exports

        /vol/filer1_gg1_vol1    -sec=sys,rw,anon=0
    ```

6.2   ORACLE GOLDENGATE SERVER CONFIGURATION

Create NetApp volumes and NFS mount them as a physical directory to install and configure the GoldenGate binaries and configuration files. You can do the same for the target Oracle Database-related files.

1. Create physical directories (in this example `/u35/filer1_gg1_vol1`) on the GoldenGate source and target servers for the Oracle GoldenGate software and configuration files.

2. Edit the `/etc/fstab` file to include entries similar to the following:
    ```
    10.61.175.180:/vol/filer1_gg1_vol1  /u35/filer1_gg1_vol1  nfs
    rw,bg,hard,rsize=32768,wsize=32768,vers=3,nointr,timeo=600,tcp,actimeo=0
    ```

3. Mount the volumes from the GoldenGate source and target servers, using the **mount** command.

Note: When using an NFS-mounted file system with Oracle GoldenGate, the setting for file system caching or buffered I/O must be disabled. This is because when one process registers the end of a trail file or transaction log and moves on to the next in sequence, data in the NFS I/O buffer gets flushed to disk. The net result is skipped data that cannot be corrected with GoldenGate parameter EOFDELAY. See section 9 for links to more information about NFS mount option best practices.

4. Create physical directories for NFS mounted volumes for target Oracle Database files (for example, data files, control files, log files, and archive log files).

```
Filer1:/vol/riker10_oradata on /u31/oradata type nfs
(rw,bg,hard,rsize=32768,wsize=32768,nfsvers=3,nointr,timeo=600,tcp,addr=10.61.175.181)
Filer1:/vol/riker10_oracntl on /u32/oracntl type nfs
(rw,bg,hard,rsize=32768,wsize=32768,nfsvers=3,nointr,timeo=600,tcp,addr=10.61.175.181)
Filer1:/vol/riker10_oralog on /u33/oralog type nfs
(rw,bg,hard,rsize=32768,wsize=32768,nfsvers=3,nointr,timeo=600,tcp,addr=10.61.175.181)
Filer1:/vol/riker10_oraarch on /u34/oraarch type nfs
(rw,bg,hard,rsize=32768,wsize=32768,nfsvers=3,nointr,timeo=600,tcp,addr=10.61.175.181)
Filer1:/vol/filer2_gg2_vol2 on /u35/filer2_gg2_vol2 type nfs
(rw,bg,hard,rsize=32768,wsize=32768,nfsvers=3,nointr,timeo=600,tcp,actimeo=0,addr=10.61.175.181)
```

7 ORACLE GOLDENGATE BASIC CONFIGURATION

The following are the steps to set up a basic configuration of Oracle GoldenGate. This configuration has online change synchronization feeding from the transaction logs of the Oracle Database by using the primary Extract (Capture) to be consumed by an Extract Data Pump and applied by the Replicat (Delivery) process.

7.1   ORACLE GOLDENGATE INSTALLATION

These are the high-level steps to install Oracle GoldenGate:

1. Download Oracle GoldenGate software from Oracle Technology Network (OTN):
   www.oracle.com/technetwork/middleware/goldengate/downloads/index.html

2. Install Oracle GoldenGate on the NFS-mounted directory created in section 6.2. Follow the generic installation instructions for the source and target machine installations available in Chapter 2 of "Installing GoldenGate" at http://download.oracle.com/docs/cd/E18101_01/doc.1111/e17799.pdf.

3. Extract the Oracle GoldenGate mediapack.zip file to the system and directory where you want Oracle GoldenGate to be installed.

4. Run the command shell and change directories to the new Oracle GoldenGate directory.

5. From the Oracle GoldenGate directory, run the GoldenGate Software Command Interface (GGSCI) program:
   GGSCI

6. In GGSCI, issue the following command to create the Oracle GoldenGate working directories:
   CREATE SUBDIRS

7. Issue the following command to exit GGSCI:
   EXIT

To use Oracle GoldenGate, you must configure the Manager process. You must specify a TCP/IP port for Manager to use. You can also specify optional parameters that control dynamic port assignments, trail file maintenance, and other properties.

The Oracle GoldenGate Manager performs a number of functions like starting the other GoldenGate processes, trail log file management, and reporting. The Manager process must be configured on both the source as well as the target system, and configuration is carried out via a parameter file just as with the other GoldenGate processes like Extract and Replicat.

Launch the GGSCI to edit the Manager parameter file, as shown below:

GGSCI (riker9.rtp.netapp.com) 1> **ggsci**

GGSCI (riker9.rtp.netapp.com) 2> **edit param mgr**

GGSCI (riker9.rtp.netapp.com) 3> **start mgr**

Manager started.

GGSCI (riker9.rtp.netapp.com) 4> **info mgr**

Manager is running (IP port riker9.rtp.netapp.com.7575).

SAMPLE INSTALLATION SESSION OUTPUT

The session output in this section demonstrates the successful installation of GoldenGate and the real-time replication between the databases.

Source Installation:

```
[oracle@riker9 filer1_gg1_vol1]$ unzip V26185-01.zip
Archive:  V26185-01.zip
  inflating: fbo_ggs_Linux_x64_ora11g_64bit.tar
  inflating: OGG_WinUnix_Rel_Notes_11.1.1.1.0.pdf
  inflating: README.txt
[oracle@riker9 filer1_gg1_vol1]$ tar -xvf fbo_ggs_Linux_x64_ora11g_64bit.tar
[oracle@riker9 ~]$ cd /u35/filer1_gg1_vol1/
[oracle@riker9 filer1_gg1_vol1]$ ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 11.1.1.1 OGGCORE_11.1.1_PLATFORMS_110421.2040
Linux, x64, 64bit (optimized), Oracle 11g on Apr 21 2011 22:42:14
Copyright (C) 1995, 2011, Oracle and/or its affiliates. All rights reserved.

GGSCI (riker9.rtp.netapp.com) 1> create subdirs

Creating subdirectories under current directory /u35/filer1_gg1_vol1

Parameter files                /u35/filer1_gg1_vol1/dirprm: created
Report files                   /u35/filer1_gg1_vol1/dirrpt: created
Checkpoint files               /u35/filer1_gg1_vol1/dirchk: created
Process status files           /u35/filer1_gg1_vol1/dirpcs: created
SQL script files               /u35/filer1_gg1_vol1/dirsql: created
Database definitions files     /u35/filer1_gg1_vol1/dirdef: created
Extract data files             /u35/filer1_gg1_vol1/dirdat: created
Temporary files                /u35/filer1_gg1_vol1/dirtmp: created
Veridata files                 /u35/filer1_gg1_vol1/dirver: created
Veridata Lock files            /u35/filer1_gg1_vol1/dirver/lock: created
Veridata Out-Of-Sync files     /u35/filer1_gg1_vol1/dirver/oos: created
Veridata Out-Of-Sync XML files /u35/filer1_gg1_vol1/dirver/oosxml: created
Veridata Parameter files       /u35/filer1_gg1_vol1/dirver/params: created
Veridata Report files          /u35/filer1_gg1_vol1/dirver/report: created
Veridata Status files          /u35/filer1_gg1_vol1/dirver/status: created
Veridata Trace files           /u35/filer1_gg1_vol1/dirver/trace: created
Stdout files                   /u35/filer1_gg1_vol1/dirout: created

GGSCI (riker9.rtp.netapp.com) 2> edit param mgr
GGSCI (riker9.rtp.netapp.com) 4> start mgr
Manager started.

GGSCI (riker9.rtp.netapp.com) 5> info mgr
Manager is running (IP port riker9.rtp.netapp.com.7575).
```

Target Installation:

```
[oracle@ riker10 filer2_gg2_vol2]$ unzip V26185-01.zip
Archive:  V26185-01.zip
  inflating: fbo_ggs_Linux_x64_ora11g_64bit.tar
  inflating: OGG_WinUnix_Rel_Notes_11.1.1.1.0.pdf
  inflating: README.txt
[oracle@ riker10 filer2_gg2_vol2]$ tar -xvf fbo_ggs_Linux_x64_ora11g_64bit.tar
[oracle@ riker10 ~]$ cd /u35/ filer2_gg2_vol2/
oracle@riker10 filer2_gg2_vol2]$ ggsci
Oracle GoldenGate Command Interpreter for Oracle
Version 11.1.1.1 OGGCORE_11.1.1_PLATFORMS_110421.2040
Linux, x64, 64bit (optimized), Oracle 11g on Apr 21 2011 22:42:14
Copyright (C) 1995, 2011, Oracle and/or its affiliates. All rights reserved.
GGSCI (riker10.rtp.netapp.com) 1> create subdirs
Creating subdirectories under current directory /u35/filer2_gg2_vol2
Parameter files                 /u35/filer2_gg2_vol2/dirprm: created
Report files                    /u35/filer2_gg2_vol2/dirrpt: created
Checkpoint files                /u35/filer2_gg2_vol2/dirchk: created
Process status files            /u35/filer2_gg2_vol2/dirpcs: created
SQL script files                /u35/filer2_gg2_vol2/dirsql: created
Database definitions files      /u35/filer2_gg2_vol2/dirdef: created
Extract data files              /u35/filer2_gg2_vol2/dirdat: created
Temporary files                 /u35/filer2_gg2_vol2/dirtmp: created
Veridata files                  /u35/filer2_gg2_vol2/dirver: created
Veridata Lock files             /u35/filer2_gg2_vol2/dirver/lock: created
Veridata Out-Of-Sync files      /u35/filer2_gg2_vol2/dirver/oos: created
Veridata Out-Of-Sync XML files  /u35/filer2_gg2_vol2/dirver/oosxml: created
Veridata Parameter files        /u35/filer2_gg2_vol2/dirver/params: created
Veridata Report files           /u35/filer2_gg2_vol2/dirver/report: created
Veridata Status files           /u35/filer2_gg2_vol2/dirver/status: created
Veridata Trace files            /u35/filer2_gg2_vol2/dirver/trace: created
Stdout files                    /u35/filer2_gg2_vol2/dirout: created

GGSCI (riker10.rtp.netapp.com) 2> edit param mgr
GGSCI (riker10.rtp.netapp.com) 3> start mgr
Manager started.
GSCI (riker10.rtp.netapp.com) 4> info mgr
Manager is running (IP port riker10.rtp.netapp.com.7575).
```

DATABASE PARAMETER FILES

The following are the parameter files that were used in configuring all the Oracle GoldenGate processes.

Figure 2) Extract (etpccdb.prm) parameter file for ETPCCDB.

```
EXTRACT ETPCCDB
USERID ggtest, PASSWORD ggtest
DISCARDFILE ./dirrpt/etpccdb.dsc, append
EXTTRAIL ./dirdat/et
REPORTCOUNT EVERY 5 MINUTES, RATE

TABLE ggtest.WAREHOUSE ;
TABLE ggtest.DISTRICT ;
TABLE ggtest.CUSTOMER ;
TABLE ggtest.HISTORY ;
TABLE ggtest.NEW_ORDER ;
TABLE ggtest.O_ORDER ;
TABLE ggtest.ORDER_LINE ;
TABLE ggtest.ITEM ;
TABLE ggtest.STOCK ;
```

Figure 3) Extract data pump (ptpccdb.prm) parameter file for PTPCCDB.

```
EXTRACT PTPCCDB
PASSTHRU
DISCARDFILE ./dirrpt/ptpccdb.dsc, append
RMTHOST riker10, MGRPORT 7575
RMTTRAIL ./dirdat/pt
REPORTCOUNT EVERY 5 MINUTES, RATE

TABLE ggtest.WAREHOUSE ;
TABLE ggtest.DISTRICT ;
TABLE ggtest.CUSTOMER ;
TABLE ggtest.HISTORY ;
TABLE ggtest.NEW_ORDER ;
TABLE ggtest.O_ORDER ;
TABLE ggtest.ORDER_LINE ;
TABLE ggtest.ITEM ;
TABLE ggtest.STOCK ;
```

Figure 4) Replicat (rtpccdb.prm) parameter file for RTPCCDB.

```
REPLICAT RTPCCDB
USERID ggtest, PASSWORD ggtest
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rtpccdb.dsc, append
REPORTCOUNT EVERY 5 MINUTES, RATE

MAP ggtest.WAREHOUSE, TARGET ggtest.WAREHOUSE ;
MAP ggtest.DISTRICT, TARGET ggtest.DISTRICT ;
MAP ggtest.CUSTOMER, TARGET ggtest.CUSTOMER ;
MAP ggtest.HISTORY, TARGET ggtest.HISTORY ;
MAP ggtest.NEW_ORDER, TARGET ggtest.NEW_ORDER ;
MAP ggtest.O_ORDER, TARGET ggtest.O_ORDER ;
MAP ggtest.ORDER_LINE, TARGET ggtest.ORDER_LINE ;
MAP ggtest.ITEM, TARGET ggtest.ITEM ;
MAP ggtest.STOCK, TARGET ggtest.STOCK ;
```

Figure 5) Manager (mgr.prm) parameter file for both source and target systems.

```
PORT 7575
PURGEOLDEXTRACTS ./dirdat/*, USECHECKPOINTS
```

7.2   ORACLE GOLDENGATE PRIMARY EXTRACT

The following is sample session output from configuring the Primary Extract process that reads data from the transaction log.

Source System:

```
GGSCI (riker9.rtp.netapp.com) 8> dblogin userid ggtest password ggtest
Successfully logged into database.

GGSCI (riker9.rtp.netapp.com) 9> add trandata tpccdb.*

GGSCI (riker9.rtp.netapp.com) 10> add extract etpccdb, tranlog, begin now
2011-06-29 18:57:35  INFO    OGG-01749  Successfully registered EXTRACT ETPCCDB to
start managing log retention at SCN 14214354.
EXTRACT added.
GGSCI (riker9.rtp.netapp.com) 11> add exttrail ./dirdat/et, extract etpccdb, megabytes
100
EXTTRAIL added.
GGSCI (riker9.rtp.netapp.com) 12> info extract eora01 detail
```

```
EXTRACT      ETPCCDB    Initialized   2011-06-29 18:57    Status STOPPED
Checkpoint Lag          00:00:00 (updated 00:02:34 ago)
Log Read Checkpoint   Oracle Redo Logs
                        2011-06-29 18:57:57  Seqno 0, RBA 0


  Target Extract Trails:

  Remote Trail Name                                      Seqno       RBA      Max MB
  ./dirdat/et                                              0          0          5


  Extract Source                               Begin            End
  Not Available                                * Initialized *   2011-06-29 18:57

Current directory      /u35/filer1_gg1_vol1
Report file            /u35/filer1_gg1_vol1/dirrpt/ETPCCDB.rpt (does not yet exist)
Parameter file         /u35/filer1_gg1_vol1/dirprm/etpccdb.prm
Checkpoint file        /u35/filer1_gg1_vol1/dirchk/ETPCCDB.cpe
Process file           /u35/filer1_gg1_vol1/dirpcs/ETPCCDB.pce
Stdout file            /u35/filer1_gg1_vol1/dirout/ETPCCDB.out
Error log              /u35/filer1_gg1_vol1/ggserr.log


GGSCI (riker9.rtp.netapp.com) 13> start extract etpccdb
Sending START request to MANAGER ...
EXTRACT ETPCCDB starting
GGSCI (riker9.rtp.netapp.com) 14> info extract etpccdb


EXTRACT      EORA01    Last Started 2011-06-29 18:57   Status RUNNING
Checkpoint Lag          00:00:00 (updated 00:00:00 ago)
Log Read Checkpoint   Oracle Redo Logs
                        2011-06-29 18:57:57  Seqno 36, RBA 7419392
```

7.3   ORACLE GOLDENGATE EXTRACT DATA PUMP

The following is sample session output from configuring the Extract Data Pump process that reads data from the local trail that was written by the primary extract process.

Source System:

```
GGSCI (riker9.rtp.netapp.com) 16> add extract ptpccdb, exttrailsource ./dirdat/et
EXTRACT added.
GGSCI (riker9.rtp.netapp.com) 17> add rmttrail ./dirdat/pt, extract ptpccdb, megabytes
100
RMTTRAIL added.
GGSCI (riker9.rtp.netapp.com) 18> start extract ptpccdb
Sending START request to MANAGER ...
EXTRACT PTPCCDB starting



GGSCI (riker9.rtp.netapp.com) 19> info extract ptpccdb
EXTRACT      PTPCCDB    Last Started 2011-06-29 18:57    Status RUNNING
Checkpoint Lag          00:00:00 (updated 00:00:04 ago)
Log Read Checkpoint   File ./dirdat/et000000
                        First Record  RBA 0

GGSCI (riker9.rtp.netapp.com) 20> info all
Program      Status      Group      Lag            Time Since Chkpt
MANAGER      RUNNING
EXTRACT      STOPPED     ETPCCDB    00:00:00       00:00:19
EXTRACT      RUNNING     PTPCCDB    00:00:00       00:00:03
```

7.4 ORACLE GOLDENGATE REPLICAT

The following is sample session output from configuring the Replicat process that reads data that was written remotely by the Extract Data Pump process on the target system.

Target System:

```
GGSCI (riker10.rtp.netapp.com) 6> dblogin userid ggtest, password ggtest
Successfully logged into database.
GGSCI (riker10.rtp.netapp.com) 7> add checkpointtable ggschkpt
Successfully created checkpoint table GGSCHKPT.
GGSCI (riker10.rtp.netapp.com) 8> add replicat rtpccdb, exttrail ./dirdat/pt
REPLICAT added.
GGSCI (riker10.rtp.netapp.com) 9> start replicat rtpccdb
Sending START request to MANAGER ...
REPLICAT RTPCCDB starting

GGSCI (riker10.rtp.netapp.com) 10> info all
Program     Status       Group        Lag            Time Since Chkpt
MANAGER     RUNNING
REPLICAT    RUNNING      RTPCCDB      00:00:00       00:00:00
```

7.5 ORACLE GOLDENGATE REPORTING STATISTICS

The following is sample session output from process statistics and lag reporting of the primary extract data pump that runs on the source system and the Replicat process that runs on the target system.

Source Statistics:

```
GGSCI (riker9.rtp.netapp.com) 15> lag *TPCCDB
Sending GETLAG request to EXTRACT ETPCCDB ...
Last record lag: 1 seconds.
At EOF, no more records to process.

Sending GETLAG request to EXTRACT PTPCCDB ...
Last record lag: 4 seconds.
At EOF, no more records to process.
GGSCI (riker9.rtp.netapp.com) 16> stats ETPCCDB, table *, total
Sending STATS request to EXTRACT ETPCCDB ...
Start of Statistics at 2011-07-14 14:49:07.

Output to ./dirdat/et:

Extracting from TPCCDB.WAREHOUSE to TPCCDB.WAREHOUSE:

*** Total statistics since 2011-07-13 13:55:18 ***
        Total inserts                           0.00
        Total updates                      246534.00
        Total deletes                           0.00
        Total discards                          0.00
        Total operations                   246534.00

Extracting from TPCCDB.DISTRICT to TPCCDB.DISTRICT:

*** Total statistics since 2011-07-13 13:55:18 ***
        Total inserts                           0.00
        Total updates                      395285.00
        Total deletes                           0.00
        Total discards                          0.00
        Total operations                   395285.00

Extracting from TPCCDB.ORDER_LINE to TPCCDB.ORDER_LINE:
```

```
*** Total statistics since 2011-07-13 14:20:00 ***
        Total inserts                        1481508.00
        Total updates                        2944244.00
        Total deletes                              0.00
        Total discards                             0.00
        Total operations                     4425752.00

Extracting from TPCCDB.STOCK to TPCCDB.STOCK:

*** Total statistics since 2011-07-13 14:20:00 ***
        Total inserts                              0.00
        Total updates                        1481508.00
        Total deletes                              0.00
        Total discards                             0.00
        Total operations                     1481508.00
End of Statistics.
```

Target Statistics:

```
GGSCI (riker10.rtp.netapp.com) 43> lag RTPCCDB
Sending GETLAG request to REPLICAT RTPCCDB ...
Last record lag: 658 seconds.
At EOF, no more records to process.

GSCI (riker10.rtp.netapp.com) 44> stats RTPCCDB, table *, latest
Sending STATS request to REPLICAT RTPCCDBA ...
Start of Statistics at 2011-07-14 14:49:36.
Replicating from TPCCDB.DISTRICT to TPCCDB.DISTRICT:
*** Latest statistics since 2011-07-13 14:21:30 ***
        Total inserts                              0.00
        Total updates                         148930.00
        Total deletes                              0.00
        Total discards                             0.00

Replicating from TPCCDB.HISTORY to TPCCDB.HISTORY:

*** Latest statistics since 2011-07-14 14:12:28 ***
        Total inserts                          63356.00
        Total updates                              0.00
        Total deletes                              0.00
        Total discards                             0.00
        Total operations                       63356.00

End of Statistics.
```

8 CONCLUSION

This report documents the initial installation and configuration of Oracle GoldenGate software deployed on NetApp shared storage to perform data migration between heterogeneous databases. You can leverage NetApp storage efficiency technologies such as Snapshot, SnapRestore, and FlexClone once the target Oracle GoldenGate server configuration files and the database files are migrated to NetApp storage. Oracle GoldenGate, combined with NetApp storage technologies, empowers enterprises to better migrate and manage data.

9 REFERENCES

- [Oracle GoldenGate Administration Guide](#)
- [Oracle GoldenGate Installation and Setup Guide](#)
- [Oracle GoldenGate Reference Guide](#)
- Oracle Knowledge Base Doc ID 1232303.1—Oracle GoldenGate Best Practice: NFS Mount Options for Use with GoldenGate
- http://media.netapp.com/documents/ontap_software.pdf
- https://now.netapp.com/NOW/knowledge/docs/ontap/rel7351/html/ontap/smg/frameset.html

10 ACRONYMS

Table 1) Acronyms.

Term	Definition
DDL	Data Definition Language
GGSCI	GoldenGate Software Command Interface
LUN	Logical Unit Number
NAS	Network-attached Storage
NFS	Network File System
ODBC	Open Database Connectivity
SAN	Storage Area Network
SQL	Structured Query Language

Go further, faster®

www.netapp.com