



A hands-On guide to implement Real-Time Enterprise Data Warehouse (EDW)

**Implement Enterprise Real-Time Change Data  
Capture and Complex Data Transformations for  
In-Memory Analytical Applications**

---

# **The Enterprise Data Warehouse**

Design, build, deploy and manage an Enterprise Data Warehouse  
by empowering Oracle advanced Data Warehouse Technologies

# **Implement Enterprise Real-Time Change Data Capture & Advanced Data Transformations for In-Memory Analytical Applications**

---

Powered by Oracle Database 12c 12.2 & Oracle GoldenGate 12c 12.3

Release 1.0  
April 2018

**Copyright © 2018, Oracle and/or its affiliates. All rights reserved.**

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

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

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

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

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

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

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

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

# ABOUT THE BOOK

This book is an in-depth implementation of the Enterprise Data Warehouse (EDW). The objective is to demonstrate Oracle GoldenGate 12c and Oracle Database 12c advanced features to deliver three major capabilities essential to the Enterprise Data Warehouse (EDW).

- Enterprise Change Data Capture
- Complex Data Transformation
- In-Memory, Real-Time BI Applications

## Enterprise Change Data Capture and High-Performance Apply

Heterogeneous core systems run across the enterprise. The challenges surrounding moving and refreshing data from a diverse systems increase linearly for two reasons: Rolling out new core systems, deploying more analytical applications. With time the Enterprise Data Warehouse grow exponentially resulting on slower queries and diminished scalability. With Oracle GoldenGate 12c change data capture, low-impact and heterogeneous support overcome these challenges commonly found with Enterprise Data Warehouse (EDW). The high-performance apply empower large-scale parallel processing to speed-up the load-refresh phase.

## Complex Data Transformations

Moving data from the core systems must goes through data transformation processing, from basic-to-complex data re-structuring. There are multiple methods to achieve data transformation:

- Oracle GoldenGate 12c column mapping capabilities
- Oracle GoldenGate 12c built-in Functions
- Invoking query to access the local and remote database through database links
- Using user-defined database stored PL/SQL Procedures and Functions
- Using user-defined database stored Java classes

## In-Memory, Real-Time BI Applications

Oracle Database 12c In-Memory and Oracle GoldenGate 12c real-time change data capture and high-performance apply makes real-time BI application real. In addition, the Enterprise Data Warehouse is running on multi-instance database with advanced



data compression. Another major component to deliver real-time BI application is materialized view store compressed in memory. Other Oracle Database 12c Data Warehouse Features:

- Database partitioning
- Temporary groups
- Parallel queries processing
- Dedicated connections
- Online table re-definitions

# How is this Book Organized

## An Overview

The deployment roadmap for an Enterprise Data Warehouse (EDW) accomplished through a phased approach, each phase composed of several tasks. From systems architecture, an EDW composed of several systems, but the major systems are:

- The core systems, storing and manipulating transactional data
- Oracle GoldenGate 12c downstream systems, for change data capture
- An Enterprise Data Warehouse Repository, the very large database system

Figure 1 represents the roadmap for deploying an Enterprise Data Warehouse (EDW) for achieving real-time analytical applications.

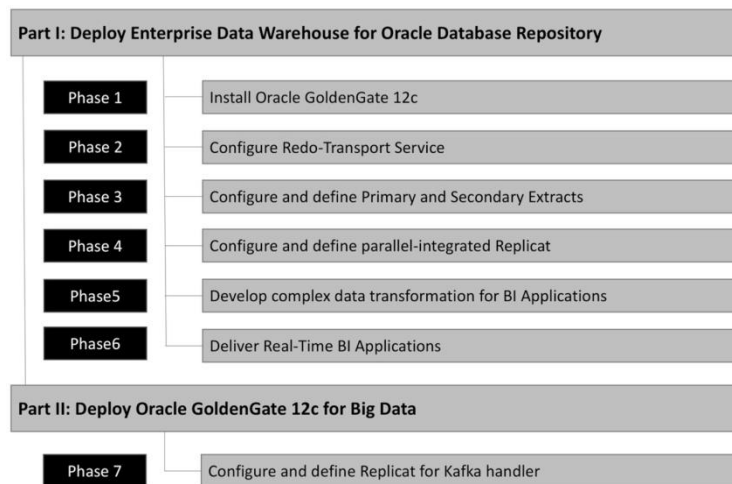


*Figure 1: Enterprise Data Warehouse Deployment*

The deployment of Enterprise Data Warehouse (EDW) divided into; Part I: deploy EDW on Oracle database and Part II: extend EDW to Big Data.

## Chapter 1: Introduction

The objective of this chapter is to present the landscape for deploying an Enterprise Data Warehouse (EDW). It begins by explaining the principle of Oracle GoldenGate 12c to deliver real-time enterprise change data capture, data transformations and real-time high-performance apply. Essential database design and configuration are explained, including handling large database blocks for high-performance I/O, database partitioning for VLDB environment, database in-memory option, advanced compression, materialized view for query re-write and parallelism for high-performance EDW load processing.

## Chapter 2: Install Oracle GoldenGate 12c

Install and prepare Oracle GoldenGate 12c instances for an Enterprise Data Warehouse (EDW) environment. There are two methods for installing Oracle GoldenGate 12c: Interactive and silent modes. This book follows the silent mode to install Oracle GoldenGate 12c on the downstream and EDW system. The interactive mode is outlined in Appendix E: Environment Setup. Certification matrix is critical for the following combination of Oracle Database releases.

|                          |                                      |
|--------------------------|--------------------------------------|
| <b>Core system</b>       | Oracle Database 11g release 11.2.0.4 |
| <b>Downstream system</b> | Oracle Database 12c release 12.1     |
| <b>EDW system</b>        | Oracle Database 12c release 12.2     |

## Chapter 3: Database Preparations

Prepare the source, the downstream and the target database for Oracle GoldenGate 12c. The source database *must* be operating on archive log mode, with minimum supplemental logging is enabled. It's highly recommended the downstream database mining server operates on archive log mode. Setting parameters on the downstream and target database to support data replication. It's essential to create a dedicated Oracle GoldenGate 12c database administrator on source, downstream and target database. And, finally create the table-level supplemental logging using remote GGSCI connection.

## Chapter 4: Configure Redo-Transport Service

The redo transport service ships the database transaction logs from the source database to the Oracle GoldenGate 12c downstream instance, which in turn has a database mining engine to build the logical change records (LCRs) for the integrated extract to perform high-performance change data capture (CDC). The configuration

begins by re-creating the password file as the password must be identical on the core and downstream systems. Then setting the database parameters to enable redo-transport by the source database and redo receiving by the downstream database. Creating the online standby redo log files for storing the transactions logs for the database mining engine. Because the redo-transport service is transported using Oracle SQLNET, configure database connect strings: from source to downstream and from downstream to source systems.

## **Chapter 5: Deploy Integrated Extract**

Begin by registering the integrated extract (IE), then configure and defined the integrated extract, its trail and the data pump for data streaming. The database mining server is generating the logical change records (LCRs) in real-time, the integrated extract performs change data capture (CDC) and write to the extract trail (local trail), which is consumed by the data pump process and compressed, encrypted and sent to the target (EDW) instance as remote trail.

## **Chapter 6: Configure Integrated Apply**

Configure parallel and integrated Replicat. Because the Replicat checkpoint is database table, configure and define the checkpoint table from GGSCI. The Enterprise Data Warehouse (EDW) must be able to perform high-performance apply operation using built-in parallel processing. Run the workload simulator on the source database and ensure data synchronization on the EDW database.

## **Chapter 7: Develop Complex Data Transformations for BI Applications**

The transformation empowers Oracle GoldenGate 12c capabilities to invoke user-defined stored database PL/SQL procedure for performing basic-to-advanced data transformation on heavily de-normalized database table – CALL\_DETAILS. The PL/SQL procedure further extends the capabilities by invoking stored database PL/SQL functions and stored database Java functions. It's essential to validate the data synchronization at this stage and data transformations are accurate.

## **Chapter 8: Deliver Real-Time BI Applications**

By empowering Oracle Database 12c features and options to deliver high-performance real-time Enterprise Data Warehouse Repository, capable to deliver real-time analytical application platform. This chapter re-configures the EDW database for intensive queries. Configuring large block I/O is essential where the EDW must be

operating using 16K-32K block size. Implement Oracle database partitioning, query re-write, in-memory database option and compression are extremely essential for the Enterprise Data Warehouse (EDW). Parallel processing is achieved at various levels: Database session, embedded hits and database parameters settings. And, finally comparing the execution plan for the same BI applications running on the source database and EDW database, the result is expected to be real-time on the Enterprise Data Warehouse system, and non-real-time on the source database system.

## Chapter 9: Oracle GoldenGate for Big Data

This chapter demonstrates Oracle GoldenGate for Big Data, for Kafka handler. Kafka is installed on dedicated server, ensuring real-time data steaming. Oracle GoldenGate 12c for big data receives the trail sent by the data pump; the Replicat ingests Kafka with the transactional data from the core systems. The final task is exporting data from Kafka to Oracle Database 12c for marketing applications.

The book contains several related appendixes; they form integral parts of the book.

## Appendix A: Oracle GoldenGate Fundamentals

Learn Oracle GoldenGate 12c fundamentals, covering the family of products, the architecture and building blocks of Oracle GoldenGate 12c. Configure and define extracts and Replicat. Learn about Oracle GoldenGate 12c uses cases, data selection, filtering and data transformations. Introduction to more advanced topics such as call-out to database stored procedures and C/C++ external programs. Configure Oracle GoldenGate 12c for access to database on Automatic Storage Management (ASM).

## Appendix B: Enterprise Data Warehouse from an Oracle Standby Database

Sometimes, due mainly to security reasons, installation of additional software in the production system is not an option, when this is the case, the alternative is to empower Oracle auto-managed standby database as the source of Oracle GoldenGate 12c change data capture (CDC).

There are two modes to deploy and operate Oracle GoldenGate 12c on an auto-managed standby database:

- Oracle GoldenGate 12c running on *Active Data Guard* mode
- Oracle GoldenGate 12c running on *Archive Only Log* (AOL) mode

If the business requirements require real-time change data capture, then active data guard mode is the option, for non-real-time archive only log (AOL) is an option and can operate on the primary or the standby archive log files.

## **Appendix C: Migration the Enterprise Data Warehouse to Oracle Cloud IaaS**

The journey to Oracle cloud is fascinating, shifting the workload to Oracle cloud Infrastructure as a Service (IaaS) is an objective setting on the table of many decision makers. The ultimate reasons are cost reduction and elastic platform that expands and shrinks based on business needs. The Enterprise Data Warehouse (EDW) is a perfect candidate for the migration to Oracle Cloud Infrastructure as a Service (IaaS).

This appendix demonstrates the shift and lift of an on-premise EDW to Oracle cloud IaaS. In order to support business continuity, the adaption of an online solution for the shift and left to Oracle cloud IaaS is developed. To achieve this, the online shift and lift empowers Oracle GoldenGate 12c downstream technology and Oracle Recovery Manager (RMAN) to transparently move the EDW to Oracle Cloud IaaS.

## **Appendix D: Automatic Conflict Detection and Resolution for Active-Active Database Availability Architecture**

With the announcement of Oracle GoldenGate 12c version 12.3, the deployment of active-active database for high-availability has been greatly automated. Design, build and deploy n-way Active-Active database architecture, where data streaming happens on all directions, empowers the automated build-in automated conflict detection and resolution and optimized data (n-1)-way re-streaming to achieve symmetrically distributed database environments.

Data conflict occurs when a replicated row updated nearly at the same time, or following network restore after an extended time of failure. The resolution is implemented in the database and not in Oracle GoldenGate. The conflict enabled using Oracle database API calls, which create an invisible database column to implement and enforce the resolution method.

## **Appendix E: Environment Setup**

Nowadays, the vast majority of production environments are built using virtualization; the predominant reasons are less cost and manageability. This environment is built using Oracle virtualization solution, Oracle VirtualBox.

The Enterprise Data Warehouse (EDW) environment composed of four systems:

- The primary database system
- The standby database system
- The target database system
- The big data Kafka system

## **Appendix F: Database Design and Workload Simulator**

The purpose of the workload simulator is to generate transactions volumes sustainable for the duration of the project phases. The settings of the simulator is adjustable and should be set to match the real system if resources allows, or reduced proportionally in case it's running on low resource testing environment. The workload simulator API developed using PL/SQL; hence it's stored in the source database and accessible to users by granting the EXECUTE privilege only.

There are four workload levels; each designed to suite the minimum available system resources. Running a workload beyond the available resources of processors and disk space, may cause system slowness or crash. Always start with low workload, and uplift the workload as needed. This workload simulator provides the following levels:

- Low-level workload
- Medium-level workload
- High-level workload
- Extreme-level workload

## **Appendix G: Glossary**

Lists the definition of terms related to the Enterprise Data Warehouse (EDW).

## **The Hands-On Labs**

The hands-on labs composed of several workshops. The workshops must be completed on the order listed below.

### **Workshop 1: Install Oracle GoldenGate 12c**

This workshop installs Oracle GoldenGate 12c on the following systems:

- Oracle GoldenGate 12c Downstream system
- Enterprise Data Warehouse (EDW) system

Table 1 lists the tasks for installing Oracle GoldenGate 12c across these systems.

| Task No | Description                                                  |
|---------|--------------------------------------------------------------|
| 1       | Install Oracle GoldenGate 12c on Downstream System           |
| 2       | Verify Oracle GoldenGate 12c instance on Downstream System   |
| 3       | Optimize the Manager process parameters on Downstream System |
| 4       | Install Oracle GoldenGate 12c on Target EDW System           |
| 5       | Verify Oracle GoldenGate 12c instance on Target EDW System   |
| 6       | Optimize the Manager process parameters on Target EDW System |

*Table 1: Tasks for installing Oracle GoldenGate 12c*

## Workshop 2: Database Preparation

This workshop prepares the source, downstream and the target EDW database for Oracle GoldenGate 12c change data capture and apply.

Table 2 lists the tasks associating with preparing the source, downstream and target EDW database.

| Task No | Description                                                    |
|---------|----------------------------------------------------------------|
| 1       | Verify the Source Database Operating Mode                      |
| 2       | Enable Database Supplemental Logging on Source Database        |
| 3       | Create GoldenGate Database Administration Schema on Source     |
| 4       | Activate Archive Log Mode on Downstream                        |
| 5       | Create GoldenGate Database Administration Schema on Downstream |
| 6       | Database Parameters Setting on Downstream                      |
| 7       | Verify GoldenGate Database Login for Downstream                |
| 8       | Create GoldenGate Administration Schema on Target EDW          |
| 9       | Verify GoldenGate Database Login for Target                    |
| 10      | Database Parameters Setting on Target EDW                      |
| 11      | Enable GoldenGate table-level supplemental logging             |

*Table 2: Tasks for preparing the source, downstream and target systems*

## Workshop 3: Configure Redo-Transport Service

This workshop configures the redo-transport service from the source database to the database mining server for performing change data capture.

Table 3 lists the tasks associating with configuring Redo-Transport Service.



| Task No | Description                                  |
|---------|----------------------------------------------|
| 1       | Run the Workload Simulator                   |
| 2       | Configure and Test the Password File         |
| 3       | Configure SQLNET Listener and Client         |
| 4       | Configure the Source Database Parameters     |
| 5       | Configure the Downstream Database Parameters |
| 6       | Create the Standby Online Redo Log Files     |
| 7       | Create the Directory Structure               |
| 8       | Verify Redo-Transport Service                |

*Table 3: Tasks for Configuring Redo-Transport Service*

## Workshop 4: Deploy Integrated Extract

This workshop deploys Oracle GoldenGate 12c integrated extract (IE). Register the integrated extract, define and configure the primary and secondary extract. Define the extract trail and remote trail.

Table 4 lists the tasks associating with deploying integrated extract.

| Task No | Description                                 |
|---------|---------------------------------------------|
| 1       | Register the Integrated Extract             |
| 2       | Configure the Integrated Extract            |
| 3       | Define the Integrated Extract               |
| 4       | Configure the Secondary Extract (data pump) |
| 5       | Define the Secondary Extract (data pump)    |

*Table 4: Tasks for Deploying Integrated Extract*

## Workshop 5: Deploy Integrated Apply

This workshop deploys Oracle GoldenGate 12c parallel, integrated Replicat. Start the extract and Replicat to verify an end-to-end data synchronization.

Table 5 lists the tasks associating with deploying integrated apply.

| Task No | Description                                |
|---------|--------------------------------------------|
| 1       | Configure the Checkpoint Table             |
| 2       | Create the Checkpoint Table                |
| 3       | Configure the Parallel-Integrated Replicat |
| 4       | Define the Parallel-Integrated Replicat    |
| 5       | Start Extracts                             |
| 6       | Start Replicat                             |

| Task No | Description                 |
|---------|-----------------------------|
| 7       | Verify Data Synchronization |

*Table 5: Tasks for Deploy Integrated Apply*

## Workshop 6: Develop Complex Data Transformations for BI Applications

This workshop conducts complex data transformations by calling stored database PL/SQL procedures, PL/SQL functions and Java classes. Start the workload simulator and verify an end-to-end data synchronization.

Table 6 lists the tasks associating with deploying integrated apply.

| Task No | Description                                        |
|---------|----------------------------------------------------|
| 1       | Develop the PL/SQL Transformation Procedure        |
| 2       | Grant Database Privileges to Execute the Procedure |
| 3       | Re-Configure the Parallel-Integrated Replicat      |
| 4       | Start the Workload Simulator                       |
| 5       | Verify Data Synchronization                        |
| 6       | Verify Data Transformations                        |

*Table 6: Tasks for Develop Complex Data Transformation for BI Applications*

## Workshop 7: Delivering Real-Time BI Applications

This workshop deploys Oracle Database 12c features to support real-time BI applications. Key features include: database partitioning, materialized view for query re-write, in-memory database, advanced compression and parallel processing. The workshop compares the result of running the same BI application on the source database and on the EDW database. The objective is to achieve real-time runtime on the EDW database.

Table 7 lists the tasks associating with delivering Real-Time BI applications

| Task No | Description                                             |
|---------|---------------------------------------------------------|
| 1       | Setting Optimizer Parameters for Data Warehouse Queries |
| 2       | Implement 16K-32K block I/O                             |
| 3       | Implement Temporary Groups                              |
| 4       | Online Database Structure Partitioning                  |
| 5       | Deploy Materialized View for Query Re-Write             |
| 6       | Implement In-Memory Database Options                    |

| Task No | Description                                   |
|---------|-----------------------------------------------|
| 7       | Implement Advanced Compression for DSS        |
| 8       | Implement Parallel Queries Outlines           |
| 9       | Run the Analytical Applications on Production |
| 10      | Run the Analytical Applications on EDW        |
| 11      | Compare the Runtime Execution Plan            |

*Table 7: Tasks for Delivering Real-Time BI Applications*

## Workshop 8: Oracle GoldenGate for Big Data

This workshop empowers Oracle GoldenGate 12c for big data to ingest data into Kafka. The workshop begins by installing Kafka and Oracle GoldenGate 12c for big data. Test the operability of Kafka, and then configure Oracle GoldenGate 12c for big data to deliver real-time transactions to Kafka.

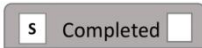
Table 8 lists the tasks associating with ingesting Kafka.

| Task No | Description                                                             |
|---------|-------------------------------------------------------------------------|
| 1       | Install Kafka on Oracle Linux                                           |
| 2       | Install Oracle GoldenGate 12c for Big Data                              |
| 3       | Test the Function of Kafka for Real-Time Messaging                      |
| 4       | Configure and Define Oracle GoldenGate 12c Replicat for Kafka Ingestion |
| 5       | Run the Workload Simulator                                              |
| 6       | Verify Data Ingestion                                                   |

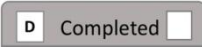
*Table 8: Tasks associated with ingesting Kafka*

# Conventions

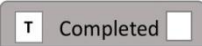
There are three symbolic conventions used throughout the workshops:



Designates the step on the source database system

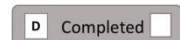


Designates the step on downstream database system



Designates the step on Enterprise Data Warehouse (EDW) system

The workshop commands are executed interactively or via a Linux shell scripts. For interactive mode, execute the command in bold only.



```
[oracle@dss01 gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE_12.3.0.1.0_PLATFORMS_170721.0154_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (dss01.mobily.com.sa) 2> miningdblogin userid ogg_admin@dsdb, password oracle
Successfully logged into mining database.

GGSCI (dss01.mobily.com.sa) 3> dblogin userid ogg_admin@BRM, password oracle
Successfully logged into database.

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 4>
```



Successful command indicator

The interactive utilities used throughout the workshop:

- SQL>** SQL\*Plus
- LSNRCTL>** Oracle Listener Command Interface
- GGSCI>** GoldenGate Software Command Interface
- vi** Linux Editor
- #** Root prompt
- \$** Oracle prompt
- SRVCTL>** Grid control prompt

# An Introduction

## Overview

The approach of designing, building and deploying an Enterprise Data Warehouse (EDW) database is complex, where the predominant deliverable is to achieve high-performance engine for Business Intelligence (BI) and analytical applications. This chapter outlines the engineering of real-time EDW capable of supporting very hostile SQL queries. When combining mixed workloads queries, it's always the business users who are affected most, and the analytical applications are not running within the expected throughput, directly impacting the decision making process.

In addition, when the business is driven by a wide-range of online and historical data growing linearly, it entails the physical separation of queries, where the pattern of handling EDW queries is super-charged with the database Online Analytical Processing (OLAP) capabilities making the EDW repository fit-for-purpose. Oracle Database 12c and Oracle GoldenGate 12c are engineered to establish a real-time Enterprise Data Warehouse (EDW) platform; this chapter discusses the following concepts, which are explained and implemented in subsequent chapters.

- Real-time change data capture
- Real-time data transformations
- Real-time high-performance apply
- Database Large Block I/O
- Database Partitioning
- Database In-Memory
- Data Compression
- Materialized Views
- Parallelism

Nonetheless, addressing the compute capacity, elasticity, and linear storage growth are equally vital elements to avoid interval system panicking to deliver non-stop, scalable and high-performance business operations

# Architecture

A major requirement found within the enterprise is to eliminate additional overhead on the core systems that are feeding the Enterprise Data Warehouse (EDW) database repository; this has been addressed using the following Oracle technologies:

- Deploying Oracle GoldenGate 12c Downstream System
- Deploying Oracle Database 12c Active Data Guard

Figure 1 depicts deploying Oracle GoldenGate 12c Downstream System. This technology delivers security, performance and real-time processing.

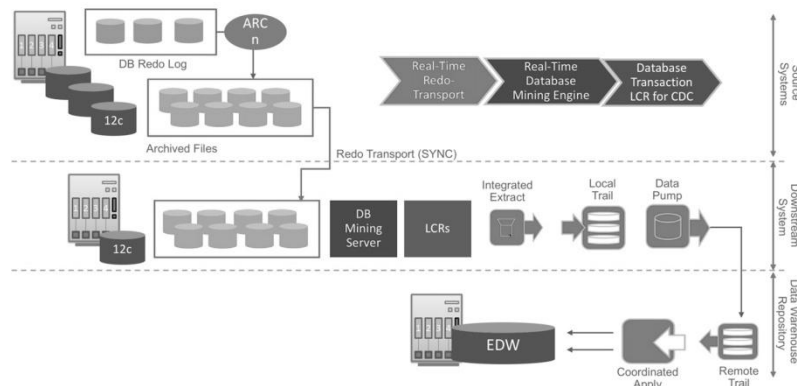


*Figure 1: Oracle GoldenGate 12c Downstream System*

The downstream system receives redo-transport from the core Oracle system, which in turn performs database mining to build the stream of Logical Change Records (LCRs). The integrated extract performs real-time change data capture, which produces the primary extract's trail to be consumed by the secondary extract for data streaming to the remote system, the Enterprise Data Warehouse (EDW).

On the remote system, the remote trails are created to be consumed by a high-performance parallel-integrated apply Replicat. In an Enterprise Data Warehouse (EDW) basic-to-advanced data transformations is unavoidable. Though data transformations may occur anywhere within the data streaming, our intention is to develop the data transformations module to be self-contained within the apply process. Using the database PL/SQL language, call-out to data transformations stored database procedure returns the data on the final data structure required by the analytical applications. The Oracle GoldenGate 12c downstream system is location transparent; it uses Oracle SQLNET to ship the redo-transport from the core system to the downstream system.

Figure 2 depicts Oracle GoldenGate 12c performing near-real time change data capture from Oracle Database 12c active data guard.

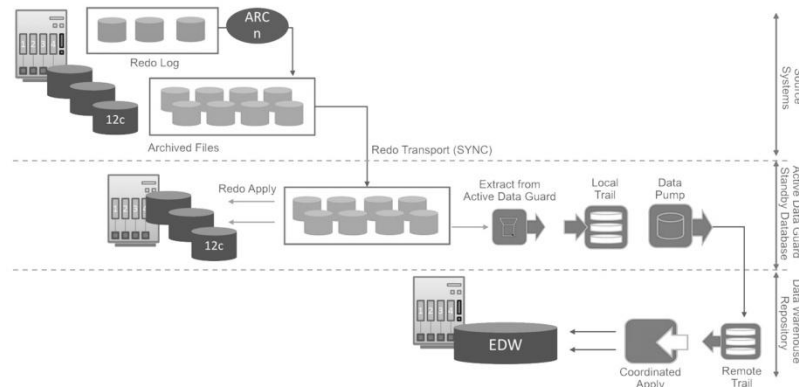


Figure 2: Change Data Capture from Oracle Database 12c Active Data Guard

The Oracle GoldenGate 12c Archive Only Log (AOL) extract perform change data capture from the archive log files. Because of the log switch occurs on the core system based on interval, the AOL change data capture is not on real time. The normal log switch occurs every 20-30 minutes, depending on the workloads.

## Real-Time Change Data Capture

There are number of replication software in the market claiming real-time change data capture capabilities, however; for the majority of them that is not totally a true statement. Oracle GoldenGate 12c release 12.3 has proven to be the software capable of delivering data replication solutions that is:

- Real-time change data capture (CDC)
- None intrusive, reading the transaction logs from online redo logs
- Extendable through call-out to external C/C++ programs and PL/SQL modules
- Online basic-to-complex data transformations

There are two methods of change data capture (CDC) using Oracle GoldenGate 12c: Classic and Integrated. The *classic* method is mainly for heterogeneous environment, where support for the major databases is available. For example, database replication between Oracle Database and MySQL database. The *integrated* method is available for Oracle-to-Oracle only, its Oracle preferred approach for database replication. The integrated capture is deployable *locally* or on a *downstream* system. When deployed locally, Oracle GoldenGate 12c uses the local instance database mining server,

however; when deploy as downstream, it empowers the downstream mining server for performing change data capture. Figure 3 depicts the deployment methods for integrated capture.

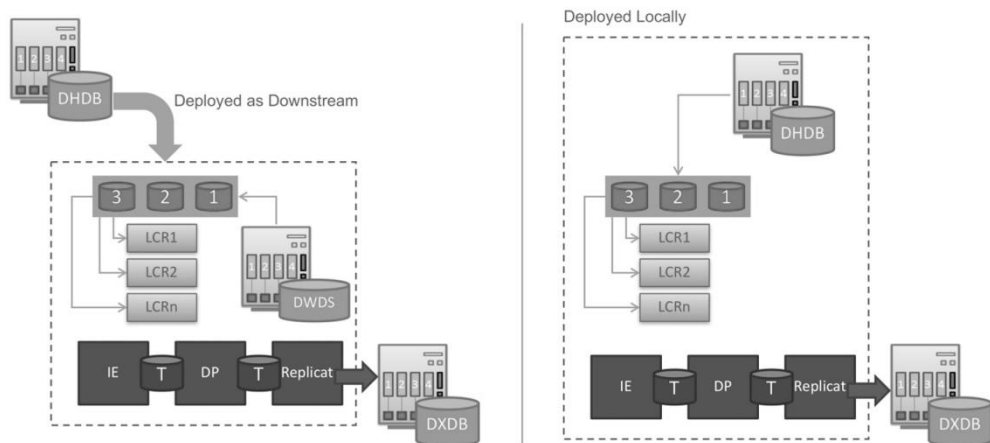


Figure 3: Oracle GoldenGate 12c Integrated Capture Options

Due mostly to security and performance, the preferred approach for an enterprise is to deploy Oracle GoldenGate 12c on a downstream system for their Enterprise Data Warehouse (EDW), this document strictly focus on Oracle GoldenGate 12c deployed on a downstream system.

## Oracle Change Data Capture Options

The classic Change Data Capture (CDC) method has two options:

- Real-time change data capture from the transaction logs (online redo log files)
- Non-Real time change data capture from the archive log files

While the fast majority of the applications require real-time change data capture capabilities, some applications including EDW may meet their business and technical requirements using non-real time change data capture from the archive log files. As explained earlier, though the real-time change data capture is non-intrusive process; however, if the need to completely eliminate resource overhead on the core system, then non-real time change data capture from the standby database archive log files is the option. Notice, the non-real time change data capture from the archive log files can be delivered from the core system archive log file, or from the Oracle Database 12c Data Guard instance standby archive log file as illustrated in Figure 4. In addition to performing real-time change data capture, parallel change data capture is available,



where multiple extracts are deployed to support parallel processing capturing data from the same database table.

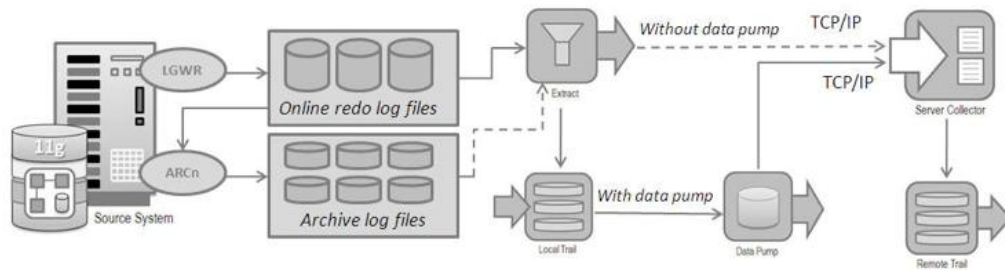


Figure 4: Change Data Capture Options

## Data Heterogeneity

Support of change data capture from heterogeneous core systems is essential for any Enterprise Data Warehouse project. Oracle GoldenGate 12c 12.3 support change data capture from the major commercial and open source databases. Table 1 lists the databases supported by Oracle GoldenGate 12c release 12.3 at the time of writing the book:

| Database   | Operating Systems                                                |
|------------|------------------------------------------------------------------|
| Oracle     | Linux x86-64, Windows x86-64, Solaris Sparc, AIX                 |
| DB2        | Linux x86-64, Windows x86-64, i-Series, AS400, AIX, Solaris, zOS |
| MySQL      | Linux x86-64, Windows x86-64                                     |
| SQL Server | Linux x86-64, Windows x86-64                                     |
| Teradata   | Linux x86-64, Windows x86-64, Solaris Sparc, AIX                 |

Table 1: Databases Supported by Oracle GoldenGate 12c

## Real-Time Data Transformations

Data transformations are essential for Enterprise Data Warehouse (EDW) projects. Oracle GoldenGate 12c provides techniques and functions to perform any form, any where of data transformations. Data transformations may occur at the extract, data pump or Replicat. For an Enterprise Data Warehouse (EDW) projects with an extreme data transformations, an Oracle GoldenGate 12c intermediate system dedicated for data transformations is an optimum solution. Figure 5 shows Oracle GoldenGate 12c intermediate system topology. In the event the transformations by the intermediate system does not requires database table lookup operations and solely using Oracle GoldenGate 12c functions, then a database is not needed, only the Oracle Database 12c binaries is required.

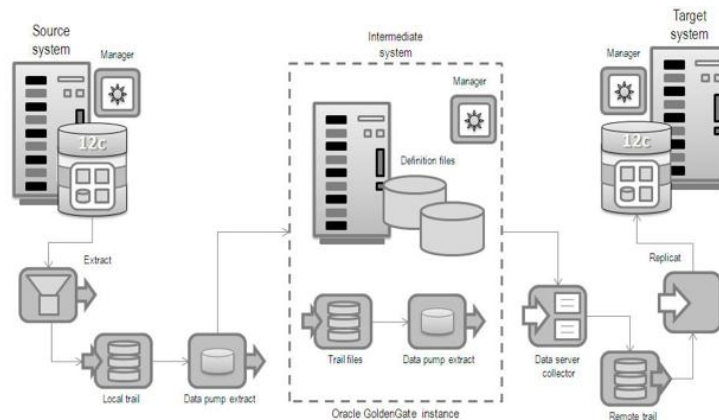


Figure 5: Oracle GoldenGate 12c intermediate system

## Oracle GoldenGate 12c Functions

Instead of using the database functions to perform data transformations, using Oracle GoldenGate 12c functions enable higher performance. Table 2 lists the major Oracle GoldenGate 12c functions and their description.

| Function  | Description                                                                                                              |
|-----------|--------------------------------------------------------------------------------------------------------------------------|
| @COMPUTE  | Returns the result of arithmetic expressions into a target column. The return value is of type string.                   |
| @NUMBIN   | Converts a binary string into number.                                                                                    |
| @NUMSTR   | Converts a string to number.                                                                                             |
| @STRCAT   | Concatenates sequence of strings.                                                                                        |
| @STRCMP   | Compares two strings to determine whether they are equal, or the first string is less or greater than the second string. |
| @STREQ    | Tests two strings to see if they equal or not. Return 1 for equal strings, returns 0 for not equal strings.              |
| @STREXT   | Returns substring from a string indicated by begin and an end positions.                                                 |
| @STRFIND  | Returns the first occurrence of a substring within a string.                                                             |
| @STRLEN   | Returns the length of a string.                                                                                          |
| @STRTRIM  | Removes the leading spaces from a string.                                                                                |
| @STRNCAT  | Concatenates one or more strings up to the specified number of characters per string.                                    |
| @STRNCMP  | Compares two strings up to specific number of characters.                                                                |
| @STRRTRIM | Trims the trailing spaces in a string.                                                                                   |
| @STRSUB   | Substitutes one string for another within a string.                                                                      |
| @STRTRIM  | Trims both leading and trailing spaces in a string.                                                                      |

|         |                                                                                                                             |
|---------|-----------------------------------------------------------------------------------------------------------------------------|
| @STRUP  | Converts the string to uppercase.                                                                                           |
| @GETENV | Returns Oracle GoldenGate environment variable value.                                                                       |
| @GETVAL | Extracts parameter value from a stored procedure when using SQLEXEC clause, requires the name procedure and parameter name. |
| @RANGE  | Used to provide parallel processing for enabling multiple secondary extract or replicate processes read the same trail.     |

Table 2: Oracle GoldenGate 12c Functions

## Functions by Examples

The following converts the customer address to upper case for the target column.

```
MAP osm$repapi.customers, TARGET osm$repapi.customers, &
COLMAP (USEDEFAULTS,
cust_addr = @STRUP(cust_address));
```

The following validates the email address contains the '@' character.

```
MAP osm$repapi.customers, TARGET osm$repapi.customers, &
COLMAP (USEDEFAULTS,
cust_addr = cust_address,
cust_email = @STRFIND (CUST_EMAIL, "@") > 0);
```

Construct an upper case email by concatenating the name with specific domain. The upper case email achieved by nesting the two function @STRCAT and @STRUP.

```
MAP osm$repapi.customers, TARGET osm$repapi.customers,
COLMAP (USEDEFAULTS,
cust_email=@STRUP(@STRCAT(cust_name, '@oracle.com')));
```

Use the current date for the column CUST\_DATE using the function @DATENOW.

```
MAP osm$repapi.customers, TARGET osm$repapi.customers, &
COLMAP (USEDEFAULTS, &
cust_addr = cust_address, cust_date = @DATENOW());
```

## SQLEXEC Technique

For advanced data transformations, Oracle GoldenGate 12c allows you to execute an SQL Query or invoke PL/SQL procedure. The values returns from the query are assigned to the target columns, which maybe part of complex regular expression, for example. The values returned by invoking a stored procedure mainly for developing advanced and complex data transformations. Figure 6 shows the depth of data transformations that may be developed using database stored procedure.

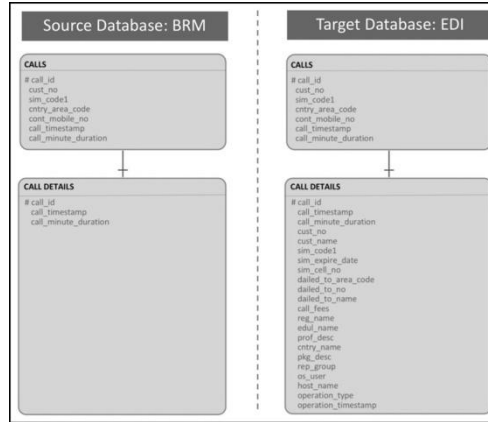


Figure 6: Data Transformations using Oracle GoldenGate 12c

The stored procedure expects one or more input parameters, and should return one or more output parameters assigned to target columns.

## Real-Time High-Performance Apply

The high-performance apply is an essential component for the Enterprise Data Warehouse (EDW). Data captured from multiple core systems, streamed in real-time to the EDW must be applied using high-performance apply process. Oracle GoldenGate 12c provides several apply process. Table 3 compare the types of apply processes configurable as a Replicat. There are 5 types of Replicat:

- Classic
- Coordinated
- Integrated
- Parallel Non-Integrated
- Parallel Integrated

## Where Data Transformations Occur?

If the target platform is an Oracle Database, then the Replicat type is any of the above listed Replicats, however; for non-Oracle database only the classic type is applicable. However; data transformations can occur anywhere, it may occur at the Extract, data pump or the Replicat. For an Enterprise Data Warehouse, the best practice is to perform data transformations either using an intermediate system or the Replicat, the source system is always offloaded from data transformations activities.

|                                    | Replicat |             |            |                         |                     |
|------------------------------------|----------|-------------|------------|-------------------------|---------------------|
|                                    | Classic  | Coordinated | Integrated | Parallel Non-Integrated | Parallel Integrated |
| Batch Processing (BATCHSQL)        | ✓        | ✓           | ✓          | ✓                       | ✓                   |
| Barrier Transactions               | ✗        | ✓           | ✓          | ✓                       | ✓                   |
| Dependency Computation             | ✗        | ✗           | ✓          | ✓                       | ✓                   |
| Auto-Parallelism                   | ✗        | ✗           | ✓          | ✓                       | ✓                   |
| DML Handler                        | ✗        | ✗           | ✓          | ✗                       | ✓                   |
| Auto-CDR                           | ✗        | ✗           | ✓          | ✗                       | ✓                   |
| Procedural Replication             | ✗        | ✗           | ✓          | ✗                       | ✓                   |
| Dependency aware Transaction Split | ✗        | ✗           | ✗          | ✓                       | ✓                   |
| Cross RAC node processing          | ✗        | ✓           | ✗          | ✓                       | ✓                   |

*Table 3: Oracle GoldenGate 12c Replicat Types and features*

## Classic Replicat

An early version type of Replicat to support basic heterogeneous applies processes. Support for high-performance array processing through a single parameter – BATCHSQL. With Classic Replicat the parallelism is achieved by manually configuring and defining multiple Replicat processes through the use of the @RANGE function.

## Coordinated Replicat

The coordinated apply is flexible. It's configured using one parameter file, and when defined using the ADD REPLICAT command, Oracle GoldenGate 12c creates 25 Replicats. These Replicats are given internal names and coordinated when applying data to the target database.

## Integrated Replicat

This type of Replicat primary features are Automatic Conflict Detection and Resolution (CDR) and procedural replication. The automatic parallelism allows coordinated Replicat features.

## Parallel Non-Integrated Replicat

This type of Replicat is mainly for parallel processing; which split the data (trail) into several parts, automatically assign the Replicat and apply the transactions to the target database without compromising data consistency.

# Parallel Integrated

This is a high-performance Replicat, it has all the features associated with Oracle GoldenGate 12c apply groups. It delivers automatic parallelism, automatic Conflict Detection and Resolution (CDR), procedural replication and support for cross node processing on a Real Application Cluster (RAC) environment.

## Working with Database Large Block I/O

The default Oracle Database block size for reading and writing to the database is 8K (8096 bytes). However; Oracle Database is capable to support 5 sizes for block I/O. Oracle database support 2K, 4K, 8K, 16K and 32K as depicted in Figure 7.

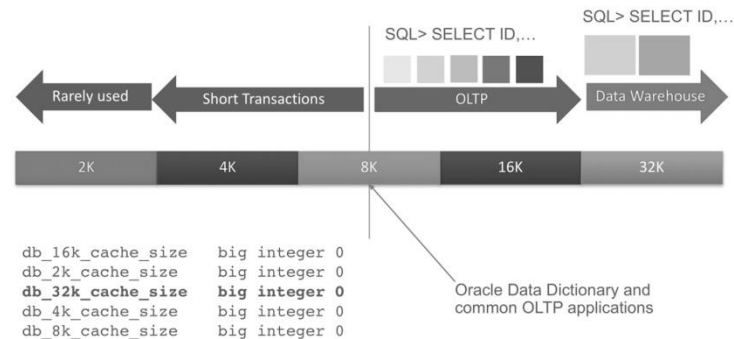


Figure 7: Oracle Database Block I/O Decision Line

The database buffer for each block sized using its database parameter as listed in table 4. These parameters are dynamic, hence does not require database restart to activate them.

| Parameter        | Description                                                                                                                                                                                 |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| db_2k_cache_size | This parameter is for sizing a non-default database buffer of size 2K. This considered relatively small and may be used for specialized embedded applications only.                         |
| db_4k_cache_size | This parameter is for sizing a non-default database buffer of size 4K. Candidate database applications include training, development and testing.                                           |
| db_8k_cache_size | This parameter is for sizing a non-default database buffer of size 8K. This is the default and sound optimum for the majority of database applications. The Oracle Database data dictionary |

|                   |                                                                                                                                     |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------|
|                   | is using 8K block size.                                                                                                             |
| db_16k_cache_size | This parameter is for sizing a non-default database buffer of size 16K. Suitable for data mart and data warehouse repository.       |
| db_32k_cache_size | This parameter is for sizing a non-default database buffer of size 32K. Suitable for an Enterprise Data Warehouse (EDW) repository. |

Table 4: Large Block I/O Parameters Setting

## Configuring Large Database Block I/O

Prior to configuring large block I/O, enough Oracle memory must have been allocated, bounded by memory ceiling enabled by MEMORY\_MAX\_TARGET parameter. The steps to configure large block I/O are outlined below:

1. Enlarge the maximum memory target for the SGA
2. Set the parameter DB\_16K\_CACHE\_SIZE or DB\_32K\_CACHE\_SIZE
3. Create tablespace using 16K or 32K block size
4. Migrate tables to the new tablespace

Now, tables that are referenced by the EDW analytical applications are fetched using 16K or 32K I/O.

Refer to Chapter 8 to learn more about configuring large database block I/O.

## Configuring Big-File Tablespace

The default tablespace file restriction is of type SMALL file, which enables a tablespace to be composed of one or more data files. However, the BIG file tablespace type composed one-and-only data file, with a maximum size of 128TB. The Big-File tablespace is designed for data warehouse database, to avoid running out of extents allocation, ensure to specify the AUTOEXTEND option when creating the Big-File tablespace.

Refer to Chapter 8 to learn more about configuring large database block I/O.

## Database Partitioning

We must partition the volatile data warehouse tables. The database partitioning was introduced on Oracle 8 with three basic types of partitioning: *Range*, *hash* and *list*. Since then, the database partitioning technology evolved across Oracle database releases.

Figure 8 illustrates the data model of table and index partitioning. A partitioned table or index logically resides on one-or-more tablespaces, which in turn stored in one-or-more data files. As shown, the data files may be any of the supported storage such as Automatic Storage Management (ASM), file system, NFS, etc.

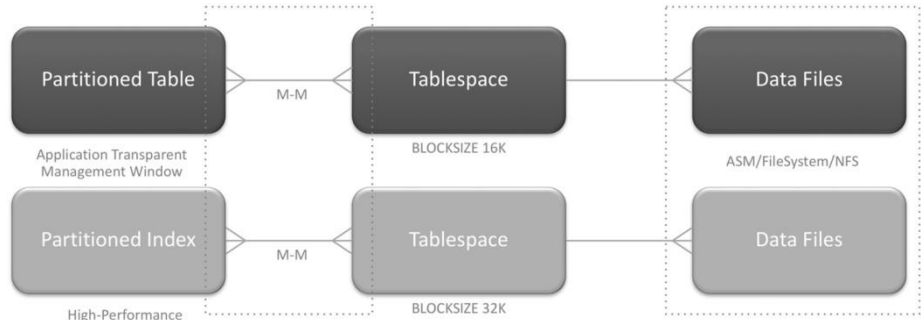


Figure 8: Partitioning Data Model

Currently, Oracle Database 12c Release 2 supports 19 partitioning techniques. The benefits for deploying Oracle database partitioning include the following:

- Transparently enhancing queries performance
- Purge of data through TRUNCATE partition SQL command
- Administration through Management Windows

The Enterprise Data Warehouse (EDW) repository applies database partitioning to large volatile tables that are queried based on interval, for example quarterly or yearly. Partitions on a partitioned table are either pre-created or created on demand, it depends on the partition type.

Figure 9 shows the partitioning techniques followed by a brief description for the non-composite types. As noticed the database partitioning was introduced with Oracle 8i release, however; it was Oracle Database 11g which innovated new partitioning methods to support wide-range of VLDB applications.

| Range                | List       | Hash       | Interval       | System | Reference | Virtual Column |
|----------------------|------------|------------|----------------|--------|-----------|----------------|
| 8i                   | 8i         | 8i         | 11g            | 11g    | 11g       | 11g            |
| Composite Partitions |            |            |                |        |           |                |
| Range-Range          | List-List  | Hash-Hash  | Interval-Range |        |           |                |
| Range-List           | List-Range | Hash-Range | Interval-List  |        |           |                |
| Range-Hash           | List-Hash  | Hash-Hash  | Interval-Hash  |        |           |                |

Figure 9: Partitioning Methods for Oracle Database



## Range Partition

RANGE partition map rows to partitions based on the partition key value. It's commonly implemented partitioning tables and indexes based on column of type DATE data type. For example, the sales transaction timestamp is used as the partition key. A default partition is created to store rows that does not map to any of the partitions list. The range partition names are either user-defined or system-generated.

## List Partition

LIST partition explicitly maps rows to partitions. The list of discrete values is defined for each partition, which in turn controlled by the current value of the partitioning key. For example, the sales transaction state address is used as the partition key. In the case the row is not mapped to of the defined partitions, an error is returned.

## Hash Partition

The hash partitioning uses hashing algorithm on the partitioning key to map rows to partition randomly, but maintaining balanced distribution across partitions. The hash partition is implemented when there is no obvious candidate for partitioning the table. Define the number the partitions by power of 2 to support even rows distribution.

## System Partition

With system partition there is no partition key column, instead the mapping of rows to pre-created partitions is controlled by the application. The application specifies the partition name as part of the DML statement.

## Interval Partition

Interval partition is an extension to range partition. A partition is created when the inserted/updated row is not mapped to a current partition. Oracle creates the partition automatically and diverts the rows to the new partitions. The new created partitions are given system-generated names.

## Reference Partition

Reference partition allows the partitioning of two related tables by parent-child referential active primary key and foreign key constraints. The partitioned key is inherited through the primary key, foreign key relationship.

## Virtual Column-Base Partition

All partitioning methods support virtual columns as being regular columns. The virtual column is based on an expression that will be used as the partition key. After the value is derived, then the partition mapping is determined.

## Composite Partitions

As the name suggest, composite partitions combines two of the above partitions to create a sub-partition in the form two-dimensional partition view. For example, RANGE-LIST composite partition creates a sub-partition; the RANGE uses TRANS\_DATE and the LIST uses TRANS\_SHIP\_STATE as the partition key columns. Figure 10 depicts the RANGE-LIST composite partition.

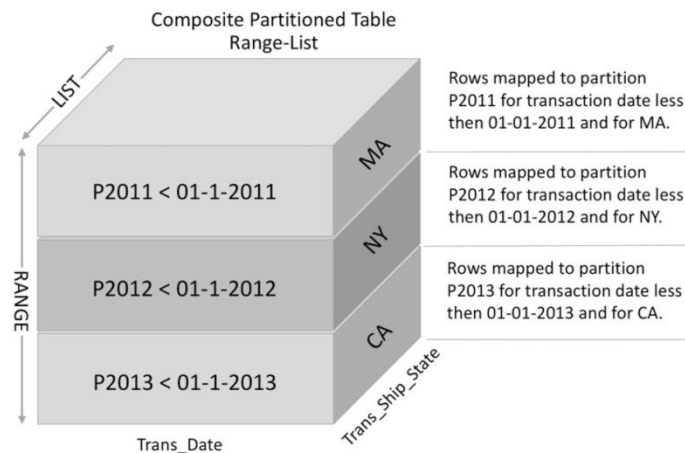


Figure 10: RANGE-LIST composite partition

## Online Table Redefinition

For database applications that must remain available to users while migrating tables from heap (normal) table to partitioned table, empower the supplied PL/SQL package DBMS\_REDEFINITION to migrate the table to the new structure, then drop the original table without the users noticing such disruptive operation.

To avoid index sanity issues with INVALID status, partition the index using the same partitioning key as its table. This is done by specifying the clause LOCAL when creating the index. This type of index is automatically maintained by the operations on the table. For example, dropping the table partition automatically drops the index partition, otherwise the index is marked as INVALID and need rebuild.

Figure 11 depicts the flow of executing the redefinition package to migrate a table from heap (normal) to partitioned table.

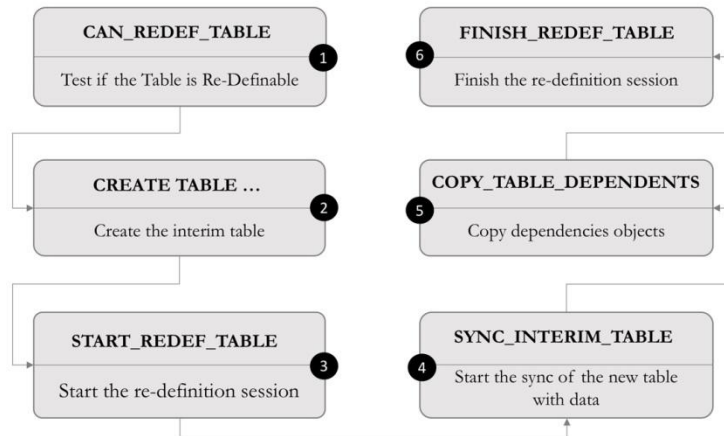


Figure 11: DBMS\_REDEFINITION execution steps

## Step 1: Test if the Table Re-definable

Prior to re-defining a table, validating the table is necessary. If the call to DBMS\_REDEFINITION.CAN\_REDEF\_TABLE is successful, then the table is re-definable. If an error is returned, then the table is not re-definable.

## Step 2: Create the Interim Table

This interim table reflects the new structure after successfully completing the calls to DBMS\_REDEFINITION session. Our use case is to transform an ordinary heap table to partitioned table.

## Step 3: Start the Re-Definition Session

The call to the procedure DBMS\_REDEFINITION.START\_REDEF\_TABLE initiates the redefinition using the interim table. It copy rows from the original table to the interim table and creates the materialized view log to record transactions occurring during the re-definition.

## Step 4: Start the Sync of the Interim Table

To keeps the interim table synchronized with the original table by moving the recorded transactions from the materialized view log to the interim table, which represents the delta recorded during the re-definition session.

## Step 5: Copy Dependencies Objects

This is an optional step; it attaches the dependent objects of the original table to the interim table. This includes indexes, privileges, constraints, triggers, etc.

## Step 6: Finish the Re-Definition Session

The call to `DBMS_REDEFINITION.FINISH_REDEF_TABLE` ends the re-definition session. This step swaps the two tables names, and enables disabled triggers. The interim table can be dropped now.

The ordinary heap table has been transformed to partitioned table without the application's users feeling such destructive operation. Verify the re-definition is successful by querying `USER_TABLES` to verify the table is partitioned.

```
SQL> select PARTITIONED
2    from user_tables
3   where table_name='CALL_DETAILS';
```

Refer to Chapter 8 to learn more about migration of tables from non-partitioned to partitioned table.

## Database In-Memory Option

The database In-Memory database option was introduced in Oracle Database 12c Release 1, as the name suggests it empowers faster access for data residing in the database System Global Area (SGA).

Figure 12 shows the data access techniques available for Oracle Database 12c.

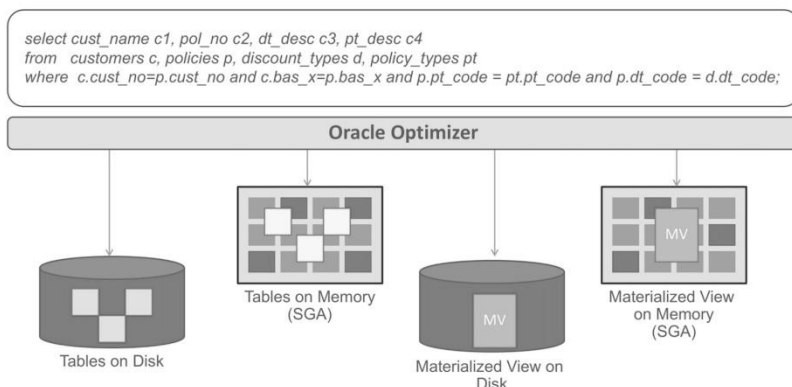


*Figure 12: Data Access Techniques for Oracle Database*

As shown in Figure 12, data access is available in the following forms:

- Tables on disk (data files)
- Tables in-memory
- Materialized views on disk (data files)
- Materialized views in-memory

Oracle Database in-memory setting is configured using database parameter `INMEMORY_SIZE`, which size the memory allocation for the in-memory objects. Figure 13 analyzes and compares the data access for a table in-memory and a table on disk.

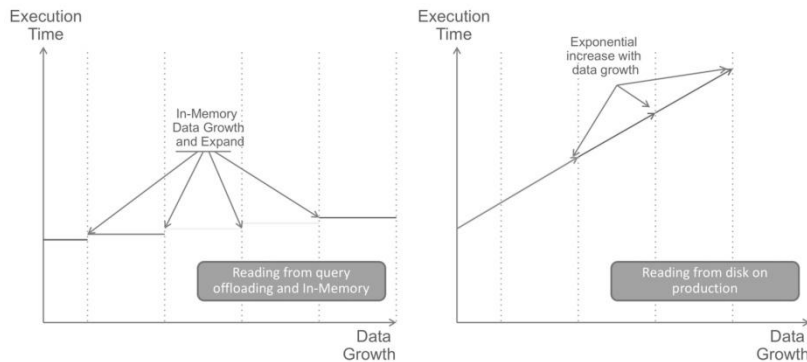


*Figure 13: Analysis of data access on disk and in-memory*

The graph on the left demonstrates data access for a materialized view residing in memory. The graph shows the query execution time remains constant for a refreshed in memory materialized view, when the in memory materialized view expands (re-refresh); then the execution time increases but remains constant for the current state of the materialized. This is a real-time query.

The graph on the right demonstrates data access for a table residing on disk. The graph shows the query execution time continues to linearly increase with data growth.

## Memory Management

The Enterprise Data Warehouse (EDW) is a hostile environment; hence the memory requirements for an EDW must be calculated precisely. The database administrator (DBA) keeps monitoring the SGA behavior and fine-tune the SGA parameters accordingly, pinning volatile objects (Tables, Materialized Views, Packages) in memory enhances the queries performance avoiding unnecessary disk access.

Figure 14 shows the System Global Area (SGA) memory distribution for a VirtualBox with an allocated memory of 120GB.

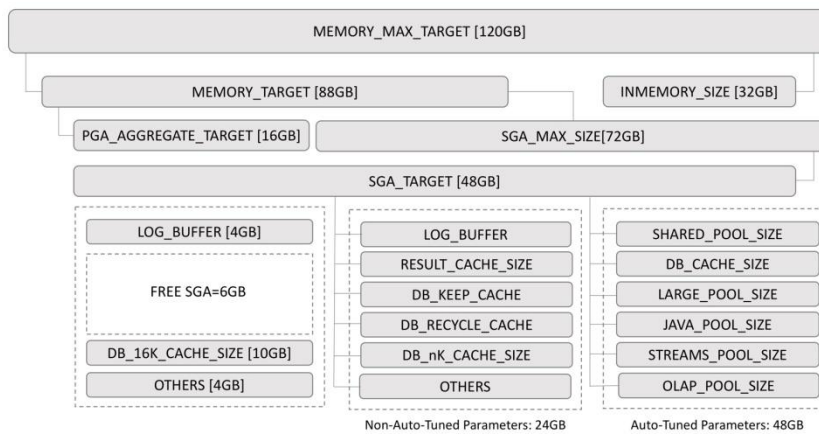


Figure 14: System Global Area Memory Allocation

Starting from the top-down, the parameter `MEMORY_MAX_TARGET` sets the ceiling for the database instance, the focus is to preserve and control the overall instance memory size, from which the database in-memory is allocated by setting the parameter `INMEMORY_SIZE`, and the `PGA_AGGREGATE_TARGET` parameter allocates the overall program global area (PGA). The second level is `SGA_MAX_SIZE`; it enables the `SGA_TARGET` parameter to control tunable and non-tunable parameters. The remaining of `SGA_MAX_SIZE` after allocating `SGA_TARGET` is allocated for the non-tunable parameters and others memory structures.

With the exception of `MEMORY_MAX_TARGET` parameter, all memory-related parameters are dynamically allocated. The following demonstrate the memory distribution shown in Figure 14.

Allocate enough shared memory from file system; login as root, execute the mount command below, and modify `/etc/fstab` for permanent settings after system reboot.

```
# mount -t tmpfs shmfs -o size=128G /dev/shm
```

Before you begin changing the database parameters, backup the parameters file, you may need it.

```
EDI> create pfile='/tmp/initEDI.ora' from spfile;
File created.
```

```

EDI> alter system set memory_max_target=120G scope=spfile;

System altered.

EDI> alter system set memory_target=88G scope=spfile;

System altered.

EDI> alter system set inmemory_size=32G scope=spfile;

System altered.

EDI> alter system set pga_aggregate_target=16G scope=spfile;

System altered.

EDI> alter system set sga_max_size=72G scope=spfile;

System altered.

EDI> alter system set sga_target=48G scope=spfile;

System altered.

EDI> alter system set db_16k_cache_size=10G scope=spfile;

System altered.

EDI> alter system set log_buffer=4G scope=spfile;

System altered.

EDI> startup force
ORACLE instance started.

Total System Global Area          7.7309E+10 bytes
Fixed Size                        12169416 bytes
Variable Size                     2.7917E+10 bytes
Database Buffers                  1.3690E+10 bytes
Redo Buffers                      1330003968 bytes
In-Memory Area                    3.4360E+10 bytes
Database mounted.
Database opened.
EDI>

```

Refer to Chapter 8 to learn more about configuring the database in-memory option.

## Database Advanced Compression

Data compression is a priority feature for data warehouse. It's designed to deliver two capabilities: reduce data storage (for disk and memory) and improve queries performance. Though its applications transparent, but not every database table is a perfect candidate for data compression, however; tables with high cardinality

(redundant data) benefit more from data compression. Storage utilization enhancement reaches 2-time or 4-time with advanced row compression.

Oracle Database 12c provides a compression advisor (DBMS\_COMPRESSION) to help you decide if the table is a candidate for compression. Depending on the application type, the specific compression is applied. Figure 15 depicts bulk load operation for data warehouse keep compressing data in a database block.

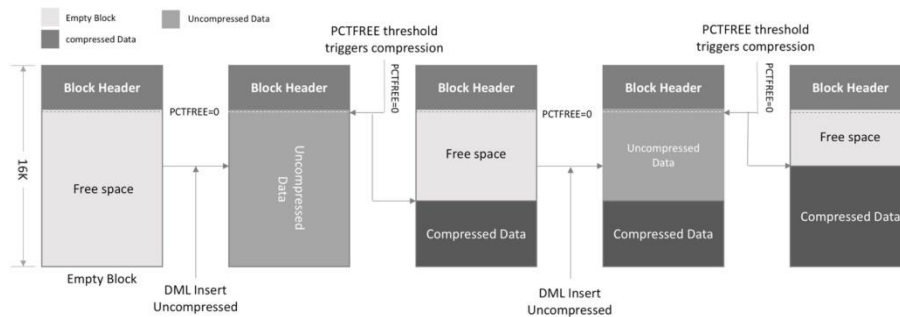


Figure 15: Oracle Advanced Compression Cycle

The algorithm, a new block is moved to the free list, it's an empty block with PCTFREE=0 for compress settings. The next DML inserts remain uncompressed until the PCTFREE is reached which trigger block data compression. The pattern of compression continues until the block is saturated and move out of the free list.

## Basic Table Compression

The table is compressed using the clause COMPRESS [BASIC]. Suitable for data warehouse applications performing bulk load using the command CREATE TABLE AS SELECT, for example. Specify the clause COMPRESS to enable compression when creating the table as shown below.

```
CREATE TABLE contacts
COMPRESS FOR QUERY
AS
SELECT * FROM contacts@cont;
```

## Advanced Row Compression

In reality, data repeated more in columns more than rows, which allows a high ratio of compression. Repeated identical column values are compressed by mean of *symbol* which represents a reference to the identical column values. The advanced row compression, duplicate values in the rows and columns in a data block are stored once



at the beginning of the block, duplicate values are replaced with a short reference (symbol). Thus, information needed to re-create the uncompressed data is stored in the block.

For activate OLTP environment specify STORE COMPRESS ADVANCED clause to enable advanced row compression when creating the table as shown below.

```
CREATE TABLE contacts
ROW STORE COMPRESS ADVANCED
AS
SELECT * FROM contacts@cont;
```

## Warehouse Compression

This is a Hybrid Columnar Compression (HCC); it's an alternative approach for storing data in a columnar format, where data is organized using row and columnar method. HCC uses a logical contract called Compression Unit (CU) which stores the compressed column data for a set of rows as shown in Figure X. A data set composed of 11 rows for a table with three columns, stored across to data blocks.

From Oracle EXADAT v2 creates a column vector for each column, compresses the column vectors, and stores the column vectors in data blocks. The collection of blocks is called a compression unit. The blocks in a compression unit contain all the columns for a set of rows, as shown in Figure 16. Warehouse compression algorithm has been optimized for query performance, specifically for scan-oriented queries used heavily in data warehouses. This approach is ideal for tables that will be queried frequently.

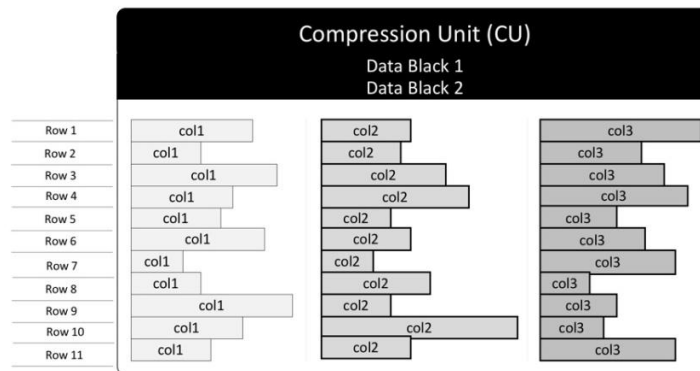


Figure 16: Hybrid Columnar Compression (HCC) Representation

HCC is available for EXADATA, Pillar Axiom, Sun ZFS Storage Appliance (ZFSSA) and FS1 storage.

```
CREATE TABLE contacts
COMPRESS FOR QUERY HIGH
AS
SELECT * FROM contacts@cont;
```

## Archive Compression

This is a Hybrid Columnar Compression (HCC); with archive compression, the compression algorithm has been optimized for maximum storage savings. This approach is ideal for tables that are infrequently accessed.

```
CREATE TABLE contacts
COMPRESS FOR ARCHIVE
AS
SELECT * FROM contacts@cont;
```

## Compression Advisor

Oracle Database 12c provides a compression advisor. The purpose of the advisor is to determine the compression ratio in order to help making the decision if the table is a candidate for compression. The following demonstrates the compression advisor for the table CUSTOMERS.

```
declare
    blkcnt_cmp    pls_integer;
    blkcnt_uncmp  pls_integer;
    rowcnt_cmp    pls_integer;
    rowcnt_uncmp  pls_integer;
    comp_ratio    number;
    comptype      varchar2(200);
begin
    DBMS_COMPRESSION.GET_COMPRESSION_RATIO(
        'USERS',          -- Tablespace name
        'OSM_REPAPI',     -- Schema
        'CUSTOMERS',      -- Table name
        NULL,             -- sub-object name
        4,                -- Compression for high queries
        blkcnt_cmp,        -- # of compressed blk
        blkcnt_uncmp,      -- # of uncompressed blk
        rowcnt_cmp,        -- # of rows compressed
        rowcnt_uncmp,      -- # of rows uncompressed
        comp_ratio,        -- Compression ratio
        comptype,          -- Compression type
        1000000,           -- Compression Min rows
        1);               -- Object type for tables
    DBMS_OUTPUT.PUT_LINE('Block count compressed = ' || blkcnt_cmp);
    DBMS_OUTPUT.PUT_LINE('Block count uncompressed = ' || blkcnt_uncmp);
    DBMS_OUTPUT.PUT_LINE('Row#/block compressed = ' || rowcnt_cmp);
    DBMS_OUTPUT.PUT_LINE('Row#/block uncompressed = ' || rowcnt_uncmp);
    DBMS_OUTPUT.PUT_LINE('Compression type = ' || comptype);
    DBMS_OUTPUT.PUT_LINE('Compression ratio = ' || comp_ratio || ' to 1');
end;
```

The output of the above compression advisor is below. It shows a major gap between compressed versus uncompressed blocks, and the gap for the number of compressed rows per block; hence, leading to an amazing compression ratio of 7.3 to 1.

```
Compression Advisor self-check validation successful. select count(*) on both
Uncompressed and EHCC Compressed format = 226071 rows
Block count compressed = 339
Block count uncompressed = 2481
Row#/block compressed = 667
Row#/block uncompressed = 91
Compression type = "Compress Query High"
Compression ratio = 7.3 to 1
```

## No Logging Option

When running DML statements, the transaction logs generated, stored on the online redo log files, and ultimately written to the archive log files; instance failure recovered by applying the online redo log file, and database failure recovered by applying both, the online redo log files and archive log files to achieve zero data loss. There three level to specify the logging option: Database, Tablespace and Table/Index.

For an Enterprise Data Warehouse (EDW) where the transactions of read-only type or bulk load, specifying NOLOGGING when creating the table avoid generating the transaction logs, hence enhancing the DML throughput and bulk load.

The creation of tables using sub-query benefits from NOLOGGING option, the table created with minimum overhead and faster.

```
create table loc_compqrylow
nologging
compress for query low
as
select * from loc_nocomp;
```

## Materialized Views

Simply, a materialize view stores the result of a query. The materialize view created based on the application's SQL for the purpose of enhancing the execution elapsed time for analytical applications running on the Enterprise Data Warehouse (EDW). Though the materialized view is likely to enhance the query performance, however; the storage requirements of the materialized view must be associated with the performance gain, and compared with the absence of materialized view.

Figure 17 shows the components involve to create a materialized view: The query, the base tables, the materialized view, and the refresh method.

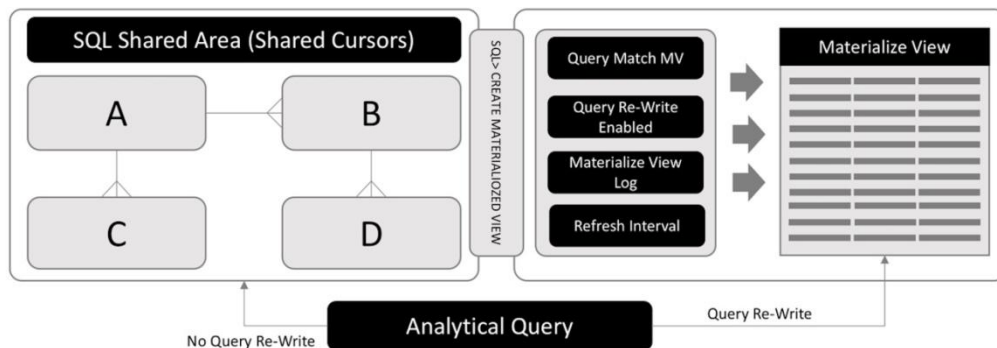


Figure 17: Materialize View Components

In order to create the materialized view, the application's SQL normally extracted from the SQL shared area. Then create the materialized view with the proper parameters to allow query re-write.

## Query Re-Write

The query re-write enables data access from the materialized view rather than from tables found in the FROM clause of the SQL statement. The optimizer evaluates the execution plan, and in the event a materialized view is available and allows faster execution plan, then the query is re-directed for data access from the materialized view. The query re-write happens if it's enabled, and the overall structure of the query and materialized view are matching.

## Materialized View Logs

In order to refresh the materialized view efficiently, a materialized view logs are created on the base tables. Figure 18 illustrates how materialized view log refreshes the materialized view. The materialized view log stores the change data capture since the last refresh, and then update the materialized view based on user-defined interval, for which one of the interval is on-commit. Because of query re-write is enabled, the optimizer opts to perform data access from the materialized view.

When the materialized view refreshed automatically using the ON COMMIT option, committed transactions updates the base tables on which a materialized view is defined, the changes are automatically propagated to the materialized view. However; when the materialized view refreshed on demand, use of the following options to refresh the materialized view:

- *Fast*, incremental refresh by applying changes to the materialized view.
- *Fast PCT*, Re-defining the rows in the materialized view affected by changed partitions in the base tables, then refreshes the materialized view.
- *Complete*, Refreshes by recalculating the defining query of the materialized view.
- *Force*, Attempts a fast refresh. If that is not possible, it does a complete refresh.

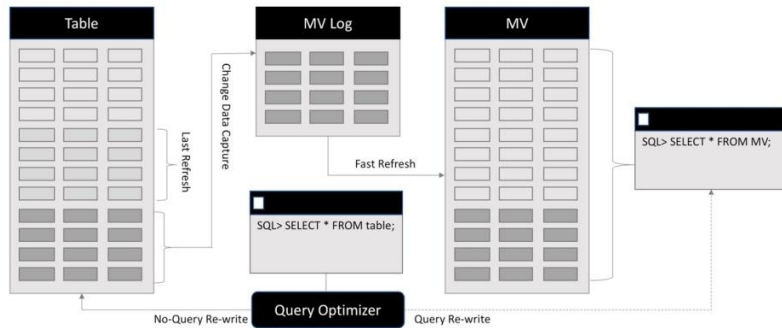


Figure 18: Materialized View Refresh Technique

## In-Memory Materialized View

Pushing the materialized view in-memory drastically enhances the queries performance by performing data access from memory. The aim is to deliver real-time analytical applications from the result of a complex query stored in-memory.

## Materialized View Compression

To further enhance the execution performance and minimized the memory requirements, the materialized view is created or altered as compressed, and in-memory.

## DBMS\_MVIEW

Materialized views are administered using the supplied PL/SQL API package DBMS\_MVIEW. Refreshing the materialized view and purge the materialized view logs are the most common usage of DBMS\_MVIEW.

## Parallel Processing

To expedite the execution of queries, the parallel execution is driven by one coordinator and several slaves processes. The coordinator manages the slaves and

ultimately merges the result. There are different methods to enable parallel query execution:

- Include the PARALLEL clause on the CREATE or ALTER table
- Add optimizer PARALLEL hint to queries and DML statements

The degree of parallelism (DOP) is specified as part of the PARALLEL clause. The database manages the degree of parallelism through the resource manager, which imposes an upper limit on the database based on the user's consumer group.

## Parallelism by Examples

The following CREATE table statement creates a table with a degree of parallelism (DOP) of 4.

```
SQL> create table customers(  
2  cust_id          number primary key,  
3  cust_name        varchar2(64) not null,  
4  cust_address     varchar2(128))  
5  parallel 4;
```

Table created.

SQL>

The following ALTER table statement changes the degree of parallelism to 8.

```
SQL> alter table customers parallel 8;
```

Table altered.

SQL>

The following ALTER table statement disables the degree of parallelism.

```
SQL> alter table customers noparallel;
```

Table altered.

SQL>

The following enable parallelism at the session level for queries and DMLs.

```
SQL> commit;
```

Commit complete.

```
SQL> alter session force parallel DML parallel 8;
```

SQL&gt;

The following spin off 8 sessions from within the query using an optimizer hit.

```
SQL> select /*+ parallel (customers, 8) */ cust_name
       2 from customers;
```

## Execution Plan

To gain an inside of the performance of the query, use Oracle SQL Developer, the EXPLAIN SQL command or DBMS\_XPLAN package to display the execution plan for the query. Figure 19 shows the execution plan using Oracle SQL Developer.

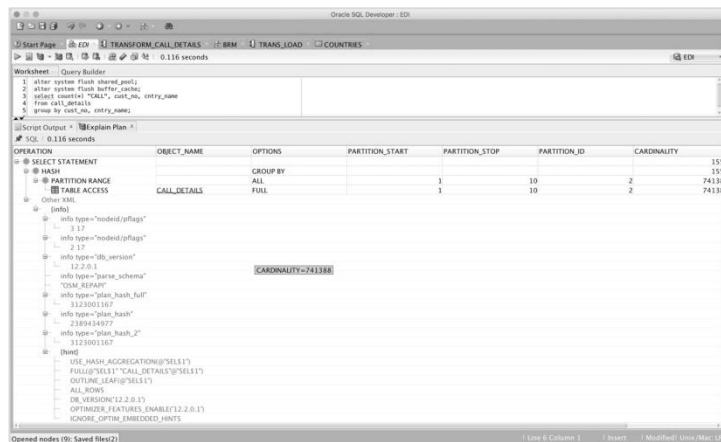


Figure 19: SQL Developer Execution Plan

## Database Parameters Settings for Data Warehouse

The Enterprise Data Warehouse (EDW) is predominantly read-only database. Oracle database 12c Enterprise Edition is shipped with OLAP and Analytics options. When deploying an Enterprise Data Warehouse database, data warehouse-specific parameters must be modified to suite the workloads of the analytical applications. Deriving accurate values for these parameters should have a positive impact on the response time, which in turn improves the overall performance of the applications.

The following describes the essential database parameters for EDW only. Query the data dictionary V\$SPPARAMETER to view the complete list of parameters.

## MEMORY\_MAX\_TARGET

It sets the upper limit of the Database Instance System Global Area (SGA) as the absolute size. It safeguards superseding the memory allocation for other instances running within the same server. In an EDW where the in-memory option configured, the value should be:

`MEMORY_MAX_TARGET = MEMORY_TARGET + INMEMORY_SIZE`

**Default Value:** 0

## MEMORY\_TARGET

This parameter sets the size of the maximum System Global Area (SGA) after allocating the memory size of the total Program Global Area (PGA).

**Default Value:** 0 to MEMORY\_MAX\_TARGET

## INMEMORY\_SIZE

The memory area allocated for the in-memory option. The value for INMEMORY\_SIZE should be calculated to accommodate the total size of the volatile (most frequently accessed) objects pinned in memory.

**Default Value:** 0

## DB\_FILE\_MULTIBLOCK\_READ\_COUNT

It specifies the maximum number of blocks read in one I/O operation during a sequential scan. The total number of I/Os needed to perform a full table scan depends on such factors as the size of the table, the multi-block read count, and whether parallel execution is being utilized for the operation.

**Default Value:** The default value corresponds to the maximum I/O size that can be efficiently performed and is platform-dependent



## **OLAP\_PAGE\_POOL\_SIZE**

Specify the size of the OLAP page pool. Since it's likely to run an EDW and an OLAP on the same database, explicit sizing of the OLAP memory is recommended.

**Default Value:** 0

## **STAR\_TRANSFORMATION\_ENABLED**

This parameter determines whether the database cost-based query transformation will be applied to star queries.

**Default Value:** FALSE

## **DB\_nK\_CACHE\_SIZE**

This parameter set the cache size for the independent memory allocated to the specific I/O block size. For the EDW, n should be set to 16 or 32. The I/O block size is specified when creating the tablespace, then tables, indexes and materialized views are migrated to the tablespace with 16 or 32 I/O block size.

**Default Value:** 0

## **QUERY\_REWRITE\_ENABLED**

This parameter enables query re-write when the database optimizer evaluates the execution plan; it may transform the query to execute against the materialized view instead of the query tables found in the FROM clause.

**Default Value:** TRUE

## **QUERY\_REWRITE\_INTEGRITY**

Determines the degree to which Oracle must enforce query rewriting. At the safest level, Oracle does not use query rewrite transformations that rely on unenforced relationships.

**Default Value:** TRUSTED

# Temporary Groups

Temporary space is not needed when the queries processing steps executed within the System Global Area (SGA), the database allocated memory. In the case of the Enterprise Data Warehouse (EDW) where the database operations are mostly executing analytical applications, the need for large enough temporary space is predominant. Hence configuring temporary groups instead of temporary tablespace avoids running out of extents when submitting long-running queries. Figure 20 depicts the structure of a temporary group.

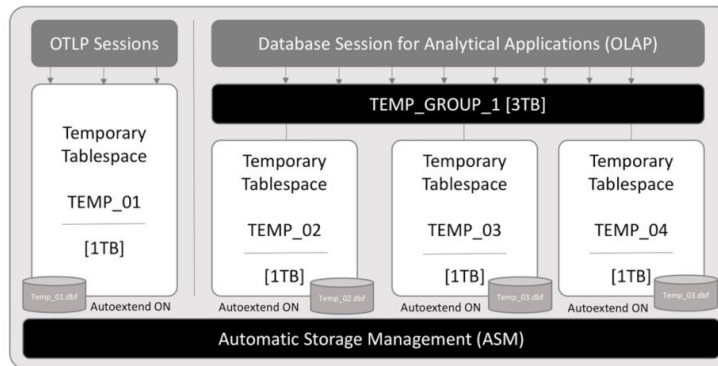


Figure 20: Temporary Area Design

The temporary groups assigned to database users as the default temporary tablespace. The following SQL commands create the temporary group, then altering the database user to use the temporary group as the default temporary area, with this setup in place; the users are unlikely to encounter temporary space related errors while running long-running, regardless of the query complexity.

The following illustrate Figure 19 configuration.

```
SQL> CREATE TEMPORARY TABLESPACE TEMP_02
2  TEMPFILE '/u01/app/oracle/oradata/DHDB/datafile/temp_02_01.dbf'
3  SIZE 1000000M
4  AUTOEXTEND ON
5  TABLESPACE GROUP TEMP_GROUP_1;
```

Tablespace created.

```
SQL> CREATE TEMPORARY TABLESPACE TEMP_03
2  TEMPFILE '/u01/app/oracle/oradata/DHDB/datafile/temp_03_01.dbf'
3  SIZE 1000000M
4  AUTOEXTEND ON
5  TABLESPACE GROUP TEMP_GROUP_1;
```

Tablespace created.

```
SQL> CREATE TEMPORARY TABLESPACE TEMP_04
2  TEMPFILE '/u01/app/oracle/oradata/DHDB/datafile/temp_04_01.dbf'
3  SIZE 1000000M
4  AUTOEXTEND ON
5  TABLESPACE GROUP TEMP_GROUP_1;
```

Tablespace created.

```
SQL> ALTER USER OSM_REPAPI TEMPORARY TABLESPACE TEMP_GROUP_1;
```

User altered.

```
SQL>
```

## Connection Management for Data Warehouse

In an Enterprise Data Warehouse, there are mainly two types of database session connections: End-User and Administrator connections. The end users running analytical applications should be connected to the database using shared connection, where administrators performing materialized view refresh, for example, should connect using dedicated connection. However; there are four types of connections:

- Database Default
- Dedicated Server
- Shared Server
- Pooled Server

Figure 21 shows NETMGR for configuring SQLNET Connections types.

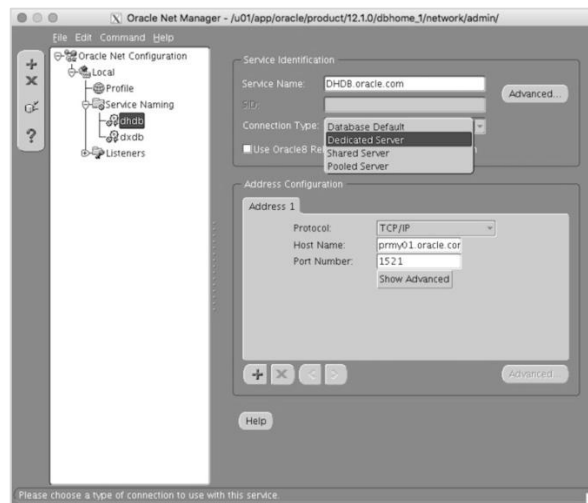


*Figure 21: NETMGR for configuring connections*

The primary idea of configuring different connection types is to control database system resources. As a general practice, configure the connection type to avoid idle database sessions. Depending on the activities of the users and the application processing transactions, the proper type of server connection is configured for the database, which is designated as Database Default.

## Shared Connections

System resources are always limited; efficient utilization of system resources through the database resource manager or setting database parameters must be developed and enforced. When the users attempt to establish a database session, the connection type is determined through the database parameter settings. Figure 22 shows shared connections to the database. The user session assigned to a dispatcher, the dispatcher place the request, a server process picks up the request, the server process the request, the processed request is placed on the response queue of the dispatcher, the response is sent to the dispatcher initiated the request and finally passed to the user session.

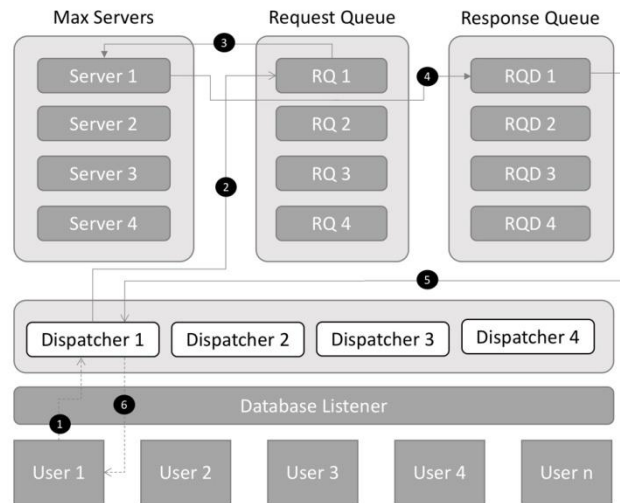


Figure 22: Shared Server Architecture

The Shared configuration requires specifying the following database parameters listed in Table 5. Direct on indirect default value is evaluated for these parameters.

| Parameter          | Description                                                                                                                                 |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| max_shared_servers | Specifies the maximum number of shared server processes allowed to be running simultaneously. Setting this parameter enables you to reserve |

|                        |                                                                                                                                                                                                                                       |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                        | process slots for other processes, such as dedicated servers.                                                                                                                                                                         |
| shared_servers         | The value of this parameter should be less than MAX_SHARED_SERVERS. If it is greater than or equal to MAX_SHARED_SERVERS, then the number of servers will not be self-tuned but will remain constant, as specified by SHARED_SERVERS. |
| shared_server_sessions | Specifies the total number of shared server sessions to allow. Setting this parameter enables you to reserve user sessions for dedicated servers.                                                                                     |

Table 5: Shared Connection Database Parameters

## Dedicated Connections

Database connections such as RMAN backup session, call-out to the PL/SQL API to refresh the materialized log should be done through dedicated connection only. When the database is configured for shared connection, then dedicated connection is enforced using DEDICATED parameter in the SQLNET configuration file as shown below.

```
DHDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = prmy01.oracle.com) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = DHDB.oracle.com)
    )
  )
```

The value for the SERVER parameter is DEDICATED, even when the database is configured for shared server, database connection using DHDB TNS alias is spawn as dedicated connection.

## Pooled Connections

For an online application running on an application server, where the number and size of database transactions is difficult to predict, pool connection is the best option for such type of transactions.

```
DHDB =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = prmy01.oracle.com) (PORT = 1521))
    )
  )
```

```

)
(CONNECT_DATA =
  (SERVER = POOLED)
  (SERVICE_NAME = DHDB.oracle.com)
)
)

```

The value for the SERVER parameter is POOLED, even when the database is configured for shared server, database connection using DHDB TNS alias is spawned as pooled connection. The connection-pooling enables the database server to time out an idle session and use the connection to service an active session. The idle logical session remains open, and the physical connection is automatically reestablished when the next request comes from that session. Pooled server is configured though shared server.

## Data Visualization

Enterprise Data Warehouse (EDW) representations using Oracle Data Visualization Desktop allows quick yet basic design and deployment of analytical applications and dashboards. Figure 23 shows a dashboard using Oracle Data Visualization Desktop, the product available for Mac OS and MS Windows.



Figure 23: Oracle Data Visualization Desktop Edition

For an Enterprise Data representation, consider using Oracle Business Intelligence technology products to design, build and deploy high-performance enterprise-wide analytical applications, including mobile applications for smart devices.

# Summary

Chapter 1 discussed the components to build an Enterprise Data Warehouse (EDW) from a concept prospective. The following chapters are deep-dive-in for deploying EDW using Oracle Database 12c options for EDW and Oracle GoldenGate 12c for high-performance non-intrusive change data capture.

There are two types of real-time change data capture (CDC): Classic and integrated. Classic capture is out of the scope of the book, the integrated capture types, functionalities and features discussed. The EDW benefits from the integrated extract (IE) deployed as downstream system. It delivers maximum security and performance.

Data transformation is essential part of an EDW. Oracle GoldenGate 12c is capable of delivering basic-to-complex data transformation. Columns mapping is part of the process parameter file, which allows flexible self-contained columns mapping and data transformation without impacting other processes. Oracle GoldenGate 12c built-in functions are key component for performing data transformations, using Oracle GoldenGate 12c functions is recommended real-time data manipulations.

Oracle GoldenGate 12c has different types of apply process – Replicat. The EDW high-performance load demands parallel apply processing. Using integrated-parallel apply delivers the most flexible, yet the highest performance for EDW, where the initial parallel processing is of degree 25.

Oracle Database 12c Release 2 options have been designed to address EDW challenges. The following are the database options to be applied to the EDW instance, without such database options the instance will not perform as expected.

- Database Large Block I/O
- Database Partitioning
- Database In-Memory
- Advanced Data Compression
- Materialized Views
- Parallelism

Applying an accurate hardware sizing is crucial for the Enterprise Data Warehouse (EDW). The database in-memory option requires additional physical memory to manage data in memory, the database materialized view and materialized view logs require additional disk space to store and keep them refreshed as configured.

The subsequent chapters implement the Enterprise Data Warehouse (EDW) from the ground up; where major part of it applying Oracle Database 12c options for EDW.

## Overview

Prior to the installation of Oracle GoldenGate 12c, a very essential pre-installation step must be completed, *certification*. Oracle GoldenGate 12c certification is validated against the operating system and the Oracle database, uncertified combinations lead to unsupported configuration. Whenever it's possible, installs the latest Oracle GoldenGate 12c release and applies all patches, this help avoiding the upgrade after deploying on production environment.

Figure 1 shows the major building blocks of software layers for the Enterprise Data Warehouse (EDW) running on Oracle Virtualized environment.

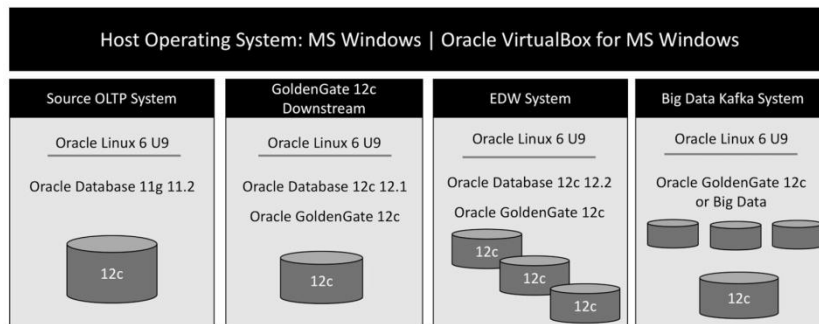


Figure 1: EDW Software Installations

At the time of writing this book, the current release of Oracle GoldenGate 12c is release 12.3 and Oracle Database 12c release 12.2. The following sections describe the basic details for each of the system.

Though the illustrations describe the installations on Oracle Linux, the same procedures should apply to UNIX operating systems, however; for MS Windows, conversion of the directories structure is essential and the installation of Microsoft Visual C++ distributable package runtime libraries is required.



## OLTP System

This is the source system, because Oracle GoldenGate 12c deployed using downstream technology, Oracle GoldenGate 12c is not installed on the source system. This is add-value security feature for enterprises that do not permit additional software on the production environment.

*Table 1 presents the OLTP system details.*

| Component        | Version                              |
|------------------|--------------------------------------|
| Operating System | Oracle Linux Release 6 Update 9      |
| Oracle Database  | Oracle Database 11g Release 11.2.0.4 |

*Table 1: Source System Details*

## Downstream System

This is Oracle GoldenGate 12c downstream system. It receives the redo-transport from the source system for database mining and change data capture. The data pump process sends the trails to the target (EDW) system.

*Table 2 presents the downstream system details.*

| Component         | Version                            |
|-------------------|------------------------------------|
| Operating System  | Oracle Linux Release 6 Update 9    |
| Oracle GoldenGate | Oracle GoldenGate 12c Release 12.3 |
| Oracle Database   | Oracle Database 12c Release 12.1   |

*Table 2: Downstream System Details*

## EDW System

This is the Enterprise Data Warehouse (EDW) system. In addition to performing high-performance apply; Oracle GoldenGate 12c executes complex data transformations functions.

*Table 3 presents the EDW system details.*

| Component         | Version                            |
|-------------------|------------------------------------|
| Operating System  | Oracle Linux Release 6 Update 9    |
| Oracle GoldenGate | Oracle GoldenGate 12c Release 12.3 |
| Oracle Database   | Oracle Database 12c Release 12.2   |

*Table 2: Downstream System Details*

# Big Data Kafka System

This is the big data system. The decision for selecting Kafka is to enable real-time data pipeline for streaming applications. Publishing, consuming and exporting messages from Kafka into an Oracle Database 12c allows spinning off real-time applications.

Table 4 presents the big data system details, running Kafka.

| Component                      | Version                                         |
|--------------------------------|-------------------------------------------------|
| Operating System               | Oracle Linux Release 6 Update 9                 |
| Oracle GoldenGate for big data | Oracle GoldenGate 12c for big data Release 12.3 |
| Oracle Database                | Oracle Database 12c Release 12.2                |
| Big Data                       | Kafka Version 1.x                               |

Table 4: Big Data Kafka System Details

## Oracle GoldenGate 12c Installation

The Oracle GoldenGate 12c installation conducted using silent installation method. The silent installation method uses a response file to automate the installation steps. Stage the installation media and prepare the response file.

### Response File

The response file for Oracle GoldenGate 12c installer composed of few lines in the form of a pair of attribute and value, as shown next.

```
1. oracle.install.responseFileVersion=rsfmt_ogginstall_response_schema_v12_3_0
2. _INSTALL_OPTION=ORA12c
3. SOFTWARE_LOCATION=/u01/app/oracle/product/12.3.0/gghome 1
4. START_MANAGER=true
5. MANAGER_PORT=7840
6. DATABASE_LOCATION=/u01/app/oracle/product/12.2.0/dbhome_1
7. INVENTORY_LOCATION=/u01/app/oraInventory
8. UNIX_GROUP_NAME=oinstall
```

**Line 1:** States the response file version, its Oracle GoldenGate version specific. The above indicates it's for Oracle GoldenGate version 12.3.

**Line 2:** Specify the database, the available options are: Oracle Database 11g or Oracle Database 12c. The above indicates its Oracle Database 12c.

**Line 3:** This is the directory location of Oracle GoldenGate 12c home.

**Line 4:** A Boolean attribute, TRUE value starts the instance manager upon successful installation.

**Line 5:** The instance manager run on port 7840.

**Line 6:** This is the directory location of Oracle Database 12c home.

**Line 7:** The Oracle GoldenGate 12c software inventory, this the metadata for tracking new installation and patches.

**Line 8:** This is Linux/UNIX specific attribute. It indicates the operating system group of the owner. This is mainly for security and management.

## Stage Oracle GoldenGate 12c Software Media

Follow the steps below to download, uncompress and stage the software.

- Download Oracle GoldenGate software from [edelivery.oracle.com](http://edelivery.oracle.com). Save the software media at /stage/ogg12c
- Navigate to the directory /stage/ogg12c, unzip the software media

```
$unzip 123010_fbo_ggs_Linux_x64_shiphome.zip
```

- While login as Oracle, create the directory specified by the attribute SOFTWARE\_LOCATION

```
$mkdir -p /u01/app/oracle/product/12.3.0/gghome 1
```

## Run the Installer

Navigate to the directory /stage/ogg12c/fbo\_ggs\_Linux\_x64\_shiphome/Disk1 and run the installer program – runInstaller.

```
$ runInstaller -silent -responsefile /home/oracle/oggcore1.rsp -nowelcome -showProgress -force -waitforcompletion
```

Starting Oracle Universal Installer...

|                                                   |                 |        |
|---------------------------------------------------|-----------------|--------|
| Checking Temp space: must be greater than 120 MB. | Actual 50219 MB | Passed |
| Checking swap space: must be greater than 150 MB. | Actual 9991 MB  | Passed |

```
Preparing to launch Oracle Universal Installer from /tmp/OraInstall2018-01-06_02-54-51PM. Please wait ...You can find the log of this install session at:/u01/app/oraInventory/logs/installActions2018-01-06_02-54-51PM.log
```

```
Prepare in progress.  
..... 9% Done.
```

Prepare successful.

```
Copy files in progress.  
..... 31% Done.  
..... 44% Done.  
..... 65% Done.  
..... 70% Done.  
..... 89% Done.  
.....
```

Copy files successful.

Link binaries in progress.

Link binaries successful.

Setup files in progress.

Setup files successful.

Setup Inventory in progress.

Setup Inventory successful.

Finish Setup successful.  
The installation of Oracle GoldenGate Core was successful.  
Please check '/u01/app/oraInventory/logs/silentInstall2018-01-06\_02-54-51PM.log' for more details.

Running Configuration Assistants in progress.

```
Configuring and starting manager... in progress.  
..... 100% Done.
```

Configuring and starting manager... successful.  
Successfully Setup Software.  
\$

## Display Oracle GoldenGate 12c Instance

A running Oracle GoldenGate 12c instance composed of the following:

- An Oracle GoldenGate 12c root directory, referred to by home directory
- A directory structure for storing configuration files and runtime details
- A manager process running on unused, unreserved TCP/IP port
- A set of extracts and Replicats processes
- A Dynamic a growing memory

Figure 2 depicts the components of an Oracle GoldenGate 12c instance.

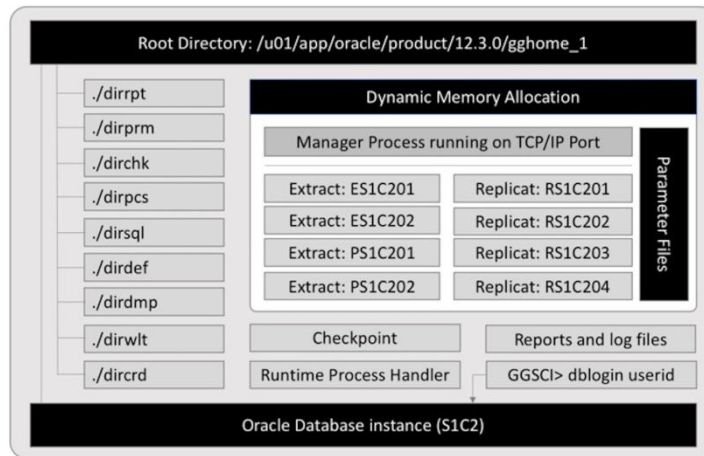


Figure 2: Oracle GoldenGate 12c Instance

Display the operating system runtime details for the manager process.

```
[oracle@linux4acdr gghome_1]$ ps -edf | grep MGR
oracle   23138      1  0 05:36 ?        00:00:00 ./mgr PARAMFILE
/u01/app/oracle/product/12.3.0/gghome_1/dirprm/mgr.prm REPORTFILE
/u01/app/oracle/product/12.3.0/gghome_1/dirrpt/MGR.rpt PROCESSID MGR
oracle   23144 12298  0 05:36 pts/2    00:00:00 grep MGR
[oracle@linux4acdr gghome_1]$
```

## The GoldenGate Software Command Interface

Start GGSCI and verify the manager process is running on port 7840.

```
[oracle@linux4acdr gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE 12.3.0.1.0 PLATFORMS 170721.0154 FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (linux4.oracle.com) 1> info mgr

Manager is running (IP port linux4.oracle.com.7840, Process ID 14030).

GGSCI (linux4.oracle.com) 2>
```

Verify the install directories are created. Note the Oracle GoldenGate 12c root directory.

```
GGSCI (linux4.oracle.com) 2> show

Parameter settings:

SET DEBUG          OFF

Current directory: /u01/app/oracle/product/12.3.0/gghome_1

Editor: vi

Reports (.rpt)           /u01/app/oracle/product/12.3.0/gghome_1/dirrpt
Parameters (.prm)        /u01/app/oracle/product/12.3.0/gghome_1/dirprm
Replicat Checkpoints (.cpr) /u01/app/oracle/product/12.3.0/gghome_1/dirchk
Extract Checkpoints (.cpe) /u01/app/oracle/product/12.3.0/gghome_1/dirchk
Process Status (.pcs)     /u01/app/oracle/product/12.3.0/gghome_1/dirpcs
SQL Scripts (.sql)        /u01/app/oracle/product/12.3.0/gghome_1/dirsq1
Database Definitions (.def) /u01/app/oracle/product/12.3.0/gghome_1/dirdef
Dump files (.dmp)         /u01/app/oracle/product/12.3.0/gghome_1/dirdmp
Masterkey wallet files (.wlt) /u01/app/oracle/product/12.3.0/gghome_1/dirwlt
Credential store files (.crd) /u01/app/oracle/product/12.3.0/gghome_1/dircrd

GGSCI (linux4.oracle.com) 3> exit
[oracle@linux4acdr gghome_1]$
```

The installer create set of directories for the GoldenGate 12c instance. Table 5 describes the directories.

| Directory | Description                                           |
|-----------|-------------------------------------------------------|
| Dirrpt    | Reports (logs)directory                               |
| Dirprm    | Stores the instance parameter files                   |
| Dirchk    | Stores the current checkpoint                         |
| Dirpcs    | Stores the operating system process id                |
| Dirsql    | For storing SQL scripts                               |
| Dirdef    | Stores the definition files for heterogeneous support |
| Dirdmp    | For storing dump files                                |
| Dirwlt    | Stores login wallet                                   |
| Dircrd    | Stores login credentials                              |

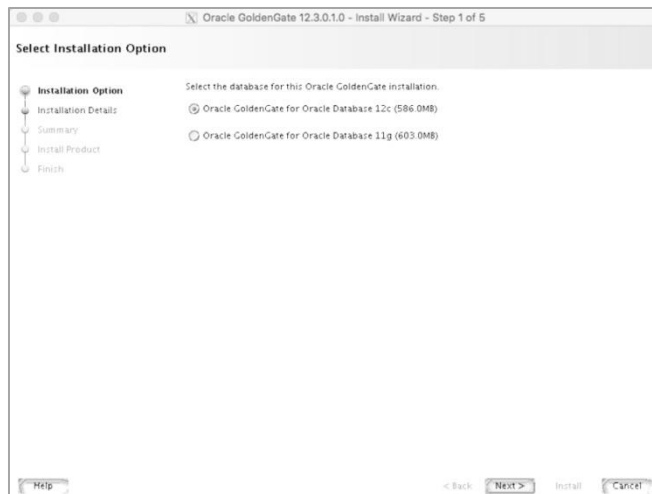
*Table 5: Oracle GoldenGate 12c Directories*

## Interactive Installer

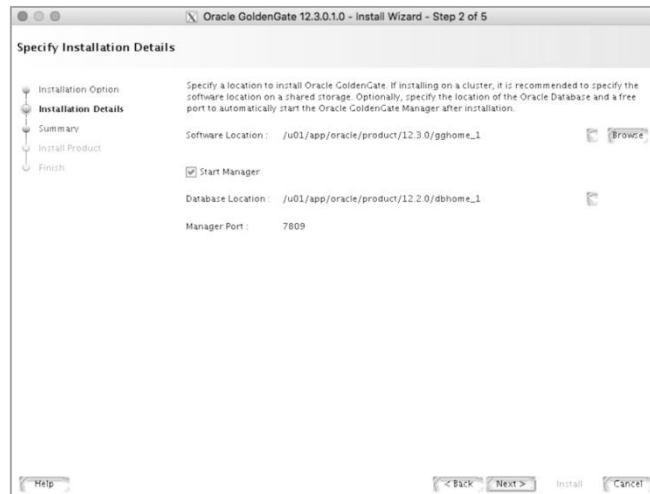
For user with access to the Linux console or X Window System, the following 5-step screenshots illustrates installing Oracle GoldenGate 12c from GUI environment. Navigate to the directory /stage/ogg12c/fbo\_ggs\_Linux\_x64\_shiphome/Disk1 and run the installer program – runInstaller.

```
$ runInstaller
```

### Step 1 of 5: Installation Options



## Step 2 of 5: Installation Details



## Step 3 of 5: Summary



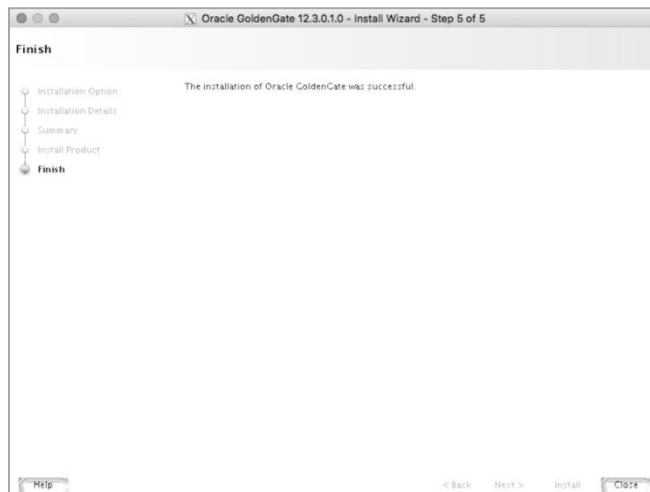
Verify the values entered by saving and viewing the response file, which may be used for future installations.



## Step 4 of 5: Install Product



## Step 5 of 5: Finish



For further details, refer to Appendix E.

## De-Installation of Oracle GoldenGate 12c

The de-installation performed from the command lines. Prior to the de-installation, stop all extracts, Replicats and manager processes. Then follow the steps below.

1. Stop all extracts, Replicats and manager processes
2. Navigate to `./deinstall` directory
3. Run the de-install program, `deinstall`
4. Verify the GoldenGate home, and enter `yes`

```
[oracle@linux4acdr ~]$ cd $OGG_HOME
[oracle@linux4acdr gghome_2]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE 12.3.0.1.0 PLATFORMS 170721.0154 FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (linux4acdr.oracle.com) 1> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING

GGSCI (linux4acdr.oracle.com) 2> stop mgr!

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

GGSCI (linux4acdr.oracle.com) 3> exit
[oracle@linux4acdr gghome_2]$ cd deinstall/
[oracle@linux4acdr deinstall]$ ./deinstall.sh

ALERT: Ensure all the processes running from the current Oracle Home are
shutdown prior to running this software uninstallation script.

Proceed with removing Oracle GoldenGate home:
/u01/app/oracle/product/12.3.0/gghome_2 (yes/no)? [no]
yes
Starting Oracle Universal Installer...

Checking swap space: must be greater than 500 MB.   Actual 9573 MB   Passed
Preparing to launch Oracle Universal Installer from /tmp/OraInstall2018-03-
31_04-45-43AM. Please wait ...Oracle Universal Installer, Version 12.2.0.1.4
Production
Copyright (C) 1999, 2016, Oracle. All rights reserved.

Starting deinstall

Deinstall in progress (Saturday, March 31, 2018 4:45:48 AM AST)
..... 100% Done.
```

```
Deinstall successful
```

```
End of install phases. (Saturday, March 31, 2018 4:46:09 AM AST)
```

```
End of deinstallations
```

```
Please check '/u01/app/oraInventory/logs/silentInstall2018-03-31_04-45-43AM.log' for more details.
```

```
[oracle@linux4acdr deinstall]$
```

# Summary

Oracle GoldenGate 12c installed on the following systems:

- Oracle GoldenGate 12c Downstream system
- The Enterprise Data Warehouse (EDW) System

The Oracle GoldenGate 12c installer supports two installation methods:

- Interactive GUI Installer
- Silent Installer

The interactive GUI installer is an option when the X Window system is available. Launch the installer and follow the 5-step wizard to install Oracle GoldenGate 12c. The interactive GUI installer allows selecting the Oracle Database release 11g or 12c.

The silent installer is an option when the X Window system is not available. The silent installer requires a response file as input. The response file composed of an attribute and value lines.

The installation creates the Oracle GoldenGate 12c instance sub-directories. Each sub-directory store Oracle GoldenGate 12c files related to parameter files (dirprm), trail files (dirdat), report log files (dirrpt), checkpoint point (dirchk), definition for heterogeneous support (dirdef), etc.

Upgrading Oracle GoldenGate 12c is likely to be a new installation. Follow the steps below to upgrade Oracle GoldenGate 12c, beginning with the source system.

1. Stop the extract processes on source
2. Ensure zero lag for the Replicat processes on target
3. Stop the Replicat processes on target and manager process on source
4. Copy the parameter files
5. Drop the checkpoint table (for target upgrade)
6. De-install Oracle GoldenGate instance on source
7. Delete the trail files
8. Install the new Oracle GoldenGate 12c under the old software locations
9. Restore the parameter files
10. Create the checkpoint table, (for target upgrade)
11. Define the extracts
12. Start the manager and extracts process

Gradually apply new features by modifying the configuration files and processes.

# Oracle Software Download Policy

The Oracle software download from <http://edelivery.oracle.com> is strictly subject to Oracle Software download terms and restrictions. You must read and accept the terms and conditions before download the software shown in Figure 3.

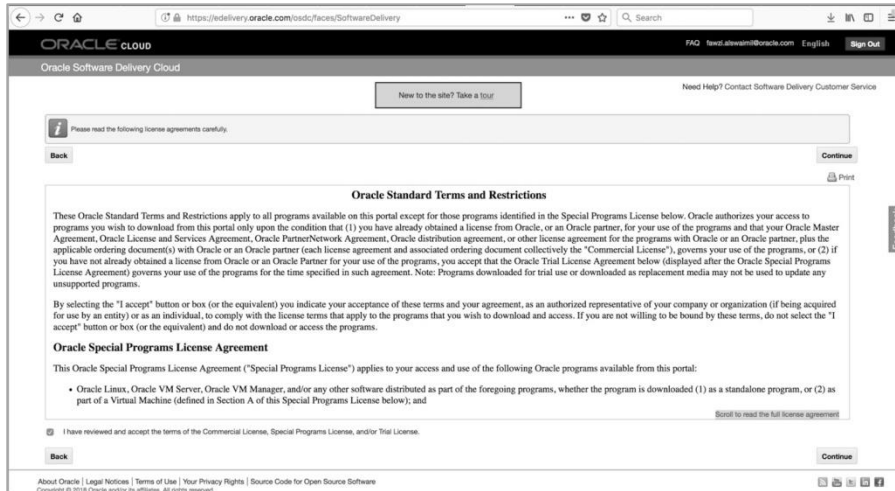


Figure 3: Oracle Software Download Standard Terms and Conditions

Refer to Appendix E for further details.

# Database Preparations

## Overview

This chapter covers the database pre-requisites to operate Oracle GoldenGate 12c. Oracle GoldenGate 12c performs complex inter-dependent processing phases to achieve optimized change data capture, yet it's non-intrusive and has minimum impact on the source system resources. The source database transactions logs, known as the online redo logs, is one of the major components for Oracle GoldenGate 12c, it's from where the data is captured. Hence, optimized configuration of the online redo log system must be addressed very carefully, in terms of redo log group size, number of groups, number of members within a group and contentions. The objective is to eliminate contentions and the log switch occurs at linear pattern.

Because Oracle GoldenGate 12c is transactional-oriented, applications design plays an important role for optimum change data capture. The applications running on the core systems must adhere to Oracle best practices, such as commit frequency is essential to avoid long running transactions as Oracle GoldenGate 12c tracks transaction in memory waiting for a commit to happen. Applications developers must understand the database isolation level when writing SQL applications to avoid wait events and row-level locking, which has a direct impact on how Oracle GoldenGate 12c handles checkpoints.

The following sections prepare the source, downstream and target database for the following:

- Enable archive operating mode
- Enable supplemental logging
- Setting the database parameters
- Create the tablespace
- Create the administration database user
- Grant database privileges to Oracle GoldenGate 12c administrator
- SQLNET configuration

## Enable the Database Archive Mode

The database online redo log groups track and record the database change vector. The following display the redo log groups, threads and its members.

```
SQL> select l.group#, l.thread#, lf.member
2   from   v$log l, v$logfile lf
3   where  l.group#=lf.group#
4   order by 1, 2
5   /
```

It's likely the database archive log mode already enabled. Query the database dictionary view V\$DATABASE to display the database archive mode.

```
[oracle@oltp01 ~]$ sqlplus /nolog

SQL*Plus: Release 11.2.0.3.0 Production on Sat Jan 27 08:02:55 2018

Copyright (c) 1982, 2011, Oracle. All rights reserved.

BRM> conn / as sysdba
Connected.
BRM> select log_mode
2   from   v$database;

LOG MODE
-----
ARCHIVELOG

BRM>
```

## Enable Database Supplemental Logging on Source

The database supplemental logging must be enabled. This is required by Oracle GoldenGate to perform database transactions logs capture. Query the current supplemental state and enable if necessary.

```
BRM> select supplemental_log_data_min
2   from v$database;

SUPPLEME
-----
NO

BRM> alter database add supplemental log data;

Database altered.

BRM> select supplemental_log_data_min
2   from v$database;
```

```
SUPPLEME
-----
YES
BRM>
```

Now, perform switch log file. This enables the database supplemental logging.

```
BRM> alter system switch logfile;

System altered.

BRM>
```

## Create GoldenGate Database Administrator on Source

Create Oracle GoldenGate, grant the privileges and test connection to the source database instance using the connect string OLTP. Because this is a test instance, the DBA privilege is granted. In a production systems, this must be avoided and explicitly grant the privileges required for the capture.

```
BRM> create tablespace ogg_dat
2 datafile '/u01/app/oracle/oradata/BRM/ogg_dat01.dbf'
3 size 50M autoextend on;

Tablespace created.

BRM> grant dba to ogg_admin identified by oracle;

Grant succeeded.

BRM> alter user ogg_admin default tablespace ogg_dat;

User altered.

BRM>
```

Verify create database session

```
BRM> conn ogg_admin/oracle@OLTP
Connected.
BRM>
```

In the event of unable to connect, please do not continue; the connection TNS error must be resolved before continuing further.



## Create GoldenGate Database Administrator on Downstream

Create Oracle GoldenGate schema, grant the privileges and test connection to the downstream database instance using the connect string (DSDB). Because this is a test instance, the DBA privilege is granted. In a production, this must be avoided and explicitly grant the privileges required for the capture.

```
DSDB>create tablespace ogg_dat;

Tablespace created.

DSDB>grant dba to ogg_admin identified by oracle;

Grant succeeded.

DSDB>alter user ogg_admin default tablespace ogg_dat;

User altered.

DSDB>
```

Test connection using DSDB connect string

```
DSDB>conn ogg_admin/oracle@DSDB
Connected.
DSDB>
```

## Database Parameters Setting on Downstream

Settings parameters required by Oracle GoldenGate Replication.

### ENABLE\_GOLDENGATE\_REPLICATION

This parameter is mandatory for Oracle Database 11g or higher, it enable the replication services.

### STREAMS\_POOL\_SIZE

This required for integrated extract to run the database mining server. The minimum recommended value is 750M, otherwise, use the automatic shared memory management (ASMM)

```
DSDB>alter system set enable_goldengate_replication=true scope=both;
```

```

System altered.

DSDB>alter system set streams_pool_size=5000M scope=both;

System altered.

DSDB>

```

Verify Database parameters settings

```

DSDB>show parameters enable_goldengate_replication

```

| NAME                          | TYPE    | VALUE |
|-------------------------------|---------|-------|
| enable goldengate replication | boolean | TRUE  |
| big integer                   | 5G      |       |

```

DSDB>show parameters stream_pool_size

```

| NAME             | TYPE        | VALUE |
|------------------|-------------|-------|
| stream pool size | big integer | 5G    |

```

DSDB> 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
[oracle@dss01 admin]$

```

## Verify GoldenGate Database Login for Downstream

Verify the connection by login to the database using Oracle GoldenGate 12c command line – GGSCI.

```

[oracle@dss01 admin]$ cd $OGG_HOME
[oracle@dss01 gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE 12.3.0.1.0 PLATFORMS 170721.0154 FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (dss01.mobily.com.sa) 2> miningdblogin userid ogg_admin@dsdb, password
oracle
Successfully logged into mining database.

GGSCI (dss01.mobily.com.sa) 3> dblogin userid ogg_admin@BRM, password oracle
Successfully logged into database.

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 4>

```

## Create GoldenGate Administrator on Target

Create Oracle GoldenGate schema, grant the privileges and test connection to the source database instance using the connect string (EDW). Because this a test instance, the DBA privilege is granted. In a production, this must be avoided and explicitly grant the privileges required for the capture.

```
[oracle@edw01 ~]$ sqlplus /nolog

SQL*Plus: Release 12.2.0.1.0 Production on Sat Jan 27 12:51:17 2018

Copyright (c) 1982, 2016, Oracle. All rights reserved.

EDT> conn / as sysdba
Connected.
EDT> create tablespace ogg_dat;

Tablespace created.

EDT> grant dba to ogg_admin identified by oracle;

Grant succeeded.

EDT> alter user ogg_admin default tablespace ogg_dat;

User altered.

EDT>
```

Verify connection is successful

```
EDT> conn ogg_admin/oracle@EDW
Connected.
EDT> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.2.0.1.0
- 64bit Production
[oracle@edw01 ~]$
```

## Verify GoldenGate Database Login for Target

Verify connection by login to the database using Oracle GoldenGate Software Command Interface (GGSCI).

```
[oracle@edw01 ~]$ cd $OGG_HOME
[oracle@edw01 gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE 12.3.0.1.0 PLATFORMS 170721.0154 FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.
```

```
Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (edw01.mobily.com.sa) 1> dblogin userid ogg_admin@EDI, password oracle
Successfully logged into database.

GGSCI (edw01.mobily.com.sa as ogg_admin@EDI) 2> exit
[oracle@edw01 gghome 1]$
```

## Database Parameters Setting on Target

ENABLE\_GOLDENGATE\_REPLICATION is the only required parameter to enable is the target required by Oracle GoldenGate 12c. However; there are additional database-related parameters to allocate the memory more accurately, these parameters are related to the System Global Area (SGA).

```
[oracle@edw01 gghome_1]$ sqlplus /nolog

SQL*Plus: Release 12.2.0.1.0 Production on Sat Jan 27 12:58:43 2018

Copyright (c) 1982, 2016, Oracle. All rights reserved.

EDI> conn / as sysdba
Connected.

EDI> alter system set enable_goldengate_replication=true scope=both;

System altered.

EDI>
```

## Enable GoldenGate Table-Level Supplemental Logging

Figure 1 shows the data model. Add table-level supplemental logging only for the tables involve in the change data capture.

Figure 1: Data Model

## List of Tables

List of the table

GGSCI>

To enable table-level supplemental logging by:

1. Start GGSCI
2. Login to the database
3. Run the command ADD TRANDATA

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 24> add trandata
osm_repapi.REGIONS

2018-02-03 18:20:21 INFO OGG-15132 Logging of supplemental redo data
enabled for table OSM_REPAPI.REGIONS.

2018-02-03 18:20:21 INFO OGG-15133 TRANDATA for scheduling columns has
been added on table OSM_REPAPI.REGIONS.

2018-02-03 18:20:21 INFO OGG-15135 TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.REGIONS.

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 25> add trandata
osm_repapi.PROFESSIONS

2018-02-03 18:20:32 INFO OGG-15132 Logging of supplemental redo data
enabled for table OSM_REPAPI.PROFESSIONS.

2018-02-03 18:20:32 INFO OGG-15133 TRANDATA for scheduling columns has
been added on table OSM_REPAPI.PROFESSIONS.

2018-02-03 18:20:32 INFO OGG-15135 TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.PROFESSIONS.

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 26> add trandata
osm_repapi.EDU_LEVELS
```

```

2018-02-03 18:20:43 INFO OGG-15132 Logging of supplemental redo data
enabled for table OSM_REPAPI.EDU_LEVELS.

2018-02-03 18:20:43 INFO OGG-15133 TRANDATA for scheduling columns has
been added on table OSM_REPAPI.EDU_LEVELS.

2018-02-03 18:20:43 INFO OGG-15135 TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.EDU_LEVELS.

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 27> add trandata
osm_repapi.NATIONALITIES

2018-02-03 18:20:54 INFO OGG-15132 Logging of supplemental redo data
enabled for table OSM_REPAPI.NATIONALITIES.

2018-02-03 18:20:54 INFO OGG-15133 TRANDATA for scheduling columns has
been added on table OSM_REPAPI.NATIONALITIES.

2018-02-03 18:20:54 INFO OGG-15135 TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.NATIONALITIES.

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 28> add trandata
osm_repapi.PACKAGES

2018-02-03 18:21:09 INFO OGG-15132 Logging of supplemental redo data
enabled for table OSM_REPAPI.PACKAGES.

2018-02-03 18:21:09 INFO OGG-15133 TRANDATA for scheduling columns has
been added on table OSM_REPAPI.PACKAGES.

2018-02-03 18:21:09 INFO OGG-15135 TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.PACKAGES.

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 29> add trandata
osm_repapi.PACKAGE_TYPES

2018-02-03 18:21:26 INFO OGG-15132 Logging of supplemental redo data
enabled for table OSM_REPAPI.PACKAGE_TYPES.

2018-02-03 18:21:26 INFO OGG-15133 TRANDATA for scheduling columns has
been added on table OSM_REPAPI.PACKAGE_TYPES.

2018-02-03 18:21:26 INFO OGG-15135 TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.PACKAGE_TYPES.

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 30> add trandata
osm_repapi.SIMS

2018-02-03 18:21:37 INFO OGG-15132 Logging of supplemental redo data
enabled for table OSM_REPAPI.SIMS.

2018-02-03 18:21:37 INFO OGG-15133 TRANDATA for scheduling columns has
been added on table OSM_REPAPI.SIMS.

2018-02-03 18:21:37 INFO OGG-15135 TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.SIMS.

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 31> add trandata
osm_repapi.CUSTOMERS

```

```

2018-02-03 18:21:53 INFO OGG-15132 Logging of supplemental redo data
enabled for table OSM_REPAPI.CUSTOMERS.

2018-02-03 18:21:53 INFO OGG-15133 TRANDATA for scheduling columns has
been added on table OSM_REPAPI.CUSTOMERS.

2018-02-03 18:21:53 INFO OGG-15135 TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.CUSTOMERS.

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 32> add trandata
osm_repapi.MOBILE_LINES

2018-02-03 18:22:15 INFO OGG-15132 Logging of supplemental redo data
enabled for table OSM_REPAPI.MOBILE_LINES.

2018-02-03 18:22:15 INFO OGG-15133 TRANDATA for scheduling columns has
been added on table OSM_REPAPI.MOBILE_LINES.

2018-02-03 18:22:15 INFO OGG-15135 TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.MOBILE_LINES.

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 33> add trandata
osm_repapi.CONTACTS

2018-02-03 18:22:26 INFO OGG-15132 Logging of supplemental redo data
enabled for table OSM_REPAPI.CONTACTS.

2018-02-03 18:22:26 INFO OGG-15133 TRANDATA for scheduling columns has
been added on table OSM_REPAPI.CONTACTS.

2018-02-03 18:22:26 INFO OGG-15135 TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.CONTACTS.

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 34> add trandata
osm_repapi.calls

2018-02-03 18:22:36 INFO OGG-15132 Logging of supplemental redo data
enabled for table OSM_REPAPI.CALLS.

2018-02-03 18:22:36 INFO OGG-15133 TRANDATA for scheduling columns has
been added on table OSM_REPAPI.CALLS.

2018-02-03 18:22:36 INFO OGG-15135 TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.CALLS.

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 35> add trandata
osm_repapi.call_details

2018-02-03 18:22:41 INFO OGG-15132 Logging of supplemental redo data
enabled for table OSM_REPAPI.CALL_DETAILS.

2018-02-03 18:22:41 INFO OGG-15133 TRANDATA for scheduling columns has
been added on table OSM_REPAPI.CALL_DETAILS.

2018-02-03 18:22:41 INFO OGG-15135 TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.CALL_DETAILS.

```

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 36> exit
```

## SQLNET Configuration

Configure SQLNET files (TNSNAMES.ORA), and then test the remote database connection.

### Configuring TNSNAMES.ORA File

Unless TNS\_ADMIN environment variable is set, the default SQLNET files are located on \$ORACLE\_HOME/network/admin.

```
DSDB =
```

```
OLTP =
```

```
XX
```

## Testing Remote Connections

Create a remote database connection from source to downstream

Create a remote database connection from downstream to source

.



# Summary

The database preparation is for meeting Oracle GoldenGate 12c mandatory pre-requisites. Depending on Oracle GoldenGate 12c operating mode, the pre-requisites may slightly vary.

- Enable archive operating mode
- Enable supplemental logging
- Setting the database parameters
- Create the tablespace
- Create the administration database user
- Grant database privileges to Oracle GoldenGate 12c administrator
- SQLNET files configuration and testing

This Enterprise Data Warehouse (EDW) implementation is deployed using Oracle GoldenGate 12c downstream technology; hence integrated extract (IE) must be registered with the database mining server. The downstream system receives the transactions logs; the database mining server produces the Logical Change Records (LCRs), which the integrated extract uses for real-time change data capture (CDC).

The source database archive log mode and the minimum supplemental logging is enabled, this allows the extracts processes perform change data capture (CDC) from the LCRs for integrated capture, and online redo logs for the classic capture.

Setting the database parameters is essentials, the two parameters are:

- `ENABLE_GOLDENGATE_REPLICATION`
- `STREAMS_POOL_SIZE`

`ENABLE_GOLDENGATE_REPLICATION` is a Boolean type and set to `TRUE`, the minimum value for the `STREAMS_POOL_SIZE` is 750M, unless Automatic Shared Memory Management (ASMM) is implemented.

Database security best practice strongly recommends creating a dedicated GoldenGate 12c database administrator with its default tablespace. Grant the minimum database privileges on source and target database. For integrated extract access to the `V$DATABASE` dynamic dictionary view is also required.

For integrated extract, configure SQLNET configuration files (`tnsnames.ora`) for shipping the transactions logs from the source to the downstream system, and from the downstream to the source for remotely enabling table-level supplemental logging.

# Configure Redo-Transport Service

## Overview

When Oracle GoldenGate 12c integrated extract (IE) deployed using downstream system, the source database must continuously ship the database transactions logs to the downstream by configuring the database redo-transport service. This is an online configuration; hence stopping the database services is not required and the source database sessions continue to operate normally.

Figure 1 shows the components of the redo-transport service.

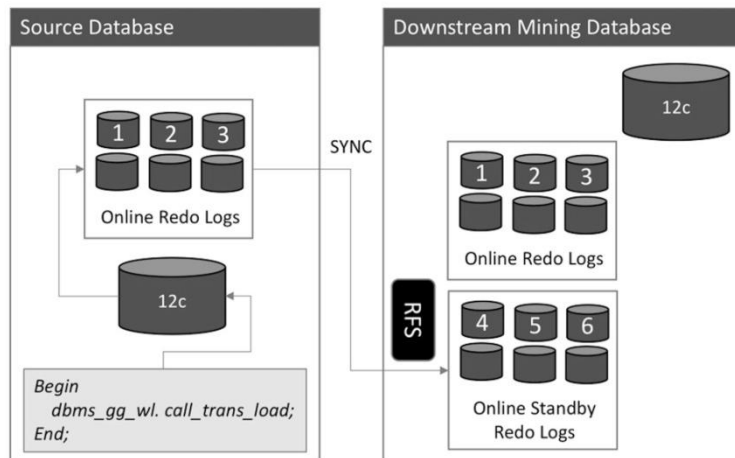


Figure 1: Redo-Transport Service

The remote file server (RFS) receives the transactions logs from the LNS process of the source, and write to the current online standby redo log file. The data is not applied to the downstream database, instead its prepared for the database mining server to produce the Logical Change Records (LCRs).

# Re-Create the Password File

The redo-transport requires the database password is identical for the source and downstream system.

Login to the primary node and downstream system as `_oracle` user and remove the current password files. Notice the password file is operating system specific, for Linux/UNIX the password file format is: `orapw<SID>`.

Use the Oracle password file utility `orapwd` to re-create the password file on the primary node only. *Do not re-create the password on the downstream system.* The utility uses the following options:

`IGNORECASE` : Non-Case sensitive password  
`ENTRIES` : The maximum number of SYSDBA entries to be granted  
`FORCE` : Override if password file exist

```
[oracle@oltp01 ~]$ cd $ORACLE_HOME/dbs
[oracle@oltp01 dbs]$ ls -al *pw*
-rw-r----- 1 oracle oinstall 1536 Jan 26 13:57 orapwBRM
[oracle@oltp01 dbs]$ orapwd file=orapwBRM password=oracle ignorecase=Y
entries=30 force=Y
[oracle@oltp01 dbs]$ ls -al *pw*
-rw-r----- 1 oracle oinstall 5120 Jan 28 07:11 orapwBRM
[oracle@oltp01 dbs]$ pwd
/u01/app/oracle/product/11.2.0/dbhome_1/dbs
[oracle@oltp01 dbs]$ scp orapwBRM
root@dss01:/u01/app/oracle/product/12.1.0/dbhome_1/dbs/orapwDSDB
root@dss01's password:
orapwBRM                                100% 5120      5.0KB/s   00:00
[oracle@oltp01 dbs]$
```

```
[oracle@dss01 ~]$ cd $ORACLE_HOME/dbs
[oracle@dss01 dbs]$ ls -al *pw*
-rw-r----- 1 oracle oinstall 5120 Jan 28 07:20 orapwDSDB
[oracle@dss01 dbs]$ su
Password:
[root@dss01 dbs]# chown oracle:oinstall orapwDSDB

[root@dss01 dbs]# exit
exit
[oracle@dss01 dbs]$ ls -al *pw*
-rw-r----- 1 oracle oinstall 5120 Jan 28 07:20 orapwDSDB
[oracle@dss01 dbs]$
```

Restart the primary node and downstream system database. The reason because the `IGNORECASE` was specified when re-creating the password file. `STARTUP FORCE` is used to quickly shutdown-restart the database instance.

```

[oracle@oltp01 dbs]$ sqlplus /nolog

SQL*Plus: Release 11.2.0.3.0 Production on Sun Jan 28 07:21:07 2018

Copyright (c) 1982, 2011, Oracle. All rights reserved.

BRM> conn / as sysdba
Connected.
BRM> startup force
ORACLE instance started.

Total System Global Area          1.2226E+10 bytes
Fixed Size                        2239616 bytes
Variable Size                     1811940224 bytes
Database Buffers                  1.0402E+10 bytes
Redo Buffers                      10031104 bytes
Database mounted.
Database opened.
BRM> exit

```

Restart the downstream database

```

[oracle@dss01 dbs]$ sqlplus /nolog

SQL*Plus: Release 12.2.0.1.0 Production on Sun Jan 28 07:25:54 2018

Copyright (c) 1982, 2016, Oracle. All rights reserved.

DSDB>conn / as sysdba
Connected.
DSDB>startup force
ORACLE instance started.

Total System Global Area          3.9863E+10 bytes
Fixed Size                        12169848 bytes
Variable Size                     3.3554E+10 bytes
Database Buffers                  6174015488 bytes
Redo Buffers                      122044416 bytes
Database mounted.
Database opened.
DSDB>

```

Verify TNS is reachable from the primary node to downstream system. *Please do not continue if the command tnsping result failed.*

```

[oracle@oltp01 dbs]$ tnsping DSDB

TNS Ping Utility for Linux: Version 11.2.0.3.0 - Production on 28-JAN-2018
07:24:36

Copyright (c) 1997, 2011, Oracle. All rights reserved.

Used parameter files:

```

```

Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP) (HOST = dss01.mobily.com.sa) (PORT = 1521))) (CONNECT_DATA =
(SERVICE_NAME = DSDB.mobily.com.sa)))
OK (0 msec)
[oracle@oltp01 dbs]$

```

## Configure the Source for Redo-Transport Service

The source database continuously ships the transactions logs to the database specified by the parameter SERVICE. This is an online database setting and does not require database re-start.

### Enable Redo-Transport Database Parameters

Configure LOG\_ARCHIVE\_DEST\_2 to transmit the Redo log files to the downstream system mining database. Table 1 describes the LOG\_ARCHIVE\_DEST\_n parameters.

| Parameter      | Description                                               |
|----------------|-----------------------------------------------------------|
| SERVICE        | The downstream database service                           |
| SYNC           | Ship the transactions logs synchronously                  |
| NOREGISTER     |                                                           |
| VALID_FOR      | The combination of the database role for this destination |
| REOPEN         |                                                           |
| DB_UNIQUE_NAME | The database unique name                                  |
| COMPRESS       | Compress the transport shipping                           |

*Table 1: LOG\_ARCHIVE\_DEST\_n Parameters*

```

[oracle@oltp01 dbs]$ sqlplus /nolog

SQL*Plus: Release 11.2.0.3.0 Production on Sat Jan 27 09:50:30 2018

Copyright (c) 1982, 2011, Oracle. All rights reserved.

BRM> conn / as sysdba
Connected.
BRM> alter system set log_archive_dest_2='SERVICE=DSDB SYNC NOREGISTER
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) REOPEN=10 DB_UNIQUE_NAME=DSDB'
scope=both;

System altered.

BRM>

```

LOG\_ARCHIVE\_DEST\_2 must be explicitly enabled

```
BRM> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2=ENABLE scope=both;

System altered.

BRM>
```

Set the data guard configuration, using the database unique names. Specifying the unique name for the primary node and the extended node 1.

```
BRM> ALTER SYSTEM SET
  2 LOG_ARCHIVE_CONFIG='DG_CONFIG=(BRM, DSDB) ' scope=both;

System altered.

BRM>
```

## Configure the Downstream for Redo-Transport

Review the size and number of members of the primary node, in preparation to create the standby Redo log files on the downstream system database.

```
BRM> select group#, bytes/1024/1024 MB from v$log
  2 order by 1;

   GROUP#      MB
-----
      1         200
      2         200
      3         200
      4         200
      5         200
      6         200

6 rows selected.

BRM> exit
```

Configure the standby redo log groups. The number of the standby Redo log groups, should be one greater than the primary node, i.e. 7.

## Create the Standby Database Directories

T

```
SQL> select file_name from dba_data_files
2 where rownum < 2;

FILE_NAME
-----
/u01/app/oracle/oradata/DSDB/datafile/o1_mf_users_f6w9cmd9_.dbf

SQL>
```

Use the output of the above query to create the directory locations for the Redo logs, the archived logs and the standby Redo logs for the downstream system mining instance.

```
SQL> host mkdir -p /u01/app/oracle/oradata/DSDB/arch

SQL> host mkdir -p /u01/app/oracle/oradata/DSDB/redo

SQL> host mkdir -p /u01/app/oracle/oradata/DSDB/standby

SQL>
```

## Enable Redo-Transport Database Parameters

Configure the downstream system mining database to receive and write shipped redo data locally, and to create the standby redo log files.

The location of the online archive log files

```
SQL> alter system set
2 log_archive_dest_1='LOCATION=/u01/app/oracle/oradata/DSDB/arch
3 VALID_FOR=(ONLINE_LOGFILE,PRIMARY_ROLE)' scope=both;

System altered.

SQL>
```

The location of the standby archive log files

```
SQL> alter system set
2 log_archive_dest_2='LOCATION=/u01/app/oracle/oradata/DSDB/standby
3 VALID_FOR=(STANDBY_LOGFILE,ALL_ROLES)' scope=both;

System altered.

SQL>
```

Enroll the source and downstream databases on data guard configuration

```
SQL> ALTER SYSTEM SET
```

```
2 LOG_ARCHIVE_CONFIG='DG_CONFIG=(BRM, DSDB)' scope=both;

System altered.

SQL>
```

Enable both destinations.

```
SQL> alter system set log_archive_dest_state_1='ENABLE' scope=both;

System altered.

SQL> alter system set log_archive_dest_state_2='ENABLE' scope=both;

System altered.

SQL>
```

## Create the Standby Redo Log Files

Show the location of the current online Redo groups (members)

```
DSDB> select member from v$logfile;

MEMBER
-----
/u01/app/oracle/oradata/DSDB/online/redo03a.log
/u01/app/oracle/fast_recovery_area/DSDB/online/redo03b.log
/u01/app/oracle/oradata/DSDB/online/redo02a.log
/u01/app/oracle/fast_recovery_area/DSDB/online/redo02b.log
/u01/app/oracle/oradata/DSDB/online/redo01a.log
/u01/app/oracle/fast_recovery_area/DSDB/online/redo01b.log
/u01/app/oracle/oradata/DSDB/online/redo04a.log
/u01/app/oracle/fast_recovery_area/DSDB/online/redo04b.log
/u01/app/oracle/oradata/DSDB/online/redo05a.log
/u01/app/oracle/fast_recovery_area/DSDB/online/redo05b.log
/u01/app/oracle/oradata/DSDB/online/redo06a.log

MEMBER
-----
/u01/app/oracle/fast_recovery_area/DSDB/online/redo06b.log

12 rows selected.

DSDB>
```

Create four standby Redo log groups: 4, 5, 6, 7, 8, 9, 10, 11, 12 and 13. The size should be greater than the current online Redo log groups of the primary node. Notice, two members of each group are defined.

```
DSDB> alter database add standby logfile group 7
```



```

2 ('/u01/app/oracle/oradata/DSDB/onlineolog/sredo07a.log',
'/u01/app/oracle/fast_recovery_area/DSDB/onlineolog/sredo07b.log') size 210M;

Database altered.

DSDB> alter database add standby logfile group 8
2 ('/u01/app/oracle/oradata/DSDB/onlineolog/sredo08a.log',
'/u01/app/oracle/fast_recovery_area/DSDB/onlineolog/sredo08b.log') size 210M;

Database altered.

DSDB> alter database add standby logfile group 9
2 ('/u01/app/oracle/oradata/DSDB/onlineolog/sredo09a.log',
'/u01/app/oracle/fast_recovery_area/DSDB/onlineolog/sredo09b.log') size 210M;

Database altered.

DSDB> alter database add standby logfile group 10
2 ('/u01/app/oracle/oradata/DSDB/onlineolog/sredo10a.log',
'/u01/app/oracle/fast_recovery_area/DSDB/onlineolog/sredo10b.log') size 210M;

Database altered.

DSDB> alter database add standby logfile group 11
2 ('/u01/app/oracle/oradata/DSDB/onlineolog/sredo11a.log',
'/u01/app/oracle/fast_recovery_area/DSDB/onlineolog/sredo11b.log') size 210M;

Database altered.

DSDB> alter database add standby logfile group 12
2 ('/u01/app/oracle/oradata/DSDB/onlineolog/sredo12a.log',
'/u01/app/oracle/fast_recovery_area/DSDB/onlineolog/sredo12b.log') size 210M;

Database altered.

DSDB> alter database add standby logfile group 13
2 ('/u01/app/oracle/oradata/DSDB/onlineolog/sredo13a.log',
'/u01/app/oracle/fast_recovery_area/DSDB/onlineolog/sredo13b.log') size 210M;

Database altered.

DSDB> exit

```

## Verify Redo-Transport

Verify redo-transport from the primary node to the downstream is successful.

```

[[oracle@oltp01 ~]]$ cd $ORACLE_DIAG
[oracle@oltp01 trace]$ pwd
/u01/app/oracle/diag/rdbms/brm/BRM/trace
[oracle@oltp01 trace]$ tail -f alert_BRM.log
Sun Jan 28 22:44:53 2018
Archived Log entry 20 added for thread 1 sequence 25 ID 0xea385b4b dest 1:
Sun Jan 28 22:45:03 2018
*****
LGWR: Setting 'active' archival for destination LOG_ARCHIVE_DEST_2

```

```

*****
Thread 1 advanced to log sequence 27 (LGWR switch)
  Current log# 3 seq# 27 mem# 0: /u01/app/oracle/oradata/BRM/redo03.log
Sun Jan 28 22:45:03 2018
Archived Log entry 23 added for thread 1 sequence 26 ID 0xea385b4b dest 1:

```

```

[oracle@oltp01 trace]$ sqlplus /nolog

SQL*Plus: Release 11.2.0.3.0 Production on Sat Feb 3 18:47:36 2018

Copyright (c) 1982, 2011, Oracle. All rights reserved.

BRM> conn / as sysdba
Connected.

BRM> alter system switch logfile;

System altered.

BRM> alter system switch logfile;

System altered.

BRM>

```

```

[oracle@oltp01 trace]$ tail -f alert_BRM.log
Sun Jan 28 22:44:53 2018
Archived Log entry 20 added for thread 1 sequence 25 ID 0xea385b4b dest 1:
Sun Jan 28 22:45:03 2018
*****
LGWR: Setting 'active' archival for destination LOG_ARCHIVE_DEST_2
*****
Thread 1 advanced to log sequence 27 (LGWR switch)
  Current log# 3 seq# 27 mem# 0: /u01/app/oracle/oradata/BRM/redo03.log
Sun Jan 28 22:45:03 2018
Archived Log entry 23 added for thread 1 sequence 26 ID 0xea385b4b dest 1:
Sun Jan 28 23:00:44 2018
*****
LGWR: Setting 'active' archival for destination LOG_ARCHIVE_DEST_2
*****
Thread 1 advanced to log sequence 28 (LGWR switch)
  Current log# 1 seq# 28 mem# 0: /u01/app/oracle/oradata/BRM/redo01.log
Sun Jan 28 23:00:44 2018
Archived Log entry 25 added for thread 1 sequence 27 ID 0xea385b4b dest 1:

```

# Summary

When Oracle GoldenGate 12c deployed using downstream technology, the redo-transport service configuration is the major one step. The source database redo-transport service ships the transactions logs to Oracle GoldenGate 12c downstream system for change data capture. The steps to configure the redo-transport service:

- Re-create the password file
- Setting redo-transport service parameters on source
- Setting redo-transport service parameters on downstream
- Creating standby online redo log files
- Configure database connect strings

The password for connecting AS SYSDBA must be identical on source and downstream system. The Oracle database utility ORAPWD creates the password file on the source, which is copied to the downstream system.

The redo-transport service uses one of the log archive destinations, assign the destination, and specify the downstream database connect string to ship the transactions logs using SYNC option for achieving real-time change data capture.

In the downstream the database mining server is running, it generates the Logical Change Records (LCRs), the integrated extract (IE) performs change data capture. The archive log locations for the online and standby redo log files are specified and added to the configuration list.

The last step is to create the online standby redo log files on the downstream system only. The standby redo log files store the transactions, in preparation to create the logical change records (LCRs), no transactions are applied to the downstream mining database.

One of the components configured to allow redo-transport service is the database TNS connect string. There will be two connects string, from source to downstream (DSDB) for shipping the transactions logs files, and from downstream to source (OLAP) used by the GGSCI to enable table-level supplemental logging.

Before proceeding further, the redo-transport service must show successful test, which can be done by simply performing a log switch, and monitor the alert file on the source and the opening of new archive file on the downstream.

The next chapter deploys Oracle GoldenGate 12c integrated extract (IE) for change data capture (CDC).

# Configure Real-Time Integrated Capture

## Overview

Applying the suitable change data capture (CDC) technology is the highest priority to achieve high-performance real-time data capture and apply. Oracle GoldenGate 12c is capable to support the following change data capture:

- Classic capture
- Locally deployed integrated capture
- Downstream integrated capture
- Archive Only Log (AOL) capture
- Active Data Guard capture

For an enterprise security and performance overhead are uncompromised, change data capture software are not permitted to run on the production environment, hence the deployment of downstream system is the adapted solution.

In order to deliver real-time Enterprise Data Warehouse (EDW), there are two change data capture options:

- Downstream integrated capture
- Active Data Guard capture

In the event of an existing Active Data Guard remote site with real-time apply, then it's a comfort zone to implement real-time EDW, otherwise Oracle GoldenGate 12c downstream system is the industry leading solution for scalable, high-performance EDW repository. The downstream system empowers a dedicated mining server to produce the logical change records (LCRs) for real-time change data capture.

This chapter explains and configures the use of Oracle GoldenGate 12c integrated extract (IE) deployed on downstream server; Figure 1 shows the architecture of Oracle GoldenGate 12c deployed on a downstream system.

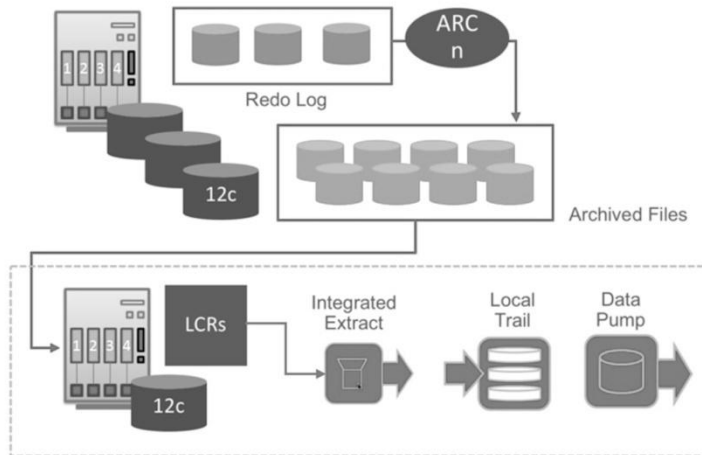


*Figure 1: Oracle GoldenGate 12 Downstream Topology*

The source database ships the redo-transport to Oracle GoldenGate 12c downstream system. The downstream run the database mining server; which generates the logical change records (LCRs). The LCRs goes though phase processing: Reader ► Preparer ► Builder ► Analyzer ► Coordinator ► LCRs.

Next, the following component configured and defined:

- The primary integrated extract (IE)
- The extract's trail
- The secondary extract (data pump)
- The remote trail

The primary integrated extract (IE) reads the LCRs, according to the configuration parameters TABLE; it performs change data capture on the list of tables, and writes the extract's trail (local trail). The secondary extract (data pump) reads the local trail and sends data to the remote trail. The data pump may also reduce the list of tables by excluding tables explicitly. The primary extract and the secondary extract may apply wildcard syntax when all tables to be subject for change data capture.

As part of Oracle GoldenGate 12c data transformation capabilities, this may occur either at the primary extract or secondary extract. However; keeping with best practices for the Enterprise Data Warehouse (EDW), data transformation should be shifted to a dedicated intermediate system or to the target EDW Oracle GoldenGate 12c instance. The Replicat running in the EDW instance can invoke stored database PL/SQL procedures, execute SQL statements and indirectly invoke database stored PL/SQL functions and Java classes running as JVM.

## Register the Integrated Extract

The first step, is registering the integrated extract with the database. It requires login to the downstream system and the source database to register the integrated extract.

```
[oracle@dss01 ~]$ cd $OGG_HOME
[oracle@dss01 gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE 12.3.0.1.0 PLATFORMS 170721.0154 FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (dss01.mobily.com.sa) 1> miningdblogin userid ogg_admin@dsdb, password
oracle
Successfully logged into mining database.

GGSCI (dss01.mobily.com.sa) 2> dblogin userid ogg_admin@BRM, password oracle
Successfully logged into database.

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 4> register extract EBRM01,
DATABASE

2018-01-27 20:07:37 WARNING OGG-02064 Oracle compatibility version
11.2.0.0.0 has limited datatype support for integrated capture. Version
11.2.0.3 required for full support.

2018-01-27 20:07:51 INFO OGG-02003 Extract EBRM01 successfully
registered with database at SCN 1191381.

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 5> exit
```

## Verify Successful Registration

Verify it's successful by querying DBA\_CAPTURE view. The integrated extract registration returns the System Change Number, which equate to the begin timestamp of the change data capture.

```
[oracle@dss01 gghome_1]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sat Feb 3 18:59:10 2018

Copyright (c) 1982, 2014, Oracle. All rights reserved.

DSDB> conn / as sysdba
Connected.
DSDB> select capture_name, captured_scn
2 from dba_capture;
```

| CAPTURE_NAME    | CAPTURED_SCN |
|-----------------|--------------|
| OGG\$CAP_EBRM01 | 1191381      |

DSDB> **exit**

## Configure the Integrated Extract

Prepare the configuration parameter for the primary extract. The TRANLOGOPTIONS to login to downstream mining database, and specify the change data capture options.

```
[oracle@dss01 gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE_12.3.0.1.0_PLATFORMS_170721.0154_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (dss01.mobily.com.sa) 1>
```

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 5> edit params EBRM01
```

```
EXTRACT EBRM01
SETENV (ORACLE_SID = DSDB)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
SETENV (TNS_ADMIN =
"/u01/app/oracle/product/12.1.0/dbhome_1/network/admin")
USERID ogg_admin@BRM, PASSWORD oracle
tranlogoptions mininguser ogg_admin@DSDB, miningpassword oracle
tranlogoptions integratedparams (max_sga_size 128, downstream_real_time_mine
Y)
EXTTRAIL ./dirdat/sa
TABLE osm_repapi.REGIONS;
TABLE osm_repapi.PROFESSIONS;
TABLE osm_repapi.EDU_LEVELS;
TABLE osm_repapi.NATIONALITIES;
TABLE osm_repapi.PACKAGES;
TABLE osm_repapi.PACKAGE_TYPES;
TABLE osm_repapi.SIMS;
TABLE osm_repapi.CUSTOMERS;
TABLE osm_repapi.MOBILE_LINES;
TABLE osm_repapi.CONTACTS;
TABLE osm_repapi.CALLS;
TABLE osm_repapi.CALL_DETAILS;
```

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 6> view params EBRM01
```

```
EXTRACT EBRM01
SETENV (ORACLE_SID = DSDB)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
SETENV (TNS_ADMIN =
"/u01/app/oracle/product/12.1.0/dbhome_1/network/admin")
USERID ogg_admin@BRM, PASSWORD oracle
tranlogoptions mininguser ogg admin@DSDB, miningpassword oracle
tranlogoptions integratedparams (max_sga_size 128, downstream_real_time_mine
Y)
TRANLOGOPTIONS EXCLUDEUSER ogg_admin
EXTTRAIL ./dirdat/sa
TABLE osm_repapi.REGIONS;
TABLE osm_repapi.PROFESSIONS;
TABLE osm_repapi.EDU_LEVELS;
TABLE osm_repapi.NATIONALITIES;
TABLE osm_repapi.PACKAGES;
TABLE osm_repapi.PACKAGE TYPES;
TABLE osm_repapi.SIMS;
TABLE osm_repapi.CUSTOMERS;
TABLE osm_repapi.MOBILE_LINES;
TABLE osm_repapi.CONTACTS;
TABLE osm_repapi.CALLS;
TABLE osm_repapi.CALL DETAILS;
```

```
GGSCI (dss01.mobily.com.sa) 3>
```

## Define the Integrated Extract

Run the command to define the Extract: ADD EXTRACT and ADD EXTTRAIL.

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 7> add extract EBRM01,
integrated tranlog, begin now
EXTRACT (Integrated) added.
```

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 8> add exttrail ./dirdat/sa,
extract EBRM01, megabytes 100
EXTTRAIL added.
```

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 9>
```

## Configure the Secondary Extract (data pump PBRM01)

Prepare the configuration parameter for the secondary extract (data pump)

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 9> edit params PBRM01
```

```
EXTRACT PBRM01
```



```

SETENV (ORACLE_SID = DSDB)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@DSDB, PASSWORD oracle
RMTHOST edw01.mobily.com.sa, MGRPORT 7850
RMTTRAIL ./dirdat/ta
TABLE osm_repapi.*;

```

```

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 10> view params PBRM01

```

```

EXTRACT PBRM01
SETENV (ORACLE_SID = DSDB)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@DSDB, PASSWORD oracle
RMTHOST edw01.mobily.com.sa, MGRPORT 7850
RMTTRAIL ./dirdat/ta
TABLE osm_repapi.*;

```

```

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 11>

```

## Define the Secondary Extract (data pump)

Define the secondary extract (data pump) by running the commands ADD EXTRACT and ADD RMTTRAIL.

```

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 11> add extract PBRM01,
exttrailsource ./dirdat/sa
EXTRACT added.

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 13> add rmttrail ./dirdat/ta,
extract PBRM01, megabytes 100
RMTTRAIL added.

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 14>

```

## Verify Extracts Status

Verify status of the extracts

```

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 14> info all

```

| Program | Status  | Group  | Lag at Chkpt | Time Since Chkpt |
|---------|---------|--------|--------------|------------------|
| MANAGER | RUNNING |        |              |                  |
| EXTRACT | STOPPED | EBRM01 | 00:00:00     | 00:11:42         |
| EXTRACT | STOPPED | PBRM01 | 00:00:00     | 00:02:01         |

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 15>
```

## INFO extract EBRM01

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 15> info extract EBRM01
```

```
EXTRACT    EBRM01      Initialized    2018-01-27 20:19    Status STOPPED
Checkpoint Lag      00:00:00 (updated 00:12:12 ago)
Log Read Checkpoint Oracle Integrated Redo Logs
                  2018-01-27 20:19:58
                  SCN 0.0 (0)
```

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 16>
```

## INFO extract PBRM01

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 16> info extract PBRM01
```

```
EXTRACT    PBRM01      Initialized    2018-01-27 20:29    Status STOPPED
Checkpoint Lag      00:00:00 (updated 00:03:04 ago)
Log Read Checkpoint File ./dirdat/sa0000000000
                  First Record  RBA 0
```

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 17>
```

## START all

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 17> start *
```

```
Sending START request to MANAGER ...
EXTRACT EBRM01 starting
```

```
Sending START request to MANAGER ...
EXTRACT PBRM01 starting
```

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 18> info all
```

| Program | Status  | Group  | Lag at Chkpt | Time Since Chkpt |
|---------|---------|--------|--------------|------------------|
| MANAGER | RUNNING |        |              |                  |
| EXTRACT | RUNNING | EBRM01 | 00:00:00     | 00:13:22         |
| EXTRACT | RUNNING | PBRM01 | 00:00:00     | 00:03:40         |

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 19> stop *
```

```
Sending STOP request to EXTRACT EBRM01 ...
Request processed.
```

```
Sending STOP request to EXTRACT PBRM01 ...  
Request processed.
```

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 20> info all
```

| Program | Status  | Group  | Lag at Chkpt | Time Since Chkpt |
|---------|---------|--------|--------------|------------------|
| MANAGER | RUNNING |        |              |                  |
| EXTRACT | STOPPED | EBRM01 | 00:03:22     | 00:00:06         |
| EXTRACT | STOPPED | PBRM01 | 00:00:00     | 00:00:06         |

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 21>
```

## Summary

The chapter applied the industry leading practice for change data capture (CDC). Oracle GoldenGate 12c downstream technology highly effective method for deploying non-intrusive, low-impact change data capture and data streaming. There are two Oracle GoldenGate 12c CDC options to deliver real-time Enterprise Data Warehouse (EDW).

- Option 1: Oracle GoldenGate 12c running on downstream system
- Option 2: Oracle GoldenGate 12c running on active data guard system

The first option empowers the source redo-transport service, the transaction logs shipped in SYNC mode to the downstream server for change data capture processing. It's only Oracle-to-Oracle database. Adapt classic change data capture for heterogeneous environment.

The second option uses an existing Oracle active data guard instance running with real-time apply. Oracle GoldenGate 12c process the online standby redo log files to perform change data capture. With low network latency, this option deliver the same result found with option 1.

Because Oracle GoldenGate 12c is not installed on the source database system, use Oracle GoldenGate 12c GGSCI remote connection to generate table-level supplemental logging through ADD TRANDATA command, while the database minimum supplemental logging is enabled using SQL ALTER command. The degree of overhead associated with supplemental logging depends on the transactions volume running on the source database.

Depending on the transactions volume of the source, the downstream system resources configured accordingly. The database mining server uses the stream pool size memory area; hence ensure the setting is capable to handle the transactions processing. The minimum recommended setting is 750M for the downstream database parameter – STREAMS\_POOL\_SIZE. Other memory related parameters, its recommended to be governed by Oracle Database 12c Automatic Shared Memory Management (ASMM).

After registering the integrated extract (IE), verify the system change number of the extract is accurate, because this is the minimum timestamp of the change data capture. While connected AS SYSDBA on the downstream database, query DBA\_CAPTURE data dictionary view for details of the integrated extract.

The next chapter deploys the delivery group for high-performance Replicat.

# Configure Real-Time High-Performance Apply

## Overview

This is a critical phase of the Enterprise Data Warehouse (EDW), the apply process must scale according to the transactions workloads of the source systems. The apply processes uses parallel processing techniques to enable real-time processing for data streaming from multiple sources.

Table 1 shows the different Replicat deployable with Oracle GoldenGate 12c release 12.3. EDW greatly benefits from Real-Application Cluster (RAC) for cross node processing feature, where a single load processing is handled by multiple instances.

|                                    | Replicat |             |            |                         |                     |
|------------------------------------|----------|-------------|------------|-------------------------|---------------------|
|                                    | Classic  | Coordinated | Integrated | Parallel Non-Integrated | Parallel Integrated |
| Batch Processing (BATCSQL)         | ✓        | ✓           | ✓          | ✓                       | ✓                   |
| Barrier Transactions               | ✗        | ✓           | ✓          | ✓                       | ✓                   |
| Dependency Computation             | ✗        | ✗           | ✓          | ✓                       | ✓                   |
| Auto-Parallelism                   | ✗        | ✗           | ✓          | ✓                       | ✓                   |
| DML Handler                        | ✗        | ✗           | ✓          | ✗                       | ✓                   |
| Auto-CDR                           | ✗        | ✗           | ✓          | ✗                       | ✓                   |
| Procedural Replication             | ✗        | ✗           | ✓          | ✗                       | ✓                   |
| Dependency aware Transaction Split | ✗        | ✗           | ✗          | ✓                       | ✓                   |
| Cross RAC node processing          | ✗        | ✓           | ✗          | ✓                       | ✓                   |

*Table 1: Oracle GoldenGate 12c Replicat Types and features*

The parallel-integrated Replicat is the most comprehensive; some of the features are applicable to Oracle GoldenGate 12c release 12.3, such as DML handler, Auto-CDR and procedural replication. The Auto-CDR is certified with Oracle Database 12c release 12.2 only. Data transformation may occur on a dedicated intermediate system as shown in Figure 1 instead of on the Replicat. The Replicat configuration invoke PL/SQL stored procedures, which invokes stored PL/SQL and Java functions for performing highly complex data transformation.

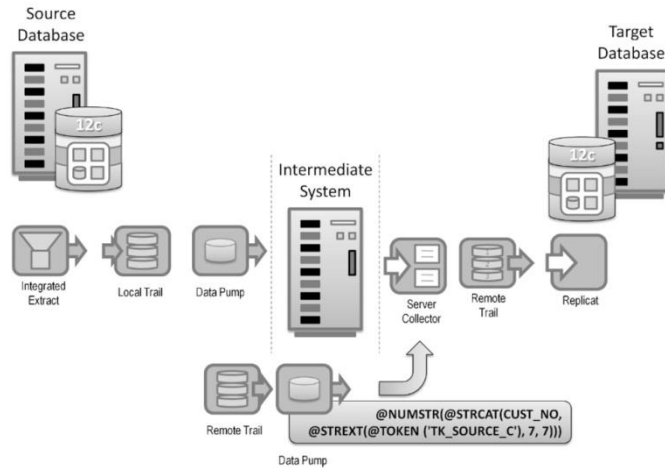


Figure 1: Intermediate System for Data Transformation

In such case, the intermediate system requires an operational database to perform data transformation using Oracle GoldenGate 12c functions, invoking stored PL/SQL procedures, etc.

## Configure the Database Checkpoint table

Create the checkpoint table as it's the default checkpoint for the Replicat.

```
[oracle@edw01 ~]$ cd $OGG_HOME
[oracle@edw01 gghome_1]$ pwd
/u01/app/oracle/product/12.3.0/gghome_1
[oracle@edw01 gghome_1]$ vi GLOBALS
```

```
checkpointtable ogg_admin.chkpttab
```

```
[oracle@edw01 gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE 12.3.0.1.0 PLATFORMS 170721.0154 FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (edw01.mobily.com.sa) 1> dblogin userid ogg_admin@EDI, password oracle
Successfully logged into database.

GGSCI (edw01.mobily.com.sa as ogg_admin@EDI) 2> add checkpointtable
```

```
No checkpoint table specified. Using GLOBALS specification  
ogg_admin.chkpttab...
```

```
Successfully created checkpoint table ogg admin.chkpttab.
```

```
GGSCI (edw01.mobily.com.sa as ogg_admin@EDI) 3>
```

## Configure the Parallel Replicat

Prepare the parameter file for the Replicat.

```
GGSCI (edw01.mobily.com.sa as ogg_admin@EDI) 3> edit params RBRM
```

```
REPLICAT RBRM  
SETENV (ORACLE_SID = EDI)  
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)  
USERID ogg_admin@EDI, PASSWORD oracle  
REPERERROR (DEFAULT, DISCARD)  
DISCARDFILE /u01/app/oracle/product/12.3.0/gghome_1/dirrpt/RBRM.dsc, PURGE  
MAP osm_repapi.REGIONS, TARGET osm_repapi.REGIONS;  
MAP osm_repapi.PROFESSIONS, TARGET osm_repapi.PROFESSIONS;  
MAP osm_repapi.EDU_LEVELS, TARGET osm_repapi.EDU_LEVELS;  
MAP osm_repapi.NATIONALITIES, TARGET osm_repapi.COUNTRIES;  
MAP osm_repapi.PACKAGES, TARGET osm_repapi.PACKAGES;  
MAP osm_repapi.PACKAGE_TYPES, TARGET osm_repapi.PACKAGE_TYPES;  
MAP osm_repapi.SIMS, TARGET osm_repapi.SIMS;  
MAP osm_repapi.CUSTOMERS, TARGET osm_repapi.CUSTOMERS;  
MAP osm_repapi.MOBILE_LINES, TARGET osm_repapi.MOBILE_LINES;  
MAP osm_repapi.CONTACTS, TARGET osm_repapi.CONTACTS;  
MAP osm_repapi.CALLS, TARGET osm_repapi.CALLS;  
MAP osm_repapi.CALL_DETAILS, TARGET osm_repapi.CALL_DETAILS;
```

```
GGSCI (edw01.mobily.com.sa as ogg_admin@EDI) 4> view params RBRM
```

```
REPLICAT RBRM  
SETENV (ORACLE_SID = EDI)  
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)  
USERID ogg_admin@EDI, PASSWORD oracle  
REPERERROR (DEFAULT, DISCARD)  
DISCARDFILE /u01/app/oracle/product/12.3.0/gghome_1/dirrpt/RBRM.dsc, PURGE  
MAP osm_repapi.REGIONS, TARGET osm_repapi.REGIONS;  
MAP osm_repapi.PROFESSIONS, TARGET osm_repapi.PROFESSIONS;  
MAP osm_repapi.EDU_LEVELS, TARGET osm_repapi.EDU_LEVELS;  
MAP osm_repapi.NATIONALITIES, TARGET osm_repapi.COUNTRIES;  
MAP osm_repapi.PACKAGES, TARGET osm_repapi.PACKAGES;  
MAP osm_repapi.PACKAGE_TYPES, TARGET osm_repapi.PACKAGE_TYPES;  
MAP osm_repapi.SIMS, TARGET osm_repapi.SIMS;  
MAP osm_repapi.CUSTOMERS, TARGET osm_repapi.CUSTOMERS;  
MAP osm_repapi.MOBILE_LINES, TARGET osm_repapi.MOBILE_LINES;  
MAP osm_repapi.CONTACTS, TARGET osm_repapi.CONTACTS;  
MAP osm_repapi.CALLS, TARGET osm_repapi.CALLS;  
MAP osm_repapi.CALL_DETAILS, TARGET osm_repapi.CALL_DETAILS;
```

```
GGSCI (edw01.mobily.com.sa as ogg_admin@EDI) 5>
```

## Define the Integrated Parallel Replicat

Define the integrated Replicat using the options: INTEGRATED and PARALLEL.

```
GGSCI (edw01.mobily.com.sa as ogg_admin@EDI) 6> ADD REPLICAT RBRM ,  
integrated, parallel, EXTTAIL ./dirdat/ta  
REPLICAT (Parallel) added.
```

```
GGSCI (edw01.mobily.com.sa as ogg_admin@EDI) 7>
```

## Start Extracts and Verify Status

Check status of the Extracts...

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 22> start *
```

```
Sending START request to MANAGER ...  
EXTRACT EBRM01 starting
```

```
Sending START request to MANAGER ...  
EXTRACT PBRM01 starting
```

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 23> info all
```

| Program | Status  | Group  | Lag at Chkpt | Time Since Chkpt |
|---------|---------|--------|--------------|------------------|
| MANAGER | RUNNING |        |              |                  |
| EXTRACT | RUNNING | EBRM01 | 00:03:22     | 00:16:32         |
| EXTRACT | RUNNING | PBRM01 | 00:00:00     | 00:16:32         |

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 24>
```

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 25> info EBRM01
```

|                     |        |                                 |                |
|---------------------|--------|---------------------------------|----------------|
| EXTRACT             | EBRM01 | Last Started 2018-01-27 20:50   | Status RUNNING |
| Checkpoint Lag      |        | 00:03:20 (updated 00:00:08 ago) |                |
| Process ID          |        | 12045                           |                |
| Log Read Checkpoint |        | Oracle Integrated Redo Logs     |                |
|                     |        | 2018-01-27 20:47:26             |                |
|                     |        | SCN 0.1195774 (1195774)         |                |

```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 26>
```



```
GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 26> info PBRM01

EXTRACT      PBRM01      Last Started 2018-01-27 20:50      Status RUNNING
Checkpoint Lag      00:00:00 (updated 00:00:05 ago)
Process ID        12046
Log Read Checkpoint File
/u01/app/oracle/product/12.3.0/gghome 1/dirdat/sa000000001
                        2018-01-27 20:50:06.124821 RBA 1495

GGSCI (dss01.mobily.com.sa as ogg_admin@BRM) 27>
```

## Start Replicat and Verify Status

Start and check status of the Replicat...

```
GGSCI (edw01.mobily.com.sa) 3> start *

Sending START request to MANAGER ...
REPLICAT RBRM starting

GGSCI (edw01.mobily.com.sa) 4> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
REPLICAT     RUNNING     RBRM       00:00:00      00:05:02

GGSCI (edw01.mobily.com.sa) 5>
```

```
GGSCI (edw01.mobily.com.sa) 5> info RBRM

REPLICAT RBRM      Last Started 2018-01-27 20:50      Status RUNNING
INTEGRATED
Parallel
Checkpoint Lag      00:00:00 (updated 00:00:04 ago)
Process ID        5889
Log Read Checkpoint File
/u01/app/oracle/product/12.3.0/gghome 1/dirdat/ta000000000
                        2018-01-27 20:50:38.250835 RBA 1575

GGSCI (edw01.mobily.com.sa) 6> exit
```

## Start Workload Simulator

Starting the workload simulator generate transaction on the source database according to the level of specified. Use the default which is recommended at the stage.

```
[oracle@oltp01 ~]$ sqlplus /nolog
```

```
SQL*Plus: Release 11.2.0.3.0 Production on Sat Feb 3 19:18:27 2018

Copyright (c) 1982, 2011, Oracle. All rights reserved.

BRM> conn osm_repapi/oracle
Connected.
BRM> exec call_trans_load;
```

## Verify Data Synchronization

Verify data is synchronized from source (BRM) to target (EDI) system database.

```
[oracle@oltp01 ~]$ sqlplus /nolog

SQL*Plus: Release 11.2.0.3.0 Production on Sat Jan 27 20:57:37 2018

Copyright (c) 1982, 2011, Oracle. All rights reserved.

BRM> conn osm_repapi/oracle
Connected.
BRM> exec call_trans_load;
```

### On Downstream

```
[oracle@dss01 gghome_1]$ cd dirdat
[oracle@dss01 dirdat]$ ls -al
total 84
drwxr-x--- 2 oracle oinstall 4096 Jan 27 20:50 .
drwxr-xr-x 29 oracle oinstall 4096 Jan 27 20:33 ..
-rw-r----- 1 oracle oinstall 1434 Jan 27 20:50 sa000000000
-rw-r----- 1 oracle oinstall 65811 Jan 27 21:02 sa000000001
[oracle@dss01 dirdat]$ ls -al
total 84
drwxr-x--- 2 oracle oinstall 4096 Jan 27 20:50 .
drwxr-xr-x 29 oracle oinstall 4096 Jan 27 20:33 ..
-rw-r----- 1 oracle oinstall 1434 Jan 27 20:50 sa000000000
-rw-r----- 1 oracle oinstall 66099 Jan 27 21:02 sa000000001
[oracle@dss01 dirdat]$
```

### On target

```
[oracle@edw01 gghome_1]$ cd dirdat
[oracle@edw01 dirdat]$ ls -al
total 124
drwxr-x--- 2 oracle oinstall 4096 Jan 27 20:29 .
drwxr-xr-x 28 oracle oinstall 4096 Jan 27 20:34 ..
-rw-r----- 1 oracle oinstall 115530 Jan 27 20:59 ta000000000
[oracle@edw01 dirdat]$ ls -al
total 124
drwxr-x--- 2 oracle oinstall 4096 Jan 27 20:29 .
drwxr-xr-x 28 oracle oinstall 4096 Jan 27 20:34 ..
-rw-r----- 1 oracle oinstall 117059 Jan 27 20:59 ta000000000
[oracle@edw01 dirdat]$
```

Use SQLPlus to verify row count is incrementing

```
[oracle@edw01 ~]$ sqlplus osm_repapi/oracle

SQL*Plus: Release 12.2.0.1.0 Production on Sat Feb 3 19:51:31 2018

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Last Successful login time: Sat Feb 03 2018 19:22:28 +03:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

EDI> select count(*) from customers;

COUNT (*)
-----
236494

EDI> /

COUNT (*)
-----
236497

EDI> select count(*) from sims;

COUNT (*)
-----
27324

EDI> /

COUNT (*)
-----
27337

EDI> select count(*) from contacts;

COUNT (*)
-----
2504150

EDI> /

COUNT (*)
-----
2504150

EDI> /

COUNT (*)
-----
2505487

EDI> select count(*) from calls;
```

```
      COUNT (*)  
-----  
      4998980  
  
EDI> /  
  
      COUNT (*)  
-----  
      5017366  
  
EDI>
```

# Summary

The Enterprise Data Warehouse (EDW) architecture must support high-performance load operation. The parallel processing is a very common method for handling large volume of data streaming originating from the enterprise systems. The transformation of data may occur during the loading phase, which also benefit from parallel processing loading technique.

One of the add-value of Oracle GoldenGate 12c is the methods for handling the load phase. Oracle GoldenGate 12c offers five types of Replicat for applying the transaction to the Enterprise Data Warehouse (EDW) repository.

- Classic Replicat
- Integrated Replicat
- Coordinated Replicat
- Parallel non-integrated Replicat
- Parallel integrated Replicat

Unless the environment requires classic Replicat, use any of the other Replicat types. The coordinated Replicat implement internal high degree of parallel (DOP) processing starting with DOP=25, which can be change by altering the Replicat definition, however; spinning off extremely unnecessary high DOP creates negative impact on the system by consuming system resources.

The parallel, integrated Replicat is interesting. The Replicat is defined using the keywords parallel and integrated. With parallel-integrated Replicat, a lightweight API generates the logical change records (LCRs), where a coordinator process spin off multiple appliers processes.

When the core systems for the Enterprise Data Warehouse are heterogeneous, the use of different modes for capture and apply is definable as listed next.

- Classic capture (oracle and non-oracle) and non-integrated Replicat
- Classic capture (oracle and non-oracle) and integrated Replicat
- Integrated capture and non-integrated Replicat
- Integrated capture and integrated Replicat

The next chapter demonstrates complex data transformation. Where Oracle GoldenGate 12c Replicat transform the data directly or by invoking complex PL/SQL store procedures and functions, and extends the capabilities by invoking Java code running on the database JVM.

# Develop Complex Data Transformation for BI Applications

## Overview

The objective of this chapter is to develop complex data transformation using Oracle GoldenGate 12c capabilities. The target EDW database is characterized by highly de-normalized and fact tables, in such cases, Oracle GoldenGate 12c column mapping and data transformations enable high-performance apply by empowering:

- Columns mapping
- Executing SQL queries
- Invoking of stored database PL/SQL sub-programs
- From within a procedure, invoking stored database Java classes
- Call-out to C/C++ user-exit

## Column Mapping

This technique map columns on the source table with columns on the target table. The source and target tables may have different names, different owners, and different columns names and data types. When performing column mapping, start by using the keyword `USEDEFAULTS` which maps similar columns names and types, then explicitly map a target column to one or source columns. Apply Oracle GoldenGate 12c functions for manipulation of source values then assign to the target column. For example: Column 1 (target) = Column 1 (source) + Column 2 (source).

## Executing SQL Queries

The purpose of executing SQL queries is to fetch lookup value to populate the de-normalized columns. For complex data transformations, from within the SQL queries invoke database stored PL/SQL functions, which return the values assigned to the target columns. The database functions `TO_CHAR`, `TO_DATE` and `TO_NUMBER` are implemented to covert the data types to match the target columns data types.

## Invoking of Stored Database PL/SQL Sub-Programs

Invoking stored database PL/SQL procedure is a common technique for performing data transformations. For each transactional row, the stored database PL/SQL procedure is invoked to perform data transformations manipulation and return the value to the target column using the Oracle GoldenGate 12c function `@GETVAL`. Figure 1 shows the flow when invoking a stored procedure from the Replicat. The invoked PL/SQL procedure invokes stored database functions to perform further complex data transformations.

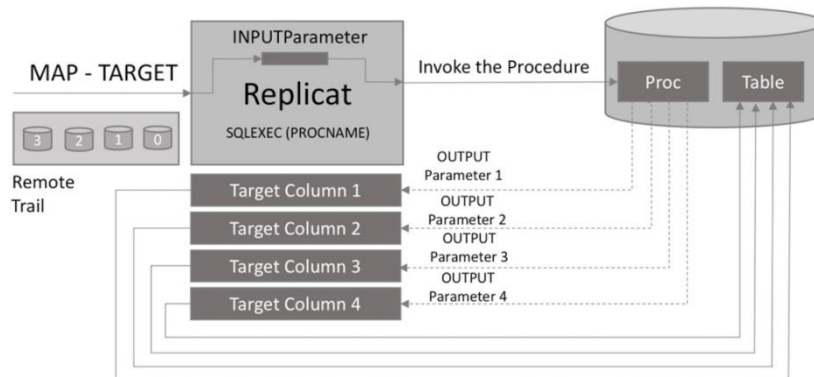


Figure 1: Invoking a stored database PL/SQL Procedure

## Invoking Database Stored Java Classes

Invoking stored database Java class is performed indirectly. Java classes stored in the database are called from PL/SQL procedures. Because these Java classes run as JVM, hence set the parameter `JAVA_POOL_SIZE` and monitor contentions, if necessary. This parameter is dynamic as show next.

```
SQL> alter system set java_pool_size=64K scope=both;
```

```
System altered.
```

```
SQL>
```

## Call-out to C/C++ User Exits

You can employ user exits as an alternative to, or in conjunction with, the column-conversion functions that are available within GoldenGate. User exits can be a better alternative to the built-in functions because, with a user exit, data is processed once (when extracted) rather than twice (extracted and then read again to perform the transformation).

## Develop Transformation on CALL\_DETAILS

The database procedure TRANSFORM\_CALL\_DETAILS is invoked from within the Replicat RBRM, it takes the call ID (pCall\_id) as input, then perform all fetches and data manipulations to transform the data. The TRANSFORM\_CALL\_DETAILS procedure has the following structure.

```
EDI> desc call_details
```

| Name                 | Null?    | Type           |
|----------------------|----------|----------------|
| CALL ID              | NOT NULL | VARCHAR2 (64)  |
| CALL_TIMESTAMP       | NOT NULL | TIMESTAMP (6)  |
| CALL_MINUTE_DURATION | NOT NULL | NUMBER (5,2)   |
| CUST_NO              |          | NUMBER         |
| CUST_NAME            |          | VARCHAR2 (128) |
| SIM_CODE1            |          | VARCHAR2 (32)  |
| SIM_EXPIRE_DATE      |          | DATE           |
| SIM_CELL_NO          |          | VARCHAR2 (16)  |
| DAILED TO AREA CODE  |          | VARCHAR2 (8)   |
| DAILED_TO_NO         |          | VARCHAR2 (16)  |
| DAILED_TO_NAME       |          | VARCHAR2 (128) |
| CALL_FEES            |          | NUMBER (6,2)   |
| REG_NAME             |          | VARCHAR2 (128) |
| EDUL_NAME            |          | VARCHAR2 (128) |
| PROF_DESC            |          | VARCHAR2 (128) |
| CNTRY_NAME           |          | VARCHAR2 (128) |
| PKGT_DESC            |          | VARCHAR2 (128) |
| REP_GROUP            |          | VARCHAR2 (8)   |
| OS_USER              |          | VARCHAR2 (32)  |
| HOST NAME            |          | VARCHAR2 (32)  |
| OPERATION_TYPE       |          | VARCHAR2 (16)  |
| OPERATION_TIMESTAMP  |          | TIMESTAMP (6)  |

```
EDI>
```

Create the user-defined data transformation procedure -  
TRANSFORM\_CALL\_DETAILS

```
Create Or Replace Procedure Transform_Call_Details (
    Pcall_Id In Varchar2,
    Pcust_No Out Number,
    Pcust_Name Out Varchar2,
    Psim_Code1 Out Varchar2,
    Psim_Expire_Date Out Date,
    Psim_Cell_No Out Varchar2,
    Pdailed_To_Area_Code Out Varchar2,
    Pdailed_To_No Out Varchar2,
    Pdailed_To_Name Out Varchar2,
    Pcall_Fees Out Number,
    Preg_Name Out Varchar2,
    Pedul_Name Out Varchar2,
    Pprof_Name Out Varchar2,
    Pcuntry_Name Out Varchar2,
    Ppkg Desc Out Varchar2) Is
```



```

--
Vcust_No                Number;
Vcust_Name              Varchar2(128);
Vsim_Code1              Varchar2(64);
Vsim_Expire_Date        Date;
Vsim_Cell_No            Varchar2(32);
Vcntry_Area_Code        Varchar2(8);
Vcont_Mobile_No         Varchar2(16);
Vdailed_To_Name         Varchar2(128);
Vreg_Name               Varchar2(128);
Vedul_Name              Varchar2(128);
Vprof_Name              Varchar2(128);
Vcntry_Name             Varchar2(128);
Vpkggt_Desc             Varchar2(132);
Vcall_Min               Number(5,2);
Vpkggt_Call_Fees_Min_Local Number(4,2);
Verr                    Varchar2(2000);
Vline                   Number:=0;

Begin
  Vline := 1;
  Select C.Cust_No, C.Sim_Code1, C.Cntry_Area_Code, C.Cont_Mobile_No,
c.Call_Minuite_Duration
  Into Vcust_No, Vsim_Code1, Vcntry_Area_Code, Vcont_Mobile_No, Vcall_Min
  From   Calls C
  Where  C.Call_Id = Pcall_Id;
  Pcust_No := Vcust_No;
  Psim_Code1 := Vsim_Code1;
  Pdailed_To_Area_Code := Vcntry_Area_Code;
  Pdailed_To_No := Vcont_Mobile_No;
  --
  Vline := 2;
  Select Cust_Name Into Vcust_Name
  From   Customers
  Where  Cust_No = Vcust_No;
  Pcust_Name := Vcust_Name;
  --
  Vline := 3;
  Select Sim_Code1 Into Vsim_Code1
  From   Calls
  Where  Call_Id = Pcall_Id;
  Psim_Code1 := Vsim_Code1;
  --
  Vline := 4;
  Select Sim_Expire_Date Into Vsim_Expire_Date
  From   Mobile_Lines
  Where  Sim_Code1 = Vsim_Code1 And
        Cust_No = Vcust_No;
  Psim_Expire_Date:=Vsim_Expire_Date;
  --
  Vline := 5;
  Select Cont_F_Name||' '||Cont_L_Name
  Into   Vdailed_To_Name
  From   Contacts
  Where  Cust_No          = Vcust_No And
        Sim_Code1        = Vsim_Code1 And
        Cntry_Area_Code = Vcntry_Area_Code And
        Cont_Mobile_No  = Vcont_Mobile_No;
  Pdailed_To_Name := Vdailed_To_Name;
  --

```

```

Vline := 6;
Select R.Reg_Name
Into   Vreg_Name
From   Customers C, Regions R
Where  C.Cust_No = Vcust_No And
       C.Reg_Id = R.Reg_Id;
Preg_Name := Vreg_Name;
--
Vline := 7;
Select L.Edul_Name Into Vedul_Name
From   Customers C, Edu_Levels L
Where  C.Cust_No = Vcust_No And
       C.Edul_Id = L.Edul_Id;
Pedul_Name := Vedul_Name;
--
Vline := 8;
Select P.Prof_Name Into Vprof_Name
From   Customers C, Professions P
Where  C.Cust_No = Vcust_No And
       C.Prof_Id = P.Prof_Id;
Pprof_Name := Vprof_Name;
--
Vline := 9;
Select Ct.Cntry_Name Into Vcntry_Name
From   Customers C, Countries Ct
Where  C.Cust_No = Vcust_No And
       C.Cntry_Code = Ct.Cntry_Code;
Pcntry_Name := Vcntry_Name;
--
Vline := 10;
Select P.Pkgt_Desc, P.Pkgt_Call_Fees_Per_Min_Local, s.sim_cell_no Into
Vpkggt_Desc, Vpkggt_Call_Fees_Min_Local, vsim_cell_no
From   Package_Types P, Sims S
Where  P.Pkgt_Code = S.Pkgt_Code And
       S.Sim_Code1 = Vsim_Code1;
Ppkggt_Desc := Vpkggt_Desc;
Pcall_Fees := Round(Vpkggt_Call_Fees_Min_Local * Vcall_Min, 2);
Psim_Cell_No := vsim_cell_no;
Exception
  When Others Then
    Verr:= Sqlerrm;
    Insert Into Trans_Log Values ('LINE='||Vline||' -
'||'CALL_ID='||Pcall_Id||' --> '||Verr);
    Commit;
End

```

## Grant Database Privileges to Execute the Procedure

Grant execute on the data transformation procedure -  
 TRANSFORM\_CALL\_DETAILS, to OGG\_ADMIN.

```

EDI> conn osm_repapi/oracle
Connected.
EDI> grant execute on Transform_call_details to ogg_admin;

```

```
Grant succeeded.
```

```
EDI>
```

## Re-Configure the Coordinated Parallel Replicat

Reconfigure the integrated Replicat to call-out the stored procedure and assigning value to the table CALL\_DETAILS additional columns.

```
[oracle@edw01 ~]$ cd $OGG_HOME
[oracle@edw01 gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE_12.3.0.1.0_PLATFORMS_170721.0154_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (edw01.mobily.com.sa) 1> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
-----
MANAGER      RUNNING
REPLICAT     RUNNING     RBRM       00:00:00      00:00:03

GGSCI (edw01.mobily.com.sa) 2>
```

```
GGSCI (edw01.mobily.com.sa) 2> edit params RBRM
```

```
REPLICAT RBRM
SETENV (ORACLE_SID = EDI)
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg admin@EDI, PASSWORD oracle
REPERERROR (DEFAULT, DISCARD)
DISCARDFILE /u01/app/oracle/product/12.3.0/gghome_1/dirrpt/RBRM.dsc, PURGE,
megabytes 1000
MAP osm_repapi.REGIONS, TARGET osm_repapi.REGIONS;
MAP osm_repapi.PROFESSIONS, TARGET osm_repapi.PROFESSIONS;
MAP osm_repapi.EDU LEVELS, TARGET osm_repapi.EDU LEVELS;
MAP osm_repapi.COUNTRIES, TARGET osm_repapi.COUNTRIES;
MAP osm_repapi.PACKAGE_TYPES, TARGET osm_repapi.PACKAGE_TYPES;
MAP osm_repapi.SIMS, TARGET osm_repapi.SIMS;
MAP osm_repapi.CUSTOMERS, TARGET osm_repapi.CUSTOMERS;
MAP osm_repapi.MOBILE_LINES, TARGET osm_repapi.MOBILE_LINES;
MAP osm_repapi.CONTACTS, TARGET osm_repapi.CONTACTS;
MAP osm_repapi.CALLS, TARGET osm_repapi.CALLS;
MAP osm_repapi.CALL_DETAILS, TARGET osm_repapi.CALL_DETAILS, &
    SQLEXEC (SPNAME osm_repapi.Transform_call_details, &
              ID TCD, &
              PARAMS (pCall_id = CALL ID)), &
```

```

COLMAP (USEDEFAULTS, &
        CUST_NO = @GETVAL(TCD.pCust_no), &
        CUST_NAME = @GETVAL(TCD.pCust_name), &
        SIM_CODE1 = @GETVAL(TCD.pSim_code1), &
        SIM_EXPIRE_DATE = @GETVAL(TCD.PSIM_EXPIRE_DATE), &
        SIM_CELL_NO = @GETVAL(TCD.PSIM_CELL_NO), &
        DAILED_TO_AREA_CODE =
        @GETVAL(TCD.PDAILED_TO_AREA_CODE), &
        DAILED_TO_NO = @GETVAL(TCD.PDAILED_TO_NO), &
        DAILED_TO_NAME = @GETVAL(TCD.PDAILED_TO_NAME), &
        CALL_FEES = @GETVAL(TCD.PCALL_FEES), &
        REG_NAME = @GETVAL(TCD.PREG_NAME), &
        EDUL_NAME = @GETVAL(TCD.PEDUL_NAME), &
        PROF_DESC = @GETVAL(TCD.PPROF_NAME), &
        CNTRY_NAME = @GETVAL(TCD.PCNTRY_NAME), &
        PKGT_DESC = @GETVAL(TCD.PPKGT_DESC), &
        REP_GROUP= @GETENV('GGENVIRONMENT', 'GROUPNAME'), &
        OS_USER=@GETENV('GGENVIRONMENT', 'OSUSERNAME'), &
        HOST_NAME=@GETENV('GGENVIRONMENT', 'HOSTNAME'), &
        OPERATION_TYPE=@GETENV('GGHEADER', 'OPTYPE'), &
        OPERATION_TIMESTAMP=@GETENV('GGHEADER',
        'COMMITTIMESTAMP'));

```

To activate the data transformation configuration, stop and restart the Replicat.

```

GGSCI (edw01.mobily.com.sa) 10> stop *

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

GGSCI (edw01.mobily.com.sa) 21> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
REPLICAT     STOPPED     RBRM       00:00:00     00:00:10

GGSCI (edw01.mobily.com.sa) 22> start *

Sending START request to MANAGER ...
REPLICAT RBRM starting

GGSCI (edw01.mobily.com.sa) 24> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
REPLICAT     RUNNING     RBRM       00:00:00     00:00:02

GGSCI (edw01.mobily.com.sa) 25>

```

## Start the Workload Simulator

Start the workload simulator using the default parameters

```
BRM> conn osm_repapi/oracle
Connected.
BRM> exec call_trans_load;
```

## Verify Data Synchronization

Verify data synchronization using SQLPlus or GGSCI report. SQLPlus is used here to verify row incrementing while the workload simulator is running on source (BRM) database.

```
[oracle@dss01 ~]$ cd $OGG_HOME
[oracle@dss01 gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE_12.3.0.1.0_PLATFORMS_170721.0154_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (dss01.mobily.com.sa) 1> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
-----
MANAGER      RUNNING
EXTRACT      RUNNING     EBRM01     00:01:28     00:00:00
EXTRACT      RUNNING     PBRM01     00:00:00     00:00:06

GGSCI (dss01.mobily.com.sa) 2>
```

The target instance...

```
[oracle@edw01 ~]$ cd $OGG_HOME
[oracle@edw01 gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE_12.3.0.1.0_PLATFORMS_170721.0154_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (edw01.mobily.com.sa) 1> info all
```

| Program  | Status  | Group | Lag at Chkpt | Time Since Chkpt |
|----------|---------|-------|--------------|------------------|
| MANAGER  | RUNNING |       |              |                  |
| REPLICAT | RUNNING | RBRM  | 00:00:00     | 00:00:01         |

GGSCI (edw01.mobily.com.sa) 2>

```
EDI> conn osm_repapi/oracle
Connected.
EDI>
```

```
EDI> select count(*) from mobile_lines;
```

```

COUNT(*)
-----
      2207

```

```
EDI> /
```

```

COUNT(*)
-----
      2213

```

```
EDI> select count(*) from contacts;
```

```

COUNT(*)
-----
    1135108

```

```
EDI> /
```

```

COUNT(*)
-----
    1139192

```

```
EDI> select count(*) from calls;
```

```

COUNT(*)
-----
    402332

```

```
EDI> /
```

```

COUNT(*)
-----
    403991

```

```
EDI> select count(*) from call_details;
```

```

COUNT(*)
-----
    408793

```

```
EDI> /
```

```

COUNT (*)
-----
410462

EDI>

```

## Verify Data Transformation

Query data from CALL\_DETAILS table.

```

EDI> column CUST_NAME format A20
EDI> column DAILED_TO_NAME format A30
EDI> select count(*) CALLS, CUST_NO, CUST_NAME, DAILED_TO_NAME
2   from   call_details
3   where  rownum < 21
4   group by CUST_NO, CUST_NAME, DAILED_TO_NAME;

```

| CALLS | CUST_NO | CUST_NAME           | DAILED_TO_NAME      |
|-------|---------|---------------------|---------------------|
| 1     | 1001818 | Fawzi Q. Alawami    | Mohammed Alawami    |
| 1     | 1001818 | Fawzi Q. Alawami    | Mohammed Alibrahim  |
| 1     | 1001818 | Fawzi Q. Alawami    | Saeed Alswaimil     |
| 1     | 1001818 | Fawzi Q. Alawami    | Hassan Alshikhs     |
| 1     | 1001818 | Fawzi Q. Alawami    | Nour Alawami        |
| 1     | 1001818 | Fawzi Q. Alawami    | Saeed Alamodi       |
| 1     | 1001818 | Fawzi Q. Alawami    | Philip Alshammari   |
| 1     | 1001818 | Fawzi Q. Alawami    | Saeed Al-Qahtani    |
| 1     | 1001818 | Fawzi Q. Alawami    | Abudllah Al-Qahtani |
| 1     | 1001818 | Fawzi Q. Alawami    | Osmah Al-Ghamdi     |
| 1     | 1001818 | Fawzi Q. Alawami    | Saeed Alsaihati     |
| ----- |         |                     |                     |
| CALLS | CUST_NO | CUST_NAME           | DAILED_TO_NAME      |
| 2     | 1001818 | Fawzi Q. Alawami    | Osmah Alsaihati     |
| 1     | 1001818 | Fawzi Q. Alawami    | Deema Al-Qahtani    |
| 1     | 1001818 | Fawzi Q. Alawami    | Deema Alsalem       |
| 1     | 1001819 | Abudllah L. Alawami | Talat Alsuwaidi     |
| 1     | 1001818 | Fawzi Q. Alawami    | Hassan Alibrahim    |
| 1     | 1001818 | Fawzi Q. Alawami    | Mohammed Alamodi    |
| 1     | 1001818 | Fawzi Q. Alawami    | Ali Alshammari      |
| 1     | 1001818 | Fawzi Q. Alawami    | Hassan Alsaihati    |

19 rows selected.

```
EDI>
```

## Generate Replicat Statistic Report

The following are optional parameters, however; adding them to the Replicat enhances the apply performance.

```
GGSCI (edw01.mobily.com.sa) 2> stats replicat RBRM
```

Sending STATS request to REPLICAT RBRM ...

Parallel Replicat Statistics:

|                        |            |
|------------------------|------------|
| Total transactions     | 3601268.00 |
| Large transactions     | 0.00       |
| Total metadata changes | 0.00       |
| DDL operations         | 0.00       |
| Replicated procedures  | 0.00       |
| Event Actions          | 0.00       |
| Datatype functionality | 0.00       |
| Store Procedures       | 755415.00  |
| Checkpoint errors      | 0.00       |
| Transactions retried   | 0.00       |

Start of Statistics at 2018-02-05 15:25:55.

Replicating from OSM\_REPAPI.CUSTOMERS to OSM\_REPAPI.CUSTOMERS:

\*\*\* Total statistics since 2018-02-05 14:57:38 \*\*\*

|                  |        |
|------------------|--------|
| Total inserts    | 754.00 |
| Total updates    | 0.00   |
| Total deletes    | 0.00   |
| Total discards   | 0.00   |
| Total operations | 754.00 |

\*\*\* Daily statistics since 2018-02-05 14:57:38 \*\*\*

|                  |        |
|------------------|--------|
| Total inserts    | 754.00 |
| Total updates    | 0.00   |
| Total deletes    | 0.00   |
| Total discards   | 0.00   |
| Total operations | 754.00 |

\*\*\* Hourly statistics since 2018-02-05 15:00:00 \*\*\*

|                  |        |
|------------------|--------|
| Total inserts    | 694.00 |
| Total updates    | 0.00   |
| Total deletes    | 0.00   |
| Total discards   | 0.00   |
| Total operations | 694.00 |

\*\*\* Latest statistics since 2018-02-05 14:57:38 \*\*\*

|                  |        |
|------------------|--------|
| Total inserts    | 754.00 |
| Total updates    | 0.00   |
| Total deletes    | 0.00   |
| Total discards   | 0.00   |
| Total operations | 754.00 |

Replicating from OSM\_REPAPI.SIMS to OSM\_REPAPI.SIMS:

\*\*\* Total statistics since 2018-02-05 14:57:38 \*\*\*

|                  |         |
|------------------|---------|
| Total inserts    | 4138.00 |
| Total updates    | 0.00    |
| Total deletes    | 0.00    |
| Total discards   | 0.00    |
| Total operations | 4138.00 |

\*\*\* Daily statistics since 2018-02-05 14:57:38 \*\*\*



```

Total inserts                      4138.00
Total updates                      0.00
Total deletes                      0.00
Total discards                     0.00
Total operations                    4138.00

*** Hourly statistics since 2018-02-05 15:00:00 ***
Total inserts                      3813.00
Total updates                      0.00
Total deletes                      0.00
Total discards                     0.00
Total operations                    3813.00

*** Latest statistics since 2018-02-05 14:57:38 ***
Total inserts                      4138.00
Total updates                      0.00
Total deletes                      0.00
Total discards                     0.00
Total operations                    4138.00

Replicating from OSM_REPAPI.MOBILE_LINES to OSM_REPAPI.MOBILE_LINES:

*** Total statistics since 2018-02-05 14:57:38 ***
Total inserts                      4138.00
Total updates                      0.00
Total deletes                      0.00
Total discards                     0.00
Total operations                    4138.00

*** Daily statistics since 2018-02-05 14:57:38 ***
Total inserts                      4138.00
Total updates                      0.00
Total deletes                      0.00
Total discards                     0.00
Total operations                    4138.00

*** Hourly statistics since 2018-02-05 15:00:00 ***
Total inserts                      3813.00
Total updates                      0.00
Total deletes                      0.00
Total discards                     0.00
Total operations                    3813.00

*** Latest statistics since 2018-02-05 14:57:38 ***
Total inserts                      4138.00
Total updates                      0.00
Total deletes                      0.00
Total discards                     0.00
Total operations                    4138.00

Replicating from OSM_REPAPI.CONTACTS to OSM_REPAPI.CONTACTS:

*** Total statistics since 2018-02-05 14:57:38 ***
Total inserts                      2077042.00
Total updates                      0.00
Total deletes                      0.00
Total discards                     0.00
Total operations                    2077042.00

```

```

*** Daily statistics since 2018-02-05 14:57:38 ***
Total inserts                2077042.00
Total updates                 0.00
Total deletes                 0.00
Total discards                0.00
Total operations              2077042.00

```

```

*** Hourly statistics since 2018-02-05 15:00:00 ***
Total inserts                1912524.00
Total updates                 0.00
Total deletes                 0.00
Total discards                0.00
Total operations              1912524.00

```

```

*** Latest statistics since 2018-02-05 14:57:38 ***
Total inserts                2077042.00
Total updates                 0.00
Total deletes                 0.00
Total discards                0.00
Total operations              2077042.00

```

Replicating from OSM\_REPAPI.CALLS to OSM\_REPAPI.CALLS:

```

*** Total statistics since 2018-02-05 14:57:38 ***
Total inserts                754764.00
Total updates                 0.00
Total deletes                 0.00
Total discards                0.00
Total operations              754764.00

```

```

*** Daily statistics since 2018-02-05 14:57:38 ***
Total inserts                754764.00
Total updates                 0.00
Total deletes                 0.00
Total discards                0.00
Total operations              754764.00

```

```

*** Hourly statistics since 2018-02-05 15:00:00 ***
Total inserts                690390.00
Total updates                 0.00
Total deletes                 0.00
Total discards                0.00
Total operations              690390.00

```

```

*** Latest statistics since 2018-02-05 14:57:38 ***
Total inserts                754764.00
Total updates                 0.00
Total deletes                 0.00
Total discards                0.00
Total operations              754764.00

```

Replicating from OSM\_REPAPI.CALL\_DETAILS to OSM\_REPAPI.CALL\_DETAILS:

```

*** Total statistics since 2018-02-05 14:57:38 ***
Total inserts                754765.00
Total updates                 0.00
Total deletes                 0.00
Total discards                0.00
Total operations              754765.00

```

```
*** Daily statistics since 2018-02-05 14:57:38 ***
Total inserts          754765.00
Total updates          0.00
Total deletes          0.00
Total discards         0.00
Total operations       754765.00

*** Hourly statistics since 2018-02-05 15:00:00 ***
Total inserts          690392.00
Total updates          0.00
Total deletes          0.00
Total discards         0.00
Total operations       690392.00

*** Latest statistics since 2018-02-05 14:57:38 ***
Total inserts          754765.00
Total updates          0.00
Total deletes          0.00
Total discards         0.00
Total operations       754765.00
```

End of Statistics.

GGSCI (edw01.mobily.com.sa) 3>

# Summary

The data transformation phase is one of the major milestones when deploying an Enterprise Data Warehouse (EDW). Oracle GoldenGate 12c perform non-intrusive change data capture (CDC) from the core transactional systems, in real-time it reach the target EDW, where transformation must is must occur. The data transformations empower any of the following techniques:

- Columns mapping
- Executing SQL queries
- Invoking of stored database PL/SQL sub-programs
- From within a procedure, invoking stored database Java classes
- Call-out to C/C++ user-exit

The column mapping allows mapping the target column to one of more source columns. Oracle GoldenGate 12c functions are used to manipulate the values and then assigned to the target column.

The execution of SQL queries from within the Replicat allows access to the database, fetching data for de-normalized columns or calling a stored database function to perform complex data transformation the functions return values assigned to target columns.

The most common technique for data transformations is invoking database stored procedure. The database stored procedure accepts one or more input parameters from the source columns, and invoke the procedure in the EDW database. The return values, output parameters, are assigned to the target columns. In such technique the procedure access the local and remote databases, use the database functions to perform any types of data transformation, and then finally use Oracle GoldenGate 12c functions for performing additional data transformations.

The use of database stored Java classes is common because of JVM in the database. This is achieved indirectly, where Oracle GoldenGate 12c invokes PL/SQL stored procedure, and the stored procedure invokes the database stored Java class.

One of the supported, but less commonly used for data transformations is user exits. They are external C/C++ invoke for complex specialized data transformations.

The next chapter move the Enterprise Data Warehouse (EDW) to memory using Oracle Database 12c R2 data warehouse features, namely: In-memory option, advanced compression and in-memory materialized view.

# Deploy Real-Time BI Applications

## Overview

This chapter covers the building blocks of the Enterprise Data Warehouse (EDW) repository. EDW database repository must be configured to support extremely intrusive analytical queries. The Enterprise Data Warehouse must scale linearly with time and data growth as the intrusive analytical queries become more resource intensive with time. The following list of EDW is discussed and implemented.

- Database parameters settings
- Big tablespace implementation and large database I/O block
- Temporary groups
- Database partitioning
- In-Memory option
- Advanced Compression
- Materialized views

## Real Application Cluster and Automatic Storage Management

Though, this Enterprise Data Warehouse (EDW) deployed on single-instance database, surely in a production environment, an EDW highly likely to be deployed on multi-instance database running Oracle Real Application Cluster (RAC). The primary reason is *parallelism*. In RAC, parallel processing is extendable to run across nodes to speed up the queries execution, where a coordinator and set of slave's processes participate to execute the query across nodes.

It's expected the storage growth is linear in an Enterprise Data Warehouse (EDW). Though this workshop uses file system for database storage, it's highly recommended to deploy Automatic Storage Management (ASM), it delivers online administration, in-place disk management and cost-effective for beta bytes storage.

# Optimizer Parameters for Data Warehouse Queries

Sizing the following essential database parameters enhances query intensive database applications. Perform SPFILE backup, which is handy to use in case you're trapped to start the instance due to parameters set incorrectly.

```
EDI> conn / as sysdba
Connected.
EDI>
```

```
EDI> create pfile='/tmp/initEDI.ora' from spfile;

File created.

EDI>
```

In Linux, set large temporary file system that appear as a mounted file system, but one which uses virtual memory instead of a persistent storage device. Update /etc/fstab to make tmpfs size permanent on next reboot.

```
/dev/mapper/vg_edw01-lv_root / ext4 defaults
1 1
UUID=8dd2f934-6b7a-43c6-afa5-af4a7a9a971b /boot ext4
defaults 1 2
/dev/mapper/vg_edw01-lv_home /home ext4 defaults
1 2
/dev/mapper/vg_edw01-lv_swap swap swap defaults
0 0
tmpfs /dev/shm tmpfs size=40g 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
```

```
[oracle@edw01 ~]$ su
Password:
[root@edw01 oracle]# mount -t tmpfs shmfs -o size=40g /dev/shm
[root@edw01 oracle]# exit
```

Set the following Database Parameters, other parameters may be applied to influence the optimizer performance for specific queries.

| Parameter         | Description                                                                                                       |
|-------------------|-------------------------------------------------------------------------------------------------------------------|
| memory_max_target | Specifies the maximum value to which a DBA can set the MEMORY_TARGET initialization parameter.                    |
| memory_target     | specifies the Oracle system-wide usable memory. The database tunes memory to the MEMORY_TARGET value, reducing or |

|                                            |                                                                                                                                                                                                                                                                                                             |
|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                            | enlarging the SGA and PGA as needed.                                                                                                                                                                                                                                                                        |
| <code>db_file_multiblock_read_count</code> | It specifies the maximum number of blocks read in one I/O operation during a sequential scan. The total number of I/Os needed to perform a full table scan depends on such factors as the size of the table, the multiblock read count, and whether parallel execution is being utilized for the operation. |
| <code>olap_page_pool_size</code>           | Specifies (in bytes) the size of the OLAP pool.                                                                                                                                                                                                                                                             |
| <code>star_transformation_enabled</code>   | Determines whether a cost-based query transformation will be applied to star queries.                                                                                                                                                                                                                       |

```

EDI> alter system set MEMORY_MAX_TARGET=40G scope=spfile;

System altered.

EDI> alter system set MEMORY_TARGET=16000M scope=spfile;

System altered.

EDI> alter system set db_file_multiblock_read_count=512 scope=spfile;

System altered.

EDI> alter system set olap_page_pool_size=2000M scope=spfile;

System altered.

EDI> alter system set star_transformation_enabled=true scope=spfile;

System altered.

EDI>

```

Restart the database to activate static parameters.

```

EDI> startup force
ORACLE instance started.

Total System Global Area          1.6777E+10 bytes
Fixed Size                        12179616 bytes
Variable Size                     6509561696 bytes
Database Buffers                  1.0234E+10 bytes
Redo Buffers                       21372928 bytes
Database mounted.
Database opened.
EDI>

```

## Implement 16K Block I/O for BIGFILE Tablespace

Enlarging I/O by implementing a large block size. Oracle Database 12c supports up to five block sizes: 2K, 4K, 8K, 16K and 32K. Due to limitation on Linux, a 16K I/O block size is implemented.

```
EDI> alter system set db_16k_cache_size=10G scope=spfile;

System altered.

EDI> startup force
ORACLE instance started.

Total System Global Area          1.6777E+10 bytes
Fixed Size                        12179616 bytes
Variable Size                     5234493280 bytes
Database Buffers                  1.1509E+10 bytes
Redo Buffers                      21372928 bytes
Database mounted.
Database opened.
EDI>
```

Create tablespace dedicated for 16K block size. Notice the tablespace is BIGFILE tablespace, which means the tablespace created using one-and-one data file. The Auto extend option enable the data file extended automatically when it's under space pressure.

```
EDI> create bigfile tablespace OSM_DAT_16K
  2  blocksize 16K
  3  datafile '/u01/app/oracle/oradata/EDI/datafile/OSM_DAT_16K_01.dbf' size
200G
  4  autoextend on;

Tablespace created.

EDI>
```

Make the tablespace OSM\_DAT\_16K is the default for schema OSM\_REPAPI, migrate the relevant BI applications tables to this tablespace.

```
EDI> alter user osm_repapi default tablespace OSM_DAT_16K;

User altered.

EDI> alter user osm_repapi quota unlimited on OSM_DAT_16K;

User altered.

EDI> alter table osm_repapi.call_details move tablespace OSM_DAT_16K;

Table altered.
```



```

EDI> alter table osm_repapi.calls move tablespace OSM_DAT_16K;

Table altered.

EDI> alter table osm_repapi.contacts move tablespace OSM_DAT_16K;

Table altered.

EDI> alter table osm_repapi.mobile_lines move tablespace OSM_DAT_16K;

Table altered.

EDI> alter table osm_repapi.sims move tablespace OSM_DAT_16K;

Table altered.

EDI> alter table osm_repapi.customers move tablespace OSM_DAT_16K;

Table altered.

EDI> alter table osm_repapi.package_types move tablespace OSM_DAT_16K;

Table altered.

EDI>

```

## Implement Temporary Groups

Temporary group enables temporary space allocation for long running queries demanding temporary sort area.

```

EDI> select file_name from dba_temp_files;

FILE_NAME
-----
/u01/app/oracle/oradata/EDI/datafile/ol_mf_temp_f6p2ppoy_.tmp

EDI> create temporary tablespace temp01
  2 tempfile '/u01/app/oracle/oradata/EDI/datafile/temp01_01.dbf' size 500M
  3 autoextend on
  4 tablespace group temp_ts_group;

Tablespace created.

EDI> alter tablespace temp tablespace group temp_ts_group;

Tablespace altered.

EDI> select * from dba_tablespace_groups;

GROUP_NAME                                TABLESPACE_NAME
-----
TEMP_TS_GROUP                             TEMP
TEMP_TS_GROUP                             TEMP01

```

```

EDI>

EDI> alter user OSM_REPAPI temporary tablespace temp_ts_group;

User altered.

EDI>

```

## Online Database Structure Partitioning

The following applies table partitioning to the CALL\_DETAILS table. Partitioned table delivers better performance, in addition provides a management window for volatile tables. The partitioning key is based on the column CALL\_TIMESTAMP year.

```

EDI> desc call_details

```

| Name                 | Null?    | Type           |
|----------------------|----------|----------------|
| CALL_ID              | NOT NULL | VARCHAR2 (64)  |
| CALL_TIMESTAMP       | NOT NULL | TIMESTAMP (6)  |
| CALL_MINUTE_DURATION |          | NUMBER (5,2)   |
| CUST_NO              |          | NUMBER         |
| CUST_NAME            |          | VARCHAR2 (128) |
| SIM_CODE1            |          | VARCHAR2 (32)  |
| SIM_EXPIRE_DATE      |          | DATE           |
| SIM_CELL_NO          |          | VARCHAR2 (16)  |
| DAILED_TO_AREA_CODE  |          | VARCHAR2 (8)   |
| DAILED_TO_NO         |          | VARCHAR2 (16)  |
| DAILED_TO_NAME       |          | VARCHAR2 (128) |
| CALL_FEES            |          | NUMBER (6,2)   |
| REG_NAME             |          | VARCHAR2 (128) |
| EDUL_NAME            |          | VARCHAR2 (128) |
| PROF_DESC            |          | VARCHAR2 (128) |
| CNTRY_NAME           |          | VARCHAR2 (128) |
| PKGT_DESC            |          | VARCHAR2 (128) |
| REP_GROUP            |          | VARCHAR2 (8)   |
| OS_USER              |          | VARCHAR2 (32)  |
| HOST_NAME            |          | VARCHAR2 (32)  |
| OPERATION_TYPE       |          | VARCHAR2 (16)  |
| OPERATION_TIMESTAMP  |          | TIMESTAMP (6)  |

```

EDI>

```

DBMS\_REDEFINITION PL/SQL API enables online table re-organization, such as migration of non-partitioned table to partitioned table as illustrated below.

- Check the table CALL\_DETAILS can be re-defined, if the call successfully completed, then the table is re-definable.

```

EDI> begin
2  dbms_redefinition.can_redef_table (
3  'OSM_REPAPI', 'CALL_DETAILS');

```

```

4  end;
5  /

PL/SQL procedure successfully completed.

EDI>

```

- Create the interim table – CALL\_DETAILS2, which has the new structure to enable partitioned table.

```

EDI> create table call_details2 (
2      CALL_ID          varchar2(64) primary key,
3      CALL_TIMESTAMP    timestamp not null,
4      CALL_MINUTE_DURATION number(5,2) not null,
5      CUST_NO           number,
6      CUST_NAME         varchar2(128),
7      SIM_CODE1        varchar2(32),
8      SIM_EXPIRE_DATE   date,
9      SIM_CELL_NO       varchar2(16),
10     DAILED_TO_AREA_CODE varchar2(8),
11     DAILED_TO_NO       varchar2(16),
12     DAILED_TO_NAME     varchar2(128),
13     CALL_FEES          number(6,2),
14     REG_NAME           varchar2(128),
15     EDUL_NAME          varchar2(128),
16     PROF_DESC          varchar2(128),
17     CNTRY_NAME         varchar2(128),
18     PKGT_DESC          varchar2(128),
19     REP_GROUP          varchar2(8),
20     OS_USER            varchar2(32),
21     HOST_NAME          varchar2(32),
22     OPERATION_TYPE     varchar2(16),
23     OPERATION_TIMESTAMP timestamp)
24 partition by range(CALL_TIMESTAMP)
25 (partition p2010 values less than (to_date('01-1-2011', 'DD-MM-YYYY')),
26  partition p2011 values less than (to_date('01-1-2012', 'DD-MM-YYYY')),
27  partition p2012 values less than (to_date('01-1-2013', 'DD-MM-YYYY')),
28  partition p2013 values less than (to_date('01-1-2014', 'DD-MM-YYYY')),
29  partition p2014 values less than (to_date('01-1-2015', 'DD-MM-YYYY')),
30  partition p2015 values less than (to_date('01-1-2016', 'DD-MM-YYYY')),
31  partition p2016 values less than (to_date('01-1-2017', 'DD-MM-YYYY')),
32  partition p2017 values less than (to_date('01-1-2018', 'DD-MM-YYYY')),
33  partition p2018 values less than (to_date('01-1-2019', 'DD-MM-YYYY')),
34  partition p2019 values less than (MAXVALUE));

Table created.

EDI>

```

- Start the redefinition process

```

EDI> begin
2      dbms_redefinition.start_redef_table (
3      'OSM_REPAPI', 'CALL_DETAILS', 'CALL_DETAILS2');
4  end;
5  /

```

```
PL/SQL procedure successfully completed.
EDI>
```

- Synchronized the new table with data. This may take time when the table has large number of rows.

```
EDI> begin
2   dbms_redefinition.sync_interim_table(
3   'OSM_REPAPI', 'CALL_DETAILS', 'CALL_DETAILS2');
4 end;
5 /

PL/SQL procedure successfully completed.
EDI>
```

```
EDI> select count(*) from call_details2;

COUNT(*)
-----
4523156

EDI>
```

- Copy dependent objects.

```
EDI> declare
2   x binary_integer;
3   begin
4   dbms_redefinition.copy_table_dependents(
5   uname=>'OSM_REPAPI',
6   orig_table=>'CALL_DETAILS',
7   int_table=>'CALL_DETAILS2',
8   copy_indexes=>1,
9   num_errors=>x);
10 end;
11 /
declare
*
ERROR at line 1:
ORA-01442: column to be modified to NOT NULL is already NOT NULL
ORA-06512: at "SYS.DBMS_REDEFINITION", line 2074
ORA-06512: at "SYS.DBMS_REDEFINITION", line 2074
ORA-06512: at "SYS.DBMS_REDEFINITION", line 1608
ORA-06512: at "SYS.DBMS_REDEFINITION", line 1487
ORA-06512: at "SYS.DBMS_REDEFINITION", line 408
ORA-06512: at "SYS.DBMS_REDEFINITION", line 1480
ORA-06512: at "SYS.DBMS_REDEFINITION", line 2056
ORA-06512: at "SYS.DBMS_REDEFINITION", line 3115
ORA-06512: at "SYS.DBMS_REDEFINITION", line 5578
ORA-06512: at line 4
```

```
EDI>
```

- Complete the redefinition

```
EDI> begin
2   dbms_redefinition.finish_redef_table(
3       'OSM_REPAPI', 'CALL_DETAILS', 'CALL_DETAILS2');
4 end;
5 /
```

PL/SQL procedure successfully completed.

```
EDI>
```

- Verify table is partitioned

```
EDI> select PARTITIONED
2   from user_tables
3  where table_name='CALL_DETAILS';
```

PAR

---

YES

```
EDI>
```

- Drop the interim table

```
EDI> drop table call_details2;
```

Table dropped.

```
EDI>
```

## Implement In-Memory Database Option

The implementation of the database in-memory option enables BI application to run from the memory.

```
EDI> alter system set inmemory_size=8G scope=spfile;
```

System altered.

```
EDI> startup force
ORACLE instance started.
```

|                          |                  |
|--------------------------|------------------|
| Total System Global Area | 1.7180E+10 bytes |
| Fixed Size               | 8641256 bytes    |
| Variable Size            | 5570037016 bytes |
| Database Buffers         | 838860800 bytes  |
| Redo Buffers             | 24911872 bytes   |

```
In-Memory Area          1.0737E+10 bytes
Database mounted.
Database opened.
EDI>
```

## Implement Advanced Compression for DSS

Advanced compression compress space and enhances the query performance. Enable compression on table CALLS\_DETAILS.

```
EDI> alter table osm_repapi.call_details
2   compress;

Table altered.

EDI>
```

## Deploy Materialized View for Query Re-Write

By deploying materialized views, query performance drastically enhanced. Instead of retrieving data from a relational table, the data is retrieved from the result of the query stored on disk or memory.

The original query shows TABLE ACCESS FULL, after deploying the materialized view, the result of the query is retrieved from the materialized view, which reduce the execution steps, hence better performance achieved.

```
EDI> alter system flush shared_pool;

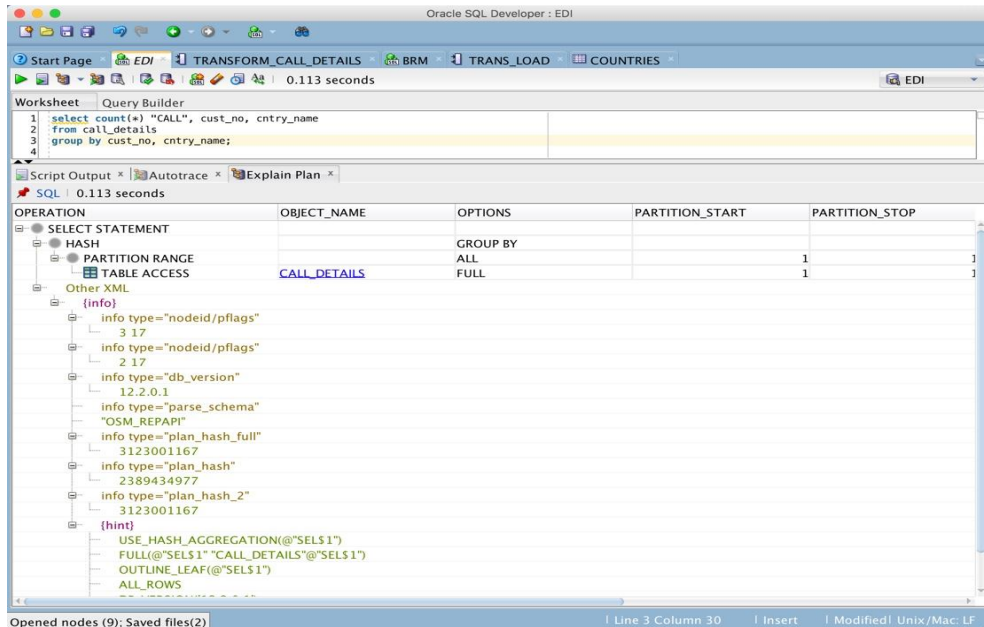
System altered.

EDI> alter system flush buffer_cache;

System altered.

EDI>
```

```
EDI> select count(*) "CALL", cust_no, cntry_name
2   from   call_details
3  group by cust_no, cntry_name
4  /
```



The execution plan shows a TABLE ACCESS operation on PARTITION RANGE.

Create the materialized view Log on tables referenced by CALL\_DETAILS\_BY\_CUST\_CNTRY.

```

EDI> create materialized view log on call_details
2   with rowid, sequence (cust_no, cntry_name)
3   including new values;

```

Materialized view log created.

EDI>

Create the materialized view CALL\_DETAILS\_BY\_CUST\_CNTRY.

```

EDI> create materialized view call_details_by_cust_cntry
2   REFRESH FAST ON DEMAND
3   ENABLE QUERY REWRITE
4   ENABLE ON QUERY COMPUTATION
5   AS
6   select count(*) "CALL", cust_no, cntry_name
7   from   call_details
8   group by cust_no, cntry_name
9   /

```

Materialized view created.

EDI>

Gather data dictionary statistics

```
EDI> begin
2   dbms_stats.gather_table_stats(
3     ownname=>'OSM_REPAPI',
4     tabname=>'call_details_by_cust_cntry');
5 end;
6 /

PL/SQL procedure successfully completed.

EDI>
```

```
EDI> alter system flush shared_pool;

System altered.

EDI> alter system flush buffer_cache;

System altered.

EDI>
```

Execute the query and observe the execution plan.

The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for Start Page, EDI, TRANSFORM\_CALL\_DETAILS, BRM, TRANS\_LOAD, and COUNTRIES. The main window displays a worksheet with a query: `select count(*) "CALL", cust_no, cntry_name from call_details group by cust_no, cntry_name;`. Below the query, the 'Script Output' tab shows the execution plan. The plan is a materialized view rewrite (MAT\_VIEW REWRITE ACCESS) on the table 'CALL\_DETAILS\_BY\_CUST\_CNTRY'. The execution plan details are as follows:

| OPERATION               | OBJECT_NAME                | OPTIONS | CARDINALITY | COST |
|-------------------------|----------------------------|---------|-------------|------|
| SELECT STATEMENT        |                            |         |             | 9    |
| MAT_VIEW REWRITE ACCESS | CALL_DETAILS_BY_CUST_CNTRY | FULL    |             | 9    |

The plan also includes an 'Other XML' section with the following details:

- info type="db\_version": 12.2.0.1
- info type="parse\_schema": "OSM\_REPAPI"
- info type="plan\_hash\_full": 653830240
- info type="plan\_hash": 2183543123
- info type="plan\_hash\_2": 653830240

The plan also includes a hint section with the following details:

- hint: FULL(@SEL\$117BF4B3"CALL\_DETAILS\_BY\_CUST\_CNTRY"@SEL\$F99C6F56)
- OUTLINE(@SEL\$1)
- REWRITE(@SEL\$1"CALL\_DETAILS\_BY\_CUST\_CNTRY")
- OUTLINE(@SEL\$DDD2B3FC)
- REWRITE(@SEL\$DDD2B3FC"CALL\_DETAILS\_BY\_CUST\_CNTRY")
- OUTLINE\_LEAF(@SEL\$117BF4B3)
- ALL\_ROWS
- DB\_VERSION("12.2.0.1")
- OPTIMIZER\_FEATURES\_ENABLE("12.2.0.1")
- IGNORE\_OPTIM\_EMBEDDED\_HINTS

## Deploy In-Memory Materialized View for Query Re-Write

This further enhances the queries performance, enabling Real-Time BI Applications.



```

EDI> alter materialized view osm_repapi.call_details_by_cust_cntry
2   inmemory
3   memcompress for capacity high priority high;

Materialized view altered.

EDI> exec dbms_mview.refresh('osm_repapi.call_details_by_cust_cntry');

PL/SQL procedure successfully completed.

EDI>

```

Re-run the query and observe the execution plan. Notice the Query elapsed execution time is 0 seconds.

The screenshot shows the Oracle SQL Developer interface. The top pane displays a query in the Query Builder:

```

1 select count(*) "CALL", cust_no, cntry_name
2 from call_details
3 group by cust_no, cntry_name;

```

The bottom pane shows the Explain Plan for the query. The execution plan is as follows:

| OPERATION               | OBJECT_NAME               | OPTIONS       | CARDINALITY | COST |
|-------------------------|---------------------------|---------------|-------------|------|
| SELECT STATEMENT        |                           |               |             | 6246 |
| MAT_VIEW REWRITE ACCESS | CALL_DETAILS_BY_CUST_C... | INMEMORY FULL |             | 6246 |

The plan also includes an "Other XML" section with the following details:

- info type="db\_version": 12.2.0.1
- info type="parse\_schema": "OSM\_REPAPI"
- info type="plan\_hash\_full": 653830240
- info type="plan\_hash": 653830240
- info type="plan\_hash\_2": 653830240
- (hint) FULL(@SEL\$117BF4B3 "CALL\_DETAILS\_BY\_CUST\_CNTRY"@SEL\$F99C6F56)
- OUTLINE(@SEL\$1)
- REWRITE(@SEL\$1 "CALL\_DETAILS\_BY\_CUST\_CNTRY")
- OUTLINE(@SEL\$DDD2B3FC)
- REWRITE(@SEL\$DDD2B3FC "CALL\_DETAILS\_BY\_CUST\_CNTRY")
- OUTLINE\_LEAF(@SEL\$117BF4B3)
- ALL\_ROWS
- DB\_VERSION("12.2.0.1")
- OPTIMIZER\_FEATURES\_ENABLE("12.2.0.1")
- IGNORE\_OPTIM\_EMBEDDED\_HINTS

The status bar at the bottom indicates "Opened nodes (9); Saved files(2)".

## Implement Parallel Queries Execution

For specific load process, parallel execution speed up the execution of long-running queries.

```

EDI> commit;

Commit complete.

```

```
EDI> alter session force parallel DML
      2 parallel 8;

Session altered.

EDI> alter session force parallel query
      2 parallel 8;

Session altered.

EDI>
```

## Run the Analytical Applications on BRM

Run the query, evaluate the execution plan.

```
BRM> select count(*), cust_no, cont_mobile_no
2   from   contacts
3   group by cust_no, cont_mobile_no;
```

```
BRM> alter system flush shared_pool;
```

System altered.

```
BRM> alter system flush buffer_cache;
```

System altered.

```
BRM>
```

The screenshot shows the Oracle SQL Developer interface. The top pane displays the query: `select count(*), cust_no, cont_mobile_no from contacts group by cust_no, cont_mobile_no;`. The bottom pane shows the execution plan for the query, which is a HASH GROUP BY operation. The plan details are as follows:

| OPERATION        | OBJECT_NAME | OPTIONS  | CARDINALITY | COST  |
|------------------|-------------|----------|-------------|-------|
| SELECT STATEMENT |             |          |             | 97294 |
| HASH             |             | GROUP BY |             | 726   |
| TABLE ACCESS     | CONTACTS    | FULL     |             | 726   |
| Other XML        |             |          |             | 205   |

The plan also includes a hint section with the following details:

- info type="db\_version": 11.2.0.3
- info type="parse\_schema": "OSM\_REPAI"
- info type="plan\_hash": 2505073297
- info type="plan\_hash\_2": 528648520
- hint: USE\_HASH\_AGGREGATION(@"SEL\$1")
- hint: FULL(@"SEL\$1" "CONTACTS"@"SEL\$1")
- hint: OUTLINE\_LEAF(@"SEL\$1")
- hint: ALL\_ROWS
- hint: DB\_VERSION("11.2.0.3")
- hint: OPTIMIZER\_FEATURES\_ENABLE("11.2.0.3")
- hint: IGNORE\_OPTIM\_EMBEDDED\_HINTS

## Run the Analytical Applications on EDI

Run the query, evaluate the execution plan. Notice the execution plan indicates the query performed using the following features:

- Parallel Query Execution
- In-Memory Materialized view
- Advanced Compression

Let's begin without advanced configuration

The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for Start Page, EDI, TRANSFORM\_CALL\_DETAILS, BRM, TRANS\_LOAD, and COUNTRIES. The main window displays a SQL query in the Worksheet:

```
1 select count(*), cust_no, cont_mobile_no
2 from contacts
3 group by cust_no, cont_mobile_no;
4
```

Below the query, the 'Script Output' tab shows the execution plan for the SQL statement. The plan is as follows:

| OPERATION        | OBJECT_NAME | OPTIONS   | CARDINALITY | COST    |
|------------------|-------------|-----------|-------------|---------|
| SELECT STATEMENT |             |           |             | 1976882 |
| HASH             |             | GROUP BY  |             | 13368   |
| TABLE ACCESS     | CONTACTS    | COST=2864 |             | 2864    |

The 'Other XML' section of the plan shows the following details:

- info type="db\_version": 12.2.0.1
- info type="parse\_schema": "OSM\_REPAPI"
- info type="plan\_hash\_full": 528648520
- info type="plan\_hash": 2505073297
- info type="plan\_hash\_2": 528648520
- hint:
  - USE\_HASH\_AGGREGATION(@"SEL\$1")
  - FULL(@"SEL\$1" "CONTACTS"@"SEL\$1")
  - OUTLINE\_LEAF(@"SEL\$1")
  - ALL\_ROWS
  - DB\_VERSION("12.2.0.1")
  - OPTIMIZER\_FEATURES\_ENABLE("12.2.0.1")
  - IGNORE\_OPTIM\_EMBEDDED\_HINTS

The bottom status bar indicates 'Opened nodes (9); Saved files(2)' and 'Line 3 Column 34 | Insert | Modified | Unix/Mac: LF'.

Now, enable advanced configuration

Create the materialized view

```
EDI> create materialized view log on contacts
2 with rowid, sequence (cust_no, cont_mobile_no)
3 including new values;
```

Materialized view log created.

EDI>

```
EDI> create materialized view CountContactMV
2 REFRESH FAST ON DEMAND
3 ENABLE QUERY REWRITE
```

```

4  ENABLE ON QUERY COMPUTATION
5  AS
6  select count(*), cust_no, cont_mobile_no
7  from    contacts
8  group by cust_no, cont_mobile_no;

```

Materialized view created.

EDI>

Gather data dictionary statistics

```

EDI> begin
2    dbms_stats.gather_table_stats(
3        ownname=>'OSM_REPAPI',
4        tabname=>'CountContactMV');
5  end;
6  /

```

PL/SQL procedure successfully completed.

EDI>

Enable in-memory feature

```

EDI> alter materialized view osm_repapi.CountContactMV
2  inmemory
3  memcompress for capacity high priority high;

```

Materialized view altered.

EDI>

Refresh the materialized view, this may take time.

```

EDI> exec dbms_mview.refresh('osm_repapi.CountContactMV');

```

PL/SQL procedure successfully completed.

EDI>

```

EDI> alter system flush shared_pool;

```

System altered.

```

EDI> alter system flush buffer_cache;

```

System altered.

EDI>

Re-run the query

Oracle SQL Developer : EDI

Start Page | EDI | TRANSFORM\_CALL\_DETAILS | BRM | TRANS\_LOAD | COUNTRIES

0 seconds

Worksheet | Query Builder

```

1 select count(*), cust_no, cont_mobile_no
2 from contacts
3 group by cust_no, cont_mobile_no;
4

```

Script Output | Autotrace | Explain Plan

SQL | 0 seconds

| OPERATION               | OBJECT_NAME    | OPTIONS       | CARDINALITY | COST |
|-------------------------|----------------|---------------|-------------|------|
| SELECT STATEMENT        |                |               | 17213477    | 730  |
| MAT_VIEW REWRITE ACCESS | COUNTCONTACTMV | INMEMORY FULL | 17213477    | 730  |

Other XML

{info}

- info type="db\_version"
  - 12.2.0.1
    - info type="parse\_schema"
      - "OSM\_REPAPI"
        - info type="plan\_hash\_full"
          - 2126189475
        - info type="plan\_hash"
          - 652088532
        - info type="plan\_hash\_2"
          - 2126189475

{hint}

```

FULL(@"SEL$B067469B" "COUNTCONTACTMV"@"SEL$324F16CD")
OUTLINE(@"SEL$1")
REWRITE(@"SEL$1" "COUNTCONTACTMV")
OUTLINE(@"SEL$FAE885B7")
REWRITE(@"SEL$FAE885B7" "COUNTCONTACTMV")
OUTLINE_LEAF(@"SEL$B067469B")
ALL_ROWS
DB_VERSION('12.2.0.1')
OPTIMIZER_FEATURES_ENABLE('12.2.0.1')
IGNORE_OPTIM_EMBEDDED_HINTS

```

Opened nodes (9); Saved files(2)

| Line 3 Column 34 | Insert | Modified | Unix/Mac: LF

## Compare the Runtime Results

The result is amazing; the execution ratio is by far faster on the EDI instance than on the BRM instance.

The BRM instance execution plan shows execution time of:  
**2.612 Seconds.**

Oracle SQL Developer - BRM

Worksheet: Query Builder

```
1 select count(+), cust_no, cont_mobile_no
2 from contacts
3 group by cust_no, cont_mobile_no
```

Explain Plan

SQL: 2.612 seconds

| OPERATION        | OBJECT_NAME | OPTIONS  | CARDINALITY | COST |
|------------------|-------------|----------|-------------|------|
| SELECT STATEMENT |             |          |             | 726  |
| HASH             |             | GROUP BY | 97294       | 726  |
| TABLE ACCESS     | CONTACTS    | FULL     | 97294       | 205  |

Other XML

info type="db\_version"

11.2.0.3

info type="parse\_schema"

"OSM\_REPAIR"

info type="plan\_hash"

2905073297

info type="plan\_hash\_2"

528648520

COST=205

USE\_HASH\_AGGREGATION(@"SEL\$1")

FULL(@"SEL\$1" "CONTACTS"@"SEL\$1")

OUTLINE\_LEAF(@"SEL\$1")

ALL\_ROWS

DB\_VERSION(11.2.0.3)

OPTIMIZER\_FEATURES\_ENABLE(11.2.0.3)

IGNORE\_OPTIM\_EMBEDDED\_HINTS

The EDI instance execution plan shows execution time of:  
**0 Seconds.**

Oracle SQL Developer - EDI

Worksheet: Query Builder

```
1 select count(+), cust_no, cont_mobile_no
2 from contacts
3 group by cust_no, cont_mobile_no
```

Script Output

SQL: 0 seconds

| OPERATION               | OBJECT_NAME    | OPTIONS       | CARDINALITY | COST |
|-------------------------|----------------|---------------|-------------|------|
| SELECT STATEMENT        |                |               |             | 730  |
| MAT_VIEW REWRITE ACCESS | COUNTCONTACTMV | INMEMORY FULL | 17213477    | 730  |

Other XML

info type="db\_version"

12.2.0.1

info type="parse\_schema"

"OSM\_REPAIR"

info type="plan\_hash\_full"

2126189475

info type="plan\_hash"

652085132

info type="plan\_hash\_2"

2126189475

COST=730

FULL(@"SEL\$B0674698" "COUNTCONTACTMV"@"SEL\$324F16CD")

OUTLINE(@"SEL\$1")

REWRITE(@"SEL\$1" "COUNTCONTACTMV")

OUTLINE(@"SEL\$FAE88587")

REWRITE(@"SEL\$FAE88587" "COUNTCONTACTMV")

OUTLINE\_LEAF(@"SEL\$B0674698")

ALL\_ROWS

DB\_VERSION(12.2.0.1)

OPTIMIZER\_FEATURES\_ENABLE(12.2.0.1)

IGNORE\_OPTIM\_EMBEDDED\_HINTS

# Summary

The major enhancements of Oracle Database 12c Release 2 is handling data warehouse challenges. The following lists of Oracle Database 12c R2 features and options are the building blocks of an Enterprise Data Warehouse (EDW). Without deploying these technologies it's likely the Enterprise Data Warehouse (EDW) is unable to deliver the scalability and speed required by enterprise analytical applications.

- Database parameters settings
- Big tablespace implementation
- Large database I/O block
- Temporary groups
- Database partitioning
- In-Memory option
- Advanced Compression
- Materialized views

The foundation of an Enterprise Data Warehouse (EDW) repository is a database running on platform supporting large block disk I/Os, without it the EDW likely to encounter low throughput, hence limited scalability.

The four rules of them for the Enterprise Data Warehouse (EDW) are: *Partitioning, In-Memory database, advanced compression and materialized view for query re-write.*

The database partitioning divides a large table into multiple physical segments. This introduces the perfect management window; administration is carried out at the individual tablespace of the segment.

The in-memory database is the fastest way to execute analytical applications with in-depth summaries and aggregations. After sizing and allocating the SGA memory for in-memory options, real-time data access is enabled for volatile tables and materialized views.

Advanced compression, no doubt has two advantages: Speedup the applications and reduces storage cost. EDW running on EXADATA benefits from Hybrid Columnar Compression (HCC) transparently.

Last but not least, materialized view implementation is the nerves of an EDW. It enables the optimizer for query re-write.

*The next chapter introduces Oracle GoldenGate 12c for Big Data, Kafka handler.*



# From the Enterprise Data Warehouse to Kafka using Oracle GoldenGate 12c for Big Data

## Overview

Enterprises always think of big data as a platform, for one reason; Total Cost of Ownership, or TCO. It's all about promoting data processing and reducing storage cost. The transition from the operational data store (ODS), to the enterprise data warehouse to the big data is an ultimate direction. An enterprise demanding real-time data for the decision making must rely on real-time data feed and processing. This chapter extends the capabilities of the Enterprise Data Warehouse to deliver an enterprise real-time big data using:

- Oracle GoldenGate 12c for Big Data
- Apache Kafka

## Oracle GoldenGate 12c for Big Data

It's an extension for Oracle GoldenGate 12c for streaming real-time transitional data into big data technologies. Currently Oracle GoldenGate for big data has support for:

- Hadoop Distributed File System (HDFS)
- Apache Hive
- Apache HBASE
- Apache Flume
- Apache Kafka
- MongoDB
- Oracle NOSQL DATABASE
- ElasticSearch

## Apache Kafka

Kafka is a streaming platform for real-time stream applications. Connectors allow the interaction with Kafka to export data from Kafka and import data to Kafka.

# Architecture

The concept of enterprise architecture, captures once from the core systems, and delivers across the enterprise, and in real-time, that is the key competitive feature for Oracle GoldenGate and Oracle GoldenGate for big data. Figure 1 depicts through multiple data pump processes, feeding the Enterprise Data Warehouse (EDW) and Kafka occurs in parallel.

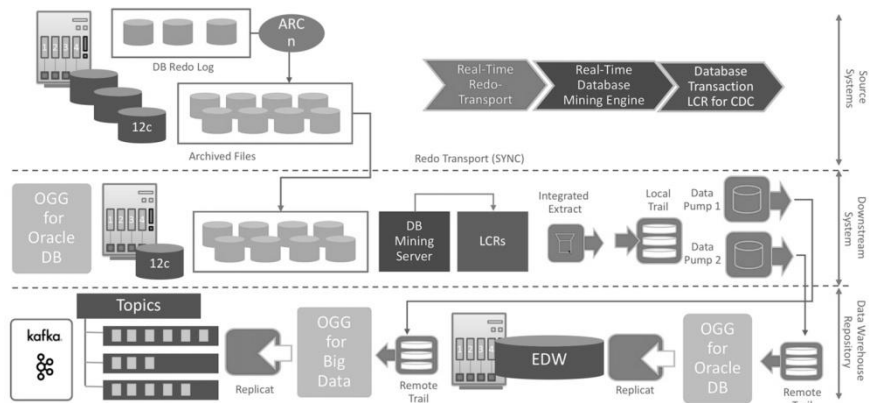


Figure 1: Feeding the Enterprise Data Warehouse and Kafka in Parallel

The core system ships the transaction logs through the redo-transport service. The redo-transport received by the downstream system mining database, which produces the logical change records (LCRs). The LCRs processed using a sequence of database mining phases, but it represents the transactions in strict sequence, which is the input to the next step, the change data capture by the integrated extract.

An integrated extract perform change data capture and build the trail. Now, it's time spinoff as many data pump processes as needed, in the case shown in Figure 1, there will be two data pump processes, the first data pump writing to EDW remote trail and the second data pump write to Kafka remote trail.

The high-performance Replicat applies transactions to EDW database, and Oracle GoldenGate 12c for big data Replicat ingests the transaction to Kafka topics. The Oracle GoldenGate 12c for Big Data uses native Java Replicat configured using parameter file, adapter file and JAR file as will be shown next.

Though for the purpose of real-time data streaming into big data Kafka was selected, however; Oracle GoldenGate 12c for big data supports ingestion through the use of adapters to HDFS, Hive, HBASE, Flume, Mongo, NOSQL Database and Elastic.

# Install Kafka

The installation of Kafka is very basic. Its simply unzipping the distribution media into a directory, referred to as Kafka software location. Download the Kafka from <https://kafka.apache.org> for Linux and follow the step shown.

#

## Start Kafka Server

Verify the installation is successful by start the Kafka server.

#

# Kafka Architecture

T

Figure 2 shows Kafka components.

*Figure 2: Kafka Components*

## Working with Kafka

This section covers the basics of working with Kafka.

- Creating Kafka topic
- Send messages

- Starting a consumer
- Setting up a multi-broker cluster
- Use Kafka connect to import/export data
- Use Kafka Streams to process data

## **Create Kafka Topics**

T

## **Sending Messages**

T

## **Starting a Consumer**

T

## **Setting up a Multi-Broker Cluster**

T

## **Use Kafka to Connect to Export/Import Data**

T

## **Use Kafka Streams to Process Data**

T

## **Install Oracle GoldenGate 12c for Big Data**

T

## **Configuring Oracle GoldenGate Data Pump to Kafka**

T

# Configuring Oracle GoldenGate Replicat for Kafka

T

## Streaming Data to Kafka

Run the workload simulator on the source database. The data pump

## Exporting Data from Kafka

T

# Summary

Big Data is an ultimate data store for the enterprise. Open source big data such as Kafka enables high-performance platform for streaming applications. One method of ingestion of transactional data into big data Kafka is using Oracle GoldenGate 12c for big data, it enables real-time data ingestion.

In addition to Apache Kafka, Oracle GoldenGate for big data has support for:

- Hadoop Distributed File System (HDFS)
- Apache Hive
- Apache HBASE
- Apache Flume
- MongoDB
- Oracle NOSQL DATABASE
- ElasticSearch

Visit <https://kafka.apache.org> for downloading the latest release of Apache Kafka for Linux. The installation of Apache Kafka on Linux is simply unzipping the distribution file on the Apache Kafka software location.

Oracle GoldenGate 12c for big data uses the familiar configuration file and GGSCI definition command as Oracle GoldenGate 12c, but with the addition of adapter properties file. The Replicat is Native Java Replicat or Oracle GoldenGate 12c Java Adapter.

After Kafka server starts, the chapter has demonstrated the below, to ensure Kafka is successfully operational.

- Creating Kafka topic
- Send messages
- Starting a consumer
- Setting up a multi-broker cluster
- Use Kafka connect to import/export data
- Use Kafka Streams to process data

The last steps is to deploy Oracle GoldenGate 12c for Big Data Kafka handler, and then export data from Kafka to an Oracle Database 12c to support online applications developed using Oracle Data Visualization Desktop Edition.

*That is the end of the implementation. Now, real-time data is streaming from the transactional applications, to the Enterprise Data Warehouse (EDW) to Big Data Kafka.*

# Oracle GoldenGate Fundamentals

# A

## Overview

The objective of this chapter is to cover Oracle GoldenGate fundamentals, focusing on use cases and applications, concept and architecture, topologies and designs, components and building blocks and end-to-end illustrative setup for a headstart. For Oracle GoldenGate knowledgeable users, the chapter intended for refreshing, for users new to Oracle GoldenGate, it builds the foundation to continue the remaining chapters. Despite the fact that the illustrative setup and commands are developed on Oracle GoldenGate 12c, they should apply to Oracle GoldenGate supported releases, unless noted otherwise.

The chapter discusses and exemplify the following topics:

- Oracle database replication evolution and history
- Supported platforms and product set
- Enterprise applications and use cases
- Network topologies and designs
- Building blocks, architecture and processes workflow
- Command-line utilities and obey files
- Parameter files and macros
- Manager process
- Primary and secondary extracts groups for change data capture
- Change data delivery group
- Trails and trail file format
- Commit change number, checkpoint file and checkpoint table
- The data server collector
- Intermediate systems and integrated capture
- Data selection, filtering, transformation and column mapping
- Data definition file utility
- Built-in functions and tokens
- Call-out to database stored procedures and C programs
- Post deployment configuration options

- Configure primary extract for Automatic storage environment (ASM)
- Implementing high-performance parallel processing
- Generate reports and lagging
- Configure extract and replicate processes for pluggable database environment
- Demonstrating three-step configuration

The aim of the book remains, an in-depth dive into Oracle GoldenGate 12c applications for area-specific solutions presented on subsequent chapters. Even though Oracle GoldenGate concepts applies across supported platforms, the fundamentals chapter discusses Oracle GoldenGate 12c for Oracle Database 11g, Oracle Database 12c and Microsoft SQL Server 2008.

For training purposes only, an Oracle VirtualBox and VM player machines are made available, the machines provide a quick setup to practice Oracle GoldenGate 12c. Visit <http://www.oracle-ggs.com> for downloading updated virtual machines and user's guide for setup instructions. Terms and conditions applied.

## Oracle GoldenGate, an enterprise software

For years, Oracle has been one of the world largest enterprise software company, now Oracle software found globally at fortune companies running core business applications. These companies continue to have real demands for solutions to optimize, consolidate and integrate business functions across the enterprise by running heterogeneous core business applications systems. Oracle GoldenGate is part of Oracle data integration solution (DIS) portfolio, when deployed as best practices, it becomes the platform for data consolidation and integration delivering best performance, flexible architecture, reliability and extendibility across the enterprise.

As Oracle continues the research and development for the world leading object relational database management system (ORDBMS), enterprise applications demand more integration among database platforms, and when it's for the enterprise, the integration must be *heterogeneous*, and that is a key differentiator to stand out and stay competitive within the data integration market. Oracle GoldenGate optimized to run on popular database management systems such as Oracle Database, Oracle MySQL, Microsoft SQL Server, IBM DB2, etc.

Enterprise applications require moving transactional data in real-time among systems hosted globally, it has been traditionally processed using in-house developed applications utilizing database links, triggers and materialized views among the enterprise databases, and when it's for the enterprise, data growth maintains a linear pattern and mostly requires online data movement with zero refresh window timeframe. Another key differentiator, Oracle GoldenGate enables real-time data



movement with *sub-second latency* using an optimized change data capture and delivery processes, while employing basic-to-complex data transformation and consolidation for data in transit.

Performance and data consistency cannot be compromised at the enterprise level. Oracle GoldenGate uses its proprietary technique to maintain data consistency across systems regardless of databases type. Using an in-memory processing, Oracle GoldenGate treats a transaction as a complete logical unit of work, when the transaction committed, it's assigned an Oracle GoldenGate commit sequence number (CSN) or discarded when the transaction is rolled back. Committed transactions are moved to the next downstream process. At the target system, the transactions reassembled and applied with the same capture order for maintaining *transaction integrity*. By having Oracle GoldenGate captures committed transactions only, this has a substantial performance enhancement on high-volume transactional systems.

And, that explains Oracle GoldenGate in terms of *heterogeneous* support, *sub-second latency* for data movement and transactional data *integrity*, making it an enterprise software for powering critical mission applications.

Enterprises running critical mission applications requires well engineered software, promoting flexible and extendable architecture, and this is what is Oracle GoldenGate is all about. Implementing Oracle GoldenGate is site specific, based on that site business and technical requirements, a flexible dynamic architecture is developed and deployed. Based on the network topology, capture and delivery groups may be designed loosely or tightly coupled, for efficient resource utilization with varying degree of parallelism and custom development for unlimited extendibility.

Because Oracle GoldenGate uses its own checkpoint sub-system, it enabled an optimized restart due to any type of failures. Extract and replicate processes continuously checkpoint their read and write positions. Data security regulation mandates data availability at all times, Oracle customers have been using Oracle Data Guard to meet data security requirements for disaster recovery, Oracle GoldenGate extract groups support capture from standby archive log file, this integration seamlessly promotes data movement from the standby database leaving the source production database dedicated for online transactions, an added advantage for report offloading instance from a standby database environment.

## Oracle data replication evolution

Data replication has been always common requirement for database applications. Transparent data replication has been part of Oracle database framework. Hence, Oracle database replication developed using PL/SQL database application

programmatic interface (API) and enabled from within PL/SQL or Oracle Enterprise Manager (GUI). For high-volume transactions, handling data replication outside of the database contributes to achieve real-time data replication across heterogeneous systems as its engineered by Oracle GoldenGate.

The following describes the history of Oracle database replication from inception to current, that is from Oracle 7 to Oracle GoldenGate 12c.

## **Oracle replication, June 1992**

Oracle replication introduced with Oracle 7.0. It was simple infrastructure, database triggers are associated with database tables, moving data to remote database using database links. Subsequently, the feature has been completely redeveloped as Oracle advanced replication.

## **Oracle advanced replication, May 1994**

Oracle 7.1 introduced Oracle advanced replication. A complete data replication framework for creating complex database replication network, such as master-to-master replication. Using PL/SQL API or Oracle Enterprise Manager (GUI), data replication API infrastructure generated and enabled for data replication. Basic predefined data conflict resolutions defined and handled internally.

## **Oracle Streams, May 2002**

Oracle Database 9.2 introduced Oracle Streams. Streams uses advance queuing technology for data capture, propagate and apply workflow. Oracle Streams supports creating complex data replication network, such as master-to-master by using PL/SQL API or Oracle enterprise manager (GUI). Oracle Streams has superseded Oracle advanced replication.

## **Oracle GoldenGate, September 2009**

It's data replication and beyond. Oracle GoldenGate enables high availability solutions, real-time data integration, transactional change data capture (CDC) and delivery, high-performance data replication and complex data transformations. Oracle GoldenGate engineered for high-performance real-time heterogeneous data replication, supporting the leading database management systems with three key differentiators: heterogeneous, sub-second latency and data integrity.

Oracle strategy for data replication solutions stated Oracle database Streams is superseded by Oracle GoldenGate as quoted on Oracle Statement of Direction, November 2012.

*“Oracle Streams continues to be supported but will not be enhanced to support any new database features in future releases. This means that Streams will continue to support Exactly what it supports today (11.2) in future releases. It also means that any new database features, data types, or capabilities will not be supported by Streams. Customers who want to take advantage of the new data types and features will need to license Oracle GoldenGate.” – Oracle Corporation*

With that said, the direction definitely strategic and positively reflects the continued research and development efforts for more comprehensive Oracle GoldenGate product set, and highly integrated with other related Oracle products and most of all Oracle Data Integrator (ODI) and Oracle Database 12c data guard technology.

## What is new for Oracle GoldenGate 12c

Oracle GoldenGate 12c version 12.1.2.0.0 was made available on October 2013, the following is a brief description of the new features and the enhancements introduced by Oracle GoldenGate 12c.

### Database support

Oracle GoldenGate 12c supports Oracle Database 12c, the database for cloud computing. Oracle GoldenGate continues supporting and enhancing existing non-Oracle database platforms, MS SQL Server, Teradata, mainframe and external adapters.

### Software installation

Oracle GoldenGate 12c is shipped with graphical user interface (GUI) installer, supporting Oracle Database 11g and Oracle Database 12c. The 5-step easy-to-use wizard de-supported the manual installation of previous releases. The GUI installer installs the software, creates the directories structure and starts the manager process. The software is bundled for Oracle Database 11g and Oracle Database 12c.

### Integrated capture

For users who want to isolate Oracle GoldenGate capture process from the production database system rely on integrated capture extracts. With integrated capture, Oracle GoldenGate 12c installed on a downstream system continuously

receives REDO log files from the source system. Oracle GoldenGate 12c has transparently enhanced the performance of integrated capture and replicate processes. Integrated capture will be the area of enhancements on future releases.

## **Oracle multitenant container database**

Oracle GoldenGate 12c supports capturing from and delivering to Oracle multitenant container database (CDB). A primary extract captures from a pluggable database (PDB), writes data to a local trail, and delivers to a remote pluggable database.

## **Extended TABLE and MAP statement**

Oracle GoldenGate 12c support three-part object name for the TABLE and MAP statement. The object name includes <container>.<schema>.<table name>.

## **Enhanced DDL support**

DDL replication does not need database triggers setup. Oracle GoldenGate 12c performs native DDL replication. This is the default mode, switching back to capture DDL mode requires Oracle GoldenGate 12c DDL setup.

## **Character set**

Oracle GoldenGate 12c enhanced the change data delivery by providing automatic character set support, eliminating setting NLS\_LANG parameter on the replicate process.

## **Initial-load data types**

Oracle GoldenGate 12c initial-load remote task supports all Oracle data types including BLOB, CLOB, NCLOB, LONG, UDT and XML. Making direct initial-load method an excellent technique.

## **MS SQL Server**

Oracle GoldenGate 12c supports compressed backup, in addition to the previously supported uncompressed backup. Compressed backup uses less I/O therefore increasing the backup speed.

# Oracle GoldenGate line of products

Oracle GoldenGate 12c bundles several Oracle software licensable separately. The following is a brief description of Oracle GoldenGate product set. A major software component of Oracle GoldenGate is adapters, used to support legacy systems and site-specific data processing requirements. Adapters mostly adapted by financial institutes using BASE24 system for payment processing. IBM mainframe is has been always supported by Oracle GoldenGate for enabling data movement to modern systems across the enterprise.

Oracle GoldenGate product set described next, starting with Oracle GoldenGate core for basic data capture and delivery.

## Oracle GoldenGate core

The core is made of Oracle GoldenGate capture and Oracle GoldenGate apply. Oracle GoldenGate core is classified as:

- Oracle GoldenGate for Oracle Database
- Oracle GoldenGate for non-Oracle database
- Oracle GoldenGate for mainframe

Oracle GoldenGate core enables real-time change data capture (CDC), moving high volumes of transactional data between databases with very low latency and data integrity using combinations of heterogeneous source and target databases. The separation of capture and apply Oracle GoldenGate core license enables faster return on investment (ROI) and cost-effective integration with non-Oracle data integration software.

An added advantage of Oracle GoldenGate core is support for Oracle Database standard edition, providing high availability solutions for non-enterprise users using Oracle database standard edition. Oracle GoldenGate core includes full use of XStream and Active Data Guard. XStream is an API for enabling client applications to receive and send real-time data, Active Data Guard enables the physical standby database to be opened for read-only while applying changes from the primary database. Both XStream and Active Data Guard require running Oracle database enterprise edition.

Two essential components of Oracle GoldenGate core are GoldenGate Software Command Interface (GGSCI) and the LOGDUMP utilities. Use GGSCI to define configure, create and deploy GoldenGate components from the command-line, use

LOGDUMP utility for reading, searching and managing trail files and troubleshooting. Refer to chapter 13 for details regarding LOGDUMP.

## Management pack for Oracle GoldenGate

The management pack composed of a set of Oracle GoldenGate products providing a graphical user interface (GUI) for managing Oracle GoldenGate using the latest Java and Web-base technologies.

### Oracle GoldenGate director

A graphical user interface (GUI) for defining, configuring, managing, monitoring and reporting on Oracle GoldenGate instances through Java client and web-based interfaces. Director provides a diagramming visual tool for creating diagrams for new or existing Oracle GoldenGate instances. Using drag-and-drop technique, the diagramming tool creates capture and delivery groups, initial load, capture, trial, files and replicates easily and instantly. Figure 1-1 shows the diagramming tool.

Refer to chapter 9 for details regarding Oracle GoldenGate director.

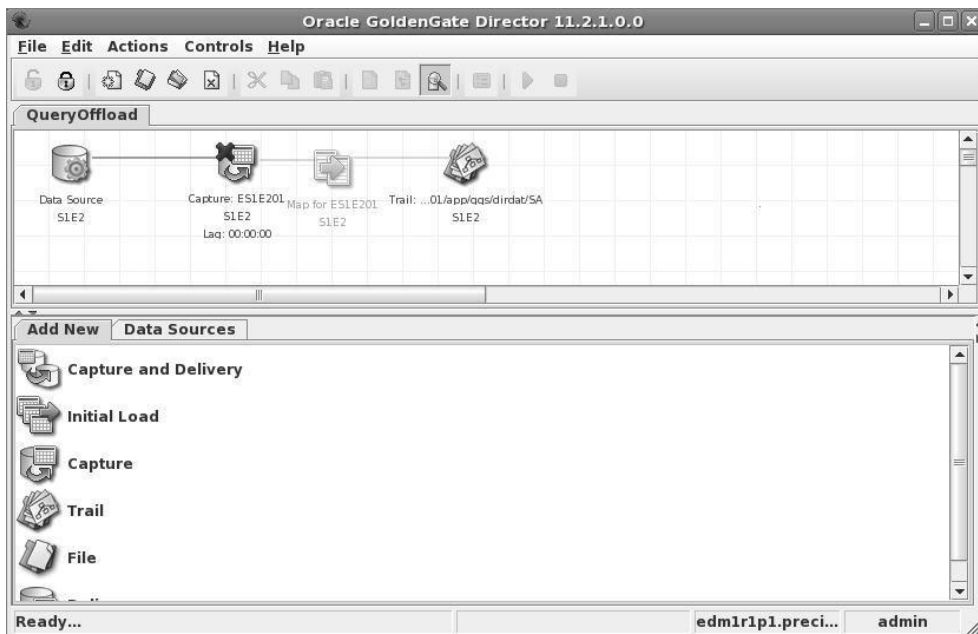


*Figure 1-1: Oracle GoldenGate director visual diagramming tool*

## Oracle GoldenGate monitor

Oracle GoldenGate monitor is a browser-based for viewing purpose only, it does not modify Oracle GoldenGate configuration files. It's used to monitor Oracle GoldenGate instance process status, operating and performance statistics, and reports runtime events. The major feature of Oracle GoldenGate monitor is external notification using either email or SNMP.

## Oracle GoldenGate Plug-in for Oracle Enterprise Manager

Oracle Enterprise Manager Cloud Control 12c management pack for Oracle GoldenGate enables managing Oracle GoldenGate instances from a centralized integrated console. Upon discovering Oracle GoldenGate instance as a target by the agent, management of Oracle GoldenGate instance delivered from within Oracle Enterprise Manager Cloud Control.

Oracle Enterprise Manager Cloud Control 12c management pack for Oracle GoldenGate provides the following functionalities:

- Centralized management of multiple Oracle GoldenGate instances
- General health and performance review

## Oracle GoldenGate veridata

For sites requiring real-time data monitoring, Oracle GoldenGate Veridata provides an easy-to-use yet powerful solution for identifying out-of-synch data before becoming very diverted causing data inconsistency. Deployed together with the Oracle GoldenGate real-time data replication product or separately, Oracle GoldenGate Veridata ensures data consistency is maintained across configured two databases, playing a major rule when performing zero downtime migration and upgrade using Oracle GoldenGate. Figure 1-2 depicts Oracle GoldenGte Veridata.

The major component for configuring Oracle GoldenGate are:

- GoldenGate Veridata server
- Apache Tomcat Web server
- Apache Tomcat Web Server administration tool
- Goldengate Veridata Web
- GoldenGate Veridata repository
- GoldenGate Veridata agent

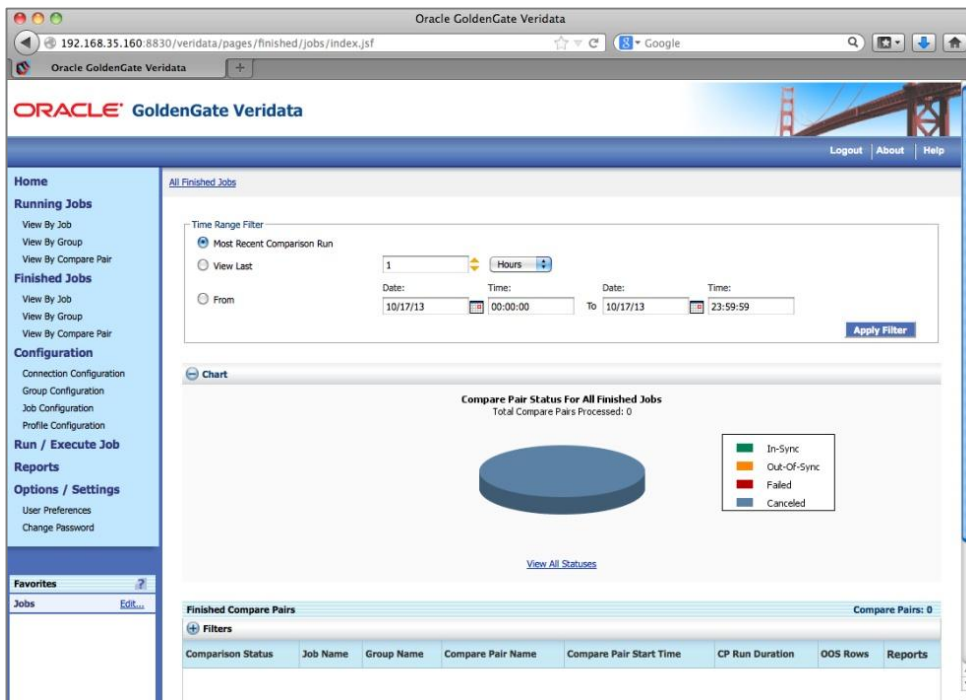


Figure 1-2: Oracle GoldenGate Veridata

Refer to chapter 6 For details regarding Oracle GoldenGate Veridata.

## Oracle GoldenGate adapters

Oracle GoldenGate provide support for different types adapters, these adapters highly promote Oracle GoldenGate integration with legacy systems running legacy applications and ETL-based applications.

### Oracle GoldenGate adapter for flat file

Flat file adapter enables Oracle GoldenGate to produce a series of sequential flat files data and control files from transactional data capture trail files of a source system. Widely used for data warehouse applications ETL and proprietary or legacy applications.

Oracle GoldenGate flat file adapter uses a special type of extract, implemented as user exit, to produce rolling data flat file to be consumed by an interface such as Oracle external tables. The data is formatted by Oracle GoldenGate to the specifications of



the target applications either as delimiter separated values or length delimited values. The advantage for using flat file adapter, near real-time data feeds to legacy systems is accomplished by minimizing the window elapsed timeframe for batch file rollover.

Refer to chapter 12 to learn more about flat file adapter.

## **Oracle GoldenGate adapter for Java**

Java Message Service (JMS) messages generated from Java applications are handled by a special type of capture, VAM message capture. Using a source definitions file a local trail created and routed to the remote system. The remote system data collector server reassemble the data and create a remote trail for the replicate process.

Refer to chapter 12 to learn more about Java adapter.

## **Oracle GoldenGate application adapter for BASE24**

BASE24 is a series of applications produced by ACI Worldwide, Inc. used by payment processing systems for transaction routing and authorization, mostly found at large financial institutions. Oracle GoldenGate provides adapters to integrate BASE24 components heterogeneously, moving data from ATM networks to the core banking systems for online processing. The adapters available for BASE24:

### **D24 adapter**

Enables transactional data to be synchronized bi-directionally in real time with high-availability support while maintaining data consistency and integrity.

### **N24 adapter**

Performs routing and coordination of the notification message to all BASE24 satellite processes up on the file refresh has been completed, providing transparent configuration for bidirectional and unidirectional data flow.

### **T24 adapter**

Resolves issues when moving data among platforms and databases, acts on the transaction log of TLF and PTLF tokens for all transaction types, ensuring the correct order and format.

## Oracle GoldenGate Logger for Enscribe

Provides the mechanism for change data capture on non-audited Enscribe database platforms. The Logger requires the installation of GGSLIB for binding the Oracle GoldenGate and the user's NonStop application. When an Enscribe operation (such as WRITE) executes, GGSLIB intercepts it and sends the record to Logger. The Logger writes the records to the universal format log trail, which can then be processed like any other log-based source.

## Supported database platforms

Oracle GoldenGate heterogeneity aims to support major database management systems. Depending on database type, support for capture and apply is determined, and if the change data capture is log-based. Table 1-1 lists the current supported database platforms by Oracle GoldenGate 11g and higher.

| Database Platform                   | Log-based Capture | Delivery |
|-------------------------------------|-------------------|----------|
| Oracle Database                     | ☑                 | ☑        |
| Microsoft SQL Server                | ☑                 | ☑        |
| Oracle MySQL                        | ☑                 | ☑        |
| IBM DB2 for Linux, UNIX and Windows | ☑                 | ☑        |
| IBM DB2 for i                       | ☑                 | ☑        |
| IBM DB2 for z/OS                    | ☑                 | ☑        |
| Sybase                              | ☑                 | ☑        |
| SQL/MX                              | ☑                 | ☑        |
| Teradata                            |                   | ☑        |
| Oracle TimesTen                     |                   | ☑        |
| HP Neoview                          |                   | ☑        |
| Netezza                             |                   | ☑        |
| ETL products                        |                   | ☑        |
| JMS message queues                  |                   | ☑        |

*Table 1-1: Oracle GoldenGate supported databases*

The vast majority of database capture method is log-based, translated to low-impact, high-performance on the source database system. The log-based capture technology is a key differentiator of Oracle GoldenGate, other extraction tools adapt queuing technology, highly impacting the source database system performance.

Oracle GoldenGate 12c for Oracle database supports two forms of log-based capture: classic capture and Integrated capture. Migration from classic to integrated capture is recommended as direction of new features and enhancements are toward integrated capture architecture. Refer to chapter 10 to learn more about integrated capture.

# Supported operating systems

From the desktop PCs to the mainframe computers enabling the foundation for a comprehensive integration environments. Oracle GoldenGate 11g and higher support the operating systems list on table 1-2.

| Operating system | Version                    | Hardware                                |
|------------------|----------------------------|-----------------------------------------|
| AIX              | 5.1, 5.2 and 5.3           | IBM PowerPC                             |
| HP Nonstop       | D46, D47, D48, G06 and H06 | NSK                                     |
| HP-UX            | 11.11, 11.23 and 11.31     | HP Intel IA64 and HP PA-RISC            |
| Oracle Linux     | 3.0, 4.0, 5.0 and 6.0      | AMD/Intel x64, Intel x86 and Intel IA64 |
| RedHat AS        | 2.1, 3.0, 4.0 and 5.0      | AMD/Intel x64, Intel x86 and Intel IA64 |
| Solaris          | 8, 9 and 10                | Sun SPARC                               |
| SuSE             | 9 and 10                   | Opteron and Intel x86                   |
| Tru64            | 5.1                        | HP OSF                                  |
| Windows          | 2000, 2003, 2008 and XP    | AMD/Intel x64, Intel x86 and Intel IA64 |
| z/OS             | 1.04 and 1.06              | Mainframe                               |

Table 1-2: Operating systems support

## Oracle GoldenGate applications

The extendibility, flexibility and adaptability of Oracle GoldenGate enables the implementation of the software across an enterprise to meet diversity of business objectives. The major areas of applying Oracle GoldenGate listed hereunder followed by a brief description and diagram.

- Query offloading techniques
- Business continuity solutions
- Zero downtime migration and upgrade
- Operational data warehouse and business intelligence
- Data integration services

### Query offloading techniques

Reports are an integral part of any database application systems, the performance overhead associated with running these report on the production systems always has a negative impact on online users, the real customers. Running standard reports such as end-of-year, month closing, etc. demand more system resources and normally scheduled during less workload hours, these are relatively back-dated reports with data gap proportional to the batch window for an online systems operating 24/7 basis using different types for input interfaces.

The solution is implementing Oracle GoldenGate at the enterprise level for the purpose of queries offloading, enabling efficient resource utilization across systems with real-time change data capture (CDC) from OLTP systems acting as data sources to an adapted database instance dedicated for reporting only. Heterogeneous change data capture is very common and proven to be cost-effective, for example the data source OLTP systems running on Oracle Database 11g for Solaris 11 and the reporting instance running on MySQL 5 on Oracle Enterprise Linux 6.

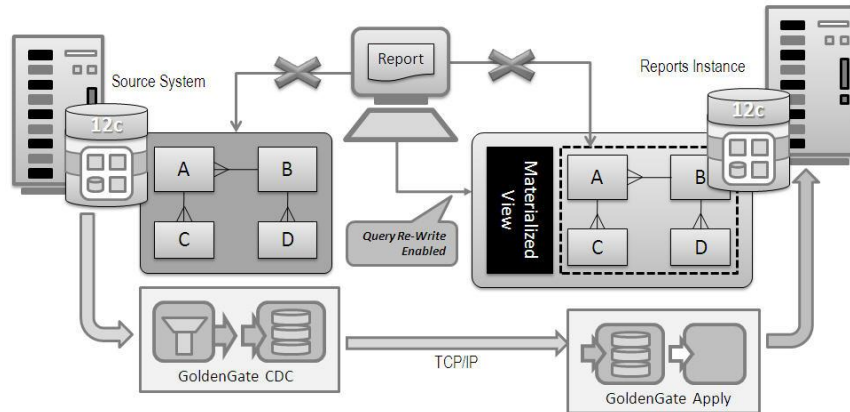


Figure 1-3: Query offloading technique

Oracle database and MS SQL Server shipped with features for optimizing out-of-box report applications. Oracle database materialized view enables query rewrite which greatly enhances report runtime. Figure 1-3 depicts a high-level design for query offloading using Oracle GoldenGate.

Refer to chapter 4 to learn how to implement Oracle GoldenGate for query offloading.

## Business continuity solutions

Oracle GoldenGate capable of delivering different architectures for business continuity solutions, where each solution implemented to suite specific high availability needs, maximize resources utilization and integration with existing high availability solutions such as Oracle Real Application Clusters (RAC) and Oracle Data Guard (DG).

In general, Oracle GoldenGate high availability solutions are divided into two areas: Disaster recovery and high availability.

## Disaster recovery

The remote database site remains on alert for disaster recovery situation. Oracle GoldenGate unidirectional configuration delivers real-time change data capture (CDC) and change data delivery ensuring the gap between the primary (source) and the standby (target) databases is very minimum, with sub-seconds latency. In the event of primary site failure, the standby database is on read-write mode ready to accept connections. Using Oracle GoldenGate Veridata is recommended for maintaining consistent view of the two databases for detecting and applying transactions causing data divergence.

Oracle GoldenGate for disaster recovery is a live-standby database. The database is open for read-write operations, unlike Oracle data guard where the database is either on recovery mode or on read-only recovery mode combined.

## High availability

When using Oracle GoldenGate for high availability solution, it applies the fault-tolerance concept system-wide, which includes the database instance and the database files, this is achieved by implementing Oracle Real Application Clusters (RAC) and Oracle Data Guard (DG) combined. And, because the local and the remote Oracle GoldenGate instances are connected via TCP/IP, constraints related to distance between the two systems are eliminated.

Figure 1-4 shows the high availability architecture is achieved by implementing Oracle GoldenGate bidirectional configuration, users connect to either system based on application, location, user type, etc., providing application scalability and best resource utilization. Data conflict is a major concern and should be carefully addressed when using Oracle GoldenGate for high availability active-active configuration. Oracle GoldenGate built-in data conflict resolution routines minimized custom development and maintenance of data conflict handling.

However; Oracle GoldenGate does not eliminate Oracle Real Application Clusters (RAC) or Oracle Data Guard (DG), to the contrary it creates a first-class high availability platform. Oracle GoldenGate takes advantage of RAC by registering the manager processes (the application) as a resource in Oracle Clusterware. This integration ensures that Oracle GoldenGate is highly available in the event of a node failure. The Oracle GoldenGate instance transparently migrated to a surviving node. Oracle GoldenGate takes advantage of Oracle Data Guard by performing change data capture from the standby database archive log, leaving the primary site intact, which provide the foundation for near real-time reporting instance, offloading the source database from long running queries and reports.

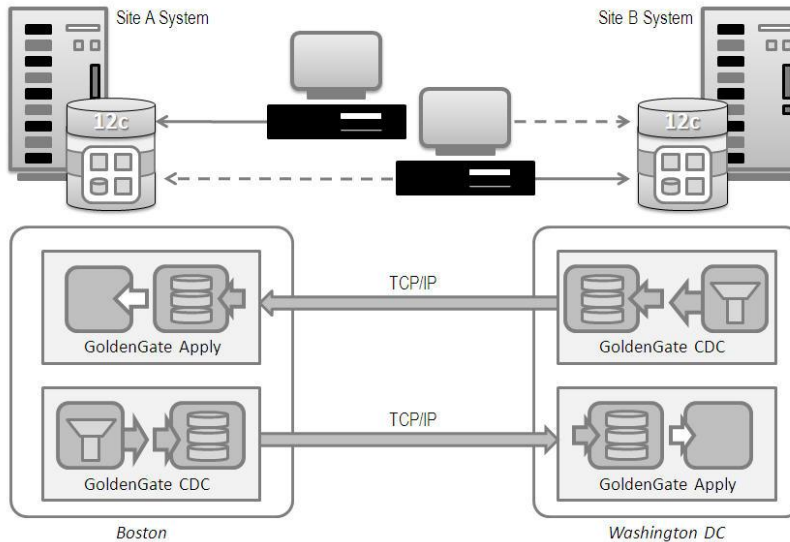


*Figure 1-4: Oracle GoldenGate for high-availability*

Regardless of Oracle GoldenGate configurations for high availability, the target database remains open for read-write operations, it's required by Oracle GoldenGate delivery processes and other applications. Another added-value when using Oracle GoldenGate for high availability solutions is heterogeneity, the source and target systems may differ in terms of hardware architecture and database software.

Refer to chapter 5 to learn how to implement Oracle GoldenGate for high availability.

## Zero downtime migration and upgrade

Oracle GoldenGate can play major rule for the migration and upgrade projects. For systems running critical mission applications, implementing Oracle GoldenGate for zero downtime works equally for homogenous and heterogeneous environment. For example, migration of Oracle Database 11g on IBM AIX (big endian) to Oracle Database 12c on Oracle Enterprise Linux (small endian), Oracle GoldenGate handles endiannes differences internally. When upgrading Oracle Database 11g to Oracle Database 12c, the Oracle Database Upgrade Assistance (DBUA) steps requires database shutdown. The downtime timeframe due to database upgrade varies, depending on different factors, but mainly the database option to upgrade. However; the most obvious benefit when using Oracle GoldenGate for migration and upgrade

is the provision of solid fail back option. The source database remains intact and active until it's time to be shutdown and decommissioned.

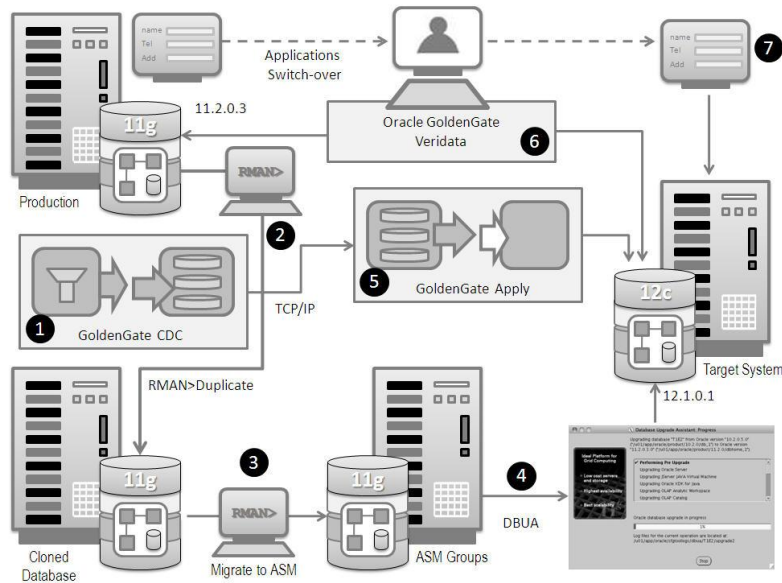


Figure 1-5: Zero downtime migration and upgrade workflow

Achieving zero downtime for migration and upgrade requires cloning the current database. The cloned database is upgraded and ultimately synchronized with the existing production database using Oracle GoldenGate. Use of Oracle GoldenGate Veridata is very recommended during the migration and upgrade. Figure 1-5 shows the steps and workflow for performing zero downtime migration and upgrade.

Refer to chapter 7 to learn how to implement Oracle GoldenGate for zero downtime migration and upgrade.

## Data warehouse and operational business intelligence

Very large database (VLDB) for data warehouse and business intelligence needs optimized data feed, mostly delivered as batch windows ETL using in-house developed workflow processes. When Oracle GoldenGate implemented for online data feed to a data warehouse instance, it delivers an operational repository available online for data warehouse, and with Oracle GoldenGate data transformation and mapping capabilities, the repository is immediately available for business intelligence applications. Oracle GoldenGate eliminates the needs for refresh batch windows and customized ETL utilities. Oracle GoldenGate parallel dynamic processing feature

further enhances data feed speed across heterogeneous systems. The Integration of Oracle GoldenGate with Oracle Data Integrator (ODI) exponentially expedites and optimizes the data feed as an E-LT, promoting highly flexible system. Figure 1-6 shows Oracle GoldenGate for data warehouse.

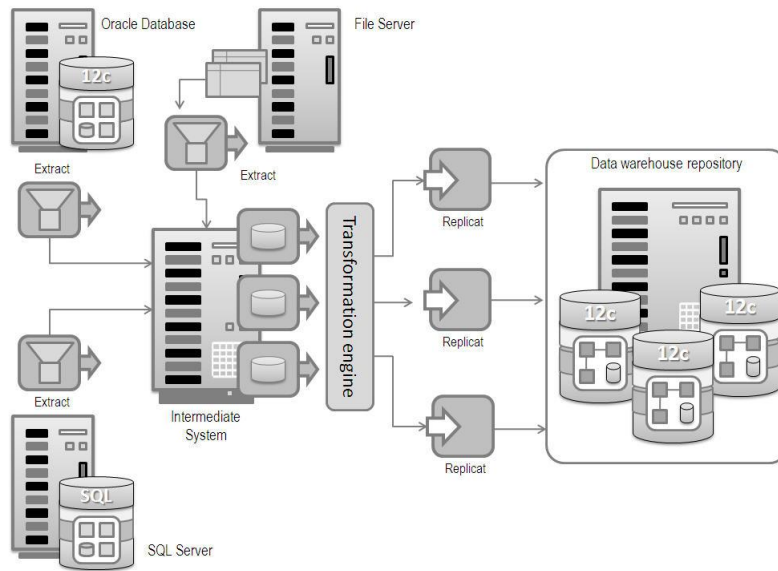


*Figure 1-6: Data warehouse feed from heterogamous systems*

Refer to chapter 10 to learn how to implement Oracle GoldenGate for data warehouse and business intelligence.

## Data integration services

Oracle GoldenGate provides different types of adapters. These adapters enable integration of modern Java technologies and legacy systems applications with Oracle and non-Oracle database. A typical example is Oracle GoldenGate BASE24 adapters used by ATM networks for sending data to payment authorization servers . Another example is a smart-phone application sending decoded messages to an application service provider (ASP), where Oracle GoldenGate implemented as a data hub for filtering and routing data across TCP/IP to consumers. Figure 1-7 depicts payment and ATM networks for transactions processing.

Financial institutes processing payments in real-time require access to heterogeneous systems for authorization, validation and reflect changes. With Oracle GoldenGate a unified infrastructure minimizes the complexity for interfacing foreign systems and



applications. Leaving each organization with its own existing databases and platform maximizes the return of investment (ROI) uncompromised.

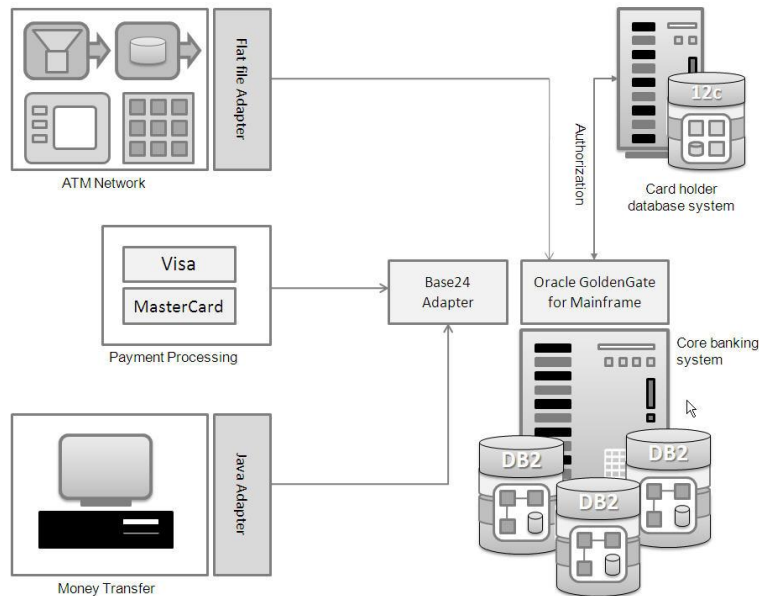


Figure 1-7: payment processing systems

Refer to chapter 12 to learn how to implement Oracle GoldenGate for data integration services.

## Oracle GoldenGate building blocks

The building blocks of Oracle GoldenGate allows high degree for design flexibility for scalable performance, delivering basic-to-complex transactional data replication with transformation capabilities. The basic configuration refer to unidirectional, one-way, data replication from the source to the target database achieving real-time data movement with sub-second latency. However; Oracle GoldenGate commonly deployed to enable integration among heterogeneous systems, moving and consolidating data from various source systems to remote systems with data integrity and consistency.

Oracle GoldenGate extract groups run as loosely or tightly coupled. The primary extract processes running on source systems for change data capture (CDC), the secondary extract processes running on an intermediate systems for transformation and routing change data capture to multiple target systems. The replicate processes

are running on target systems, with parallel processing support for achieving high-performance. Figure 1-8 shows the building blocks found on the source and target systems for configuring an Oracle GoldenGate instance.

An Oracle GoldenGate instance configured for online synchronization consists of the components:

- Source and target databases
- Manager process
- Online extract process
- Local trail files
- Data pump extract process
- TCP/IP Network
- Data server collector
- Remote trail files
- Online replicate process

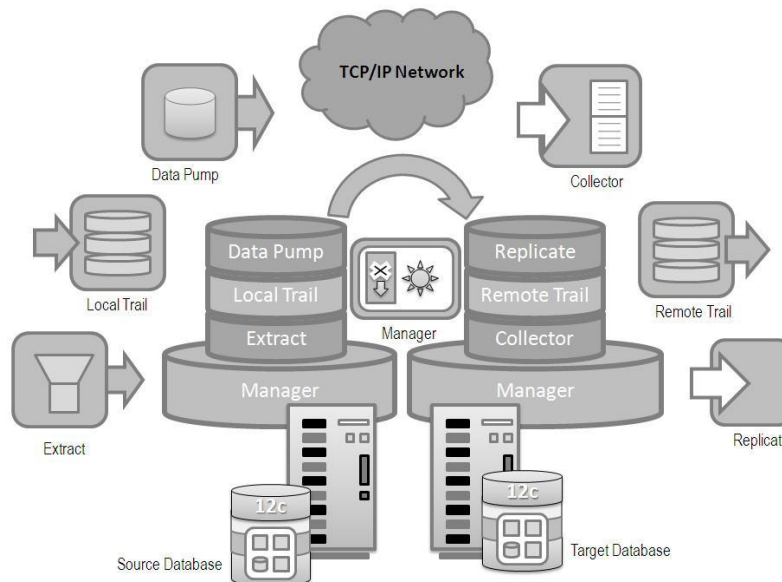


*Figure 1-8: Oracle GoldenGate building blocks*

The functionality for each of the building block components is described next. Figure 1-9 illustrates a simplified configuration approach, showing the components for performing initial-load and data synchronization. The initial-load reads from the source database tables, the data synchronization uses the transaction logs for online data capture and delivery. The use of data pump extract is optional but highly

recommend for supporting fault-tolerance and non-interruptible online capture in the event of network failure and avoiding redundant processing by the online extract.

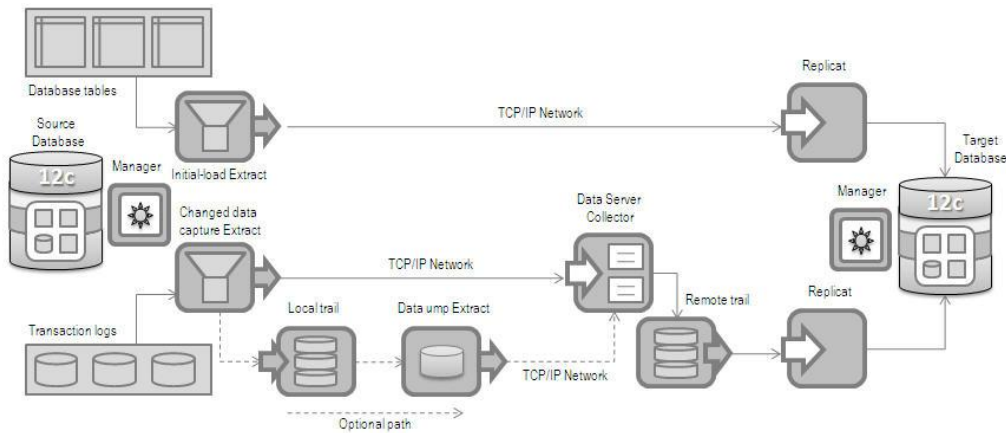


Figure 1-9: Building blocks connected

For initial-load, an initial-load extract read from the source database table and sends insert-type transaction to the remote replicate, which use the database interface to apply the transactions. Refer to chapter 3 to learn how to perform initial-load.

For change synchronization, a primary extract captures all types of transactions (DML and DDL), directly sending data to data server collector on the remote system, which in turn re-assemble the data and writes to the remote trail in a conical file format. The replicate process in the remote system, reads the trails, creates transaction units, and apply them to the target database in the same commit order occurred in the source database, ensuring data integrity and consistency.

The dotted arrows show the option of using what is called data pump extract, which reads the local trails created by the primary extract process and send them to the data server collector. This configuration avoid redundant processing in case of network failure between the source and target systems. Additionally, it offloads the primary extract from performing non-related task of routing to remote systems.

## Source and target databases

Oracle GoldenGate heterogeneous support enables moving data among dissimilar source and target databases. However; The source database must be configured to meet Oracle GoldenGate operating requirements, which is database type specific.

Also, the database platform architecture must be compatible as Oracle GoldenGate 12c is available for 64-bit and 32-bit architecture.

For Linux, to find the platform architecture, use the operating system command:

```
$ uname -a
```

For 64-bit architecture, it will show x86\_64, and x86 for 32-bit architecture.

For MS Windows, use SYSTEMINFO command as shown below:

```
C:\>systeminfo | findstr /i TYPE:
OS Build Type:             Uniprocessor Free
System type:               X86-based PC
```

## Oracle database version

For Oracle source database where the change data capture process is running, the database supplemental logging must be enabled. And, the database should be operating on archive log mode. To find the database version, run the following SQL\*Plus query.

```
SQL> SELECT banner
      2 FROM v$version;
```

BANNER

```
-----
Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 - 64bit
Production
PL/SQL Release 12.1.0.1.0 - Production
CORE      12.1.0.1.0      Production
TNS for Linux: Version 12.1.0.1.0 - Production
NLSRTL Version 12.1.0.1.0 - Production
```

```
SQL>
```

## MS SQL Server database version

To determine the source MS SQL Server database version, from Microsoft SQL Server Management Studio, execute the query:

```
SELECT @@VERSION
```

The source MS SQL Server database must be operating on Full Recovery model mode. Right mouse on database properties and select Options page as shown on

figure 1-10. However, if the target database is for read-only purpose, it may be operated on Simple model mode to avoid growing the log file unnecessarily when performing long running operation such as initial-load.

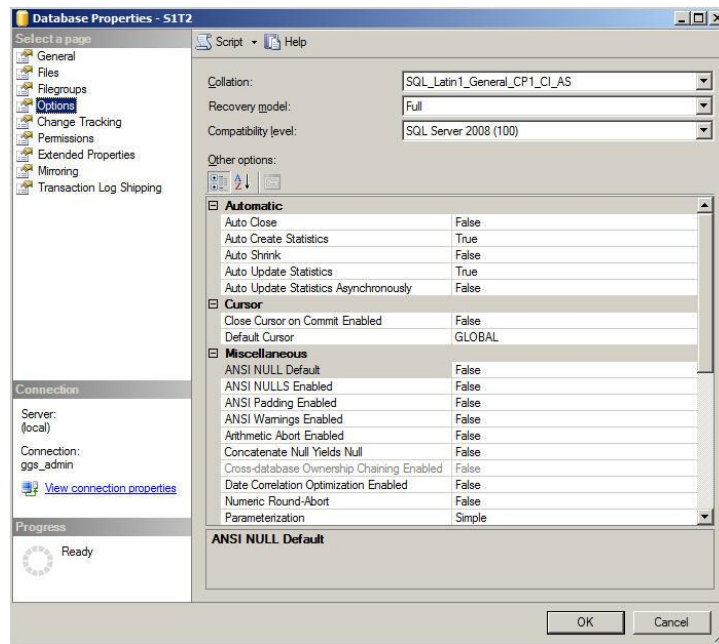


Figure 1-10: MS SQL Server recovery model

## The manager process

Each Oracle GoldenGate instance has a dedicated manager process. After successfully installing Oracle GoldenGate, the manager process is the first process configured and started for managing a running Oracle GoldenGate instance. The manager process listen on specific static TCP/IP port and continues running for handling the instances activities. Starting the manager process requires adding the PORT parameter only to mgr.prm file. Refer to chapter 3 to learn more about setting the managers process parameters.

The manager process is responsible for the following activities:

- Response to errors, events and produces reports
- Manages and allocates ports for dynamic processes
- Start Oracle GoldenGate processes
- Start dynamic Oracle GoldenGate processes
- Performs trail files purge management

- The status of the manager process is limited to either of the following status:
  - STOPPED, manually stopped or unable to start
  - RUNNING, started successfully and listening on specific TCP/IP port

The manager process configuration enables automatic start and restart of extract and replicate processes, this provides unattended extract management for enhancing Oracle GoldenGate high availability.

Oracle GoldenGate manager process on Microsoft SQL Server runs as Windows service, the default services name is GGSMGR. To create multiple Windows Oracle GoldenGate instances, use the GLOBALS parameter MGRSERVNAME <service name> for defining unique Oracle GoldenGate Windows service for each instance.

While Oracle GoldenGate instance is running, shutdown the manager process, quickly make the modifications to mgr.prm and restart it. This technique enhances Oracle GoldenGate availability when re-configuring the manager process.

## The capture and deliver processes

There are four types of Oracle GoldenGate processes, depending on the task, the topology adapted, extracts and replicates processes are configured and defined.

### Online extract

The online extract process perform change data capture (CDC) for committed transactions. The online extract reads the transaction logs from either the online log file or the archive log files. The online redo log files has the advantage of performance for sub-second latency, which cannot be achieved by the archive log file option. The two capture options are mutually exclusive. Figure 1-11 compare the two options. Best practices recommend using online redo log capture with data pump option for fault tolerance and real-time data replication.

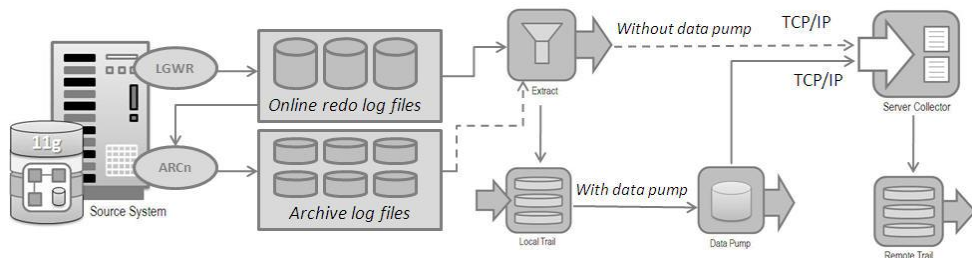


Figure 1-11: Online extract process options

The online extract either writes to local trail files and uses a data pump extract to route data in large block across TCP/IP to a remote dynamic server collector, or across TCP/IP sends the data directly to a remote dynamic process on the target system called data server collector.

The status of the online extract is one of the following:

- STOPPED: The extract process stopped using STOP command
- STARTING: The manager invoking the extract process
- RUNNING: The extract process currently running, reading the transaction log
- ABENDED: The extract process terminated abnormally

The online extract checkpoints it's reads and writes progress to checkpoint file. In the event of failure, the extract process is capable to recapturing from the point of failure forward of the transaction logs.

## Initial-load extract

Reads the source database tables for establishing the target database tables. When coordinated with an online extract for change data capture, an initial-load extract enables high availability as it does not lock the source database tables.

An initial-load extract has several options:

- Writes to remote initial-load files
- Directly invoke initial-load replicate process
- Invokes an initial-load replicate process as remote task

Initial-load extract does not maintain checkpoint, in the event of failure the initial-load extract must be restarted after truncating the target tables. Oracle GoldenGate initial-load techniques is suitable for heterogeneous environment, when the source and target database platforms are different. Oracle GoldenGate techniques for initial-load are:

- File to replicate
- Direct load
- File to database utility
- Database utility

Refer to chapter 3 to learn more about initial-load techniques and architecture using GoldenGate and non-GoldenGate methods.

## Online replicate

The online replicate process reads the remote trail files and applies transactions to the target database, ensuring the transaction committed order of the source database for data consistency and integrity. The online replicate process expects canonical formatted trail files.

The status of the online replicate process is one of the following:

- **STOPPED:** The extract process stopped using STOP command
- **STARTING:** The manager invoking the extract process
- **RUNNING:** The extract process currently running, reading remote trail files
- **ABENDED:** The extract process terminated abnormally

The online replicate checkpoints its read and write progress using either of two options:

- Concurrently to checkpoint file and database table, known as checkpoint table
- Checkpoint file only

It's highly recommended to use checkpoint file and checkpoint table, which is the default. In the event of failure, the replicate process is capable to restart from the point of failure. By repositioning the relative byte address (RBA), the replicate process is capable of re-applying the remote trail transactions to the remote database.

## Special run replicate

Special run replicate is used by the initial-load extract. A special run replicate is started by an initial-load extract and terminated upon completion of the initial-load extract. The special run replicate does not maintain checkpoint, in case of failure the process needs to be restarted.

## Remote task replicate

Designed for initial-load. This is used when the initial-load extract directly invokes a remote replicate for performing the initial-load replicate. When invoking a replicate as a remote task, the data collector server is not used.

Note, when using these for initial-load methods on heterogeneous environment such as from Oracle database to MS SQL Server, a source definition file is required to be created at the source system, then transferred to the target system as ASCII file.



## Data pump extract

The data pump is an optional extract process, referred to as secondary extract. It reads the local trail files and routes data to the target system. It's highly recommended to use a data pump configurations, the advantage of configuring a data pump extract include:

- In the case of network failure, the primary extract continues writing to the local trail, up on network availability, the data pump process starts sending data to remote system. Without a data pump the primary extract stores change data capture in memory until the network become available. This will cause the primary extract to eventually ABENDED due to system resources unavailability.
- Data movement with intensive data transformation occurring at the source system, the data pump extract offloads the primary extract and leaving it dedicated for change data capture only. TCP/IP communication with the data server collector become a function of the data pump extract only.
- For environment composed of multiple target systems, multiple data pumps read the same local trail and route data to the target systems. When one of the connection is not available, only that specific remote system effected. Without data pump it would require using multiple primary extracts each sending data to specific target system data server collector.
- For security and offloading purposes, implementing an intermediate system architecture is an ideal solution. Two data pump extracts must be configured, on source and intermediate system for routing data. Data transformation occurs at the intermediate system entirely offloading the source system.

The data pump extract not to be confused with Oracle database data pump. This is an Oracle GoldenGate process for reading and routing local trail file to remote systems across TCP/IP network.

## The data server collector

The data server collector is a background process, receiving data from either the primary extract (capture) or secondary extract (data pump).

Two types of data server collector are possible, dynamic and statics. Static collector server started manually for the purpose of assigning specific TCP/IP port number to the data collector server, and it can be associated with multiple extract processes. Dynamic collector server is dynamically started, on port allocated by the manager process, and it can be associated with one-only-one extract process.

When using a dynamic mode, starting, stopping and managing the data server collector is fully automatic. The main function is to receive data sent by the extract process, reassemble the data to Oracle GoldenGate universal format (canonical) and write to the remote trail. The data server collector background process is not associated with a checkpoint file, it's either the primary or secondary extract associated with the data collector writing its reads and writes positions to the checkpoint file.

## The trails

It's a form of advanced queue structure for managing a temporary storage areas used by Oracle GoldenGate online extract and replicates trail. The default trail files are written in canonical format by the primary extract, and read by the online replicate process. In addition, Oracle GoldenGate supports alternative formats for use by other utilities. Non-default formats:

- `FORMATASCII`, required by file to database initial-load utility
- `FORMATSQL`, generates DML SQL statements
- `FORMATXML`, generates XML files

None of the non-default trail file formats are not supported by the replicate process.

## Local trail files

When implementing a data pump option, the local trail files created by the online primary extract of the source system. The local trail files composed of two characters prefix and defined using the `EXTTRAIL` or `EXTFILE` parameters. The local trail file created by appending 6-digit to the trail name. If the local extract trail name is 'sa', then the local trail files takes the ranged-name from sa000000 to sa999999. The default trail file size is 10MB, customized using the `MEGABYTES` parameter of the `ADD EXTTRAIL` or `ADD EXTFILE` commands. An extract must be associated with one local trail only.

## Remote trail files

The remote trail files resides on the target system, created by the data collector server. The remote trail files composed of two characters prefix and defined using the `RMTTRAIL` or `RMTFILE` parameters. The remote trail file created by appending 6-digit to the trail name. If the local extract trail name is 'ta', then the remote trail files takes the ranged-name from ta000000 to ta999999. The default trail file size is 10MB, customized using the `MEGABYTES` parameter of the `ADD RMTTRAIL` or `ADD`

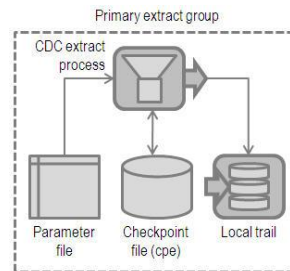
RMTFILE commands. A replicate must be associated with one extract remote trail only.

## Processes grouping

The real-time data movement from source to target systems handled by logical groups. The logical grouping allows an effective allocation and management of physical resource associated with each group. Data down-streaming starts at the capture group, optionally through the secondary group and ends at the replicate group. The number of primary, secondary and replicate groups is effected by the database schemas partitioning. The name of the group is limited to eight characters and created by the ADD EXTRACT and ADD REPLICAT commands.

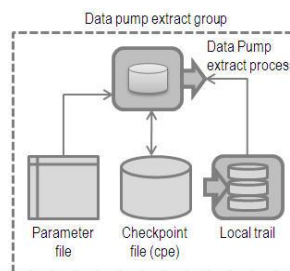
### Primary extract group

This is the capture group, it consists of an extract process for performing change data capture (CDC), a parameter file with the name <group name.prm>, a checkpoint file with the name <group name.cpe>, and an output local trail or remote trail files. The group is also associated with its own discard and report files. The primary process captures committed transaction logs, ensuring transaction integrity and data consistency.



### Secondary extract group

This is the data pump group, it consist of an extract process for performing data routing, an input parameter file that has the name <group name.prm> and a checkpoint file with the name <group name.cpe> and an input local trail. The group is also associated with its own discard and report files. The data pump uses TCP/IP to communicate with a dynamic dedicated data server collector, which re-assemble and write to the remote trail. This group is optional but recommend for fault-tolerance configuration.



## Replicate group

This the change data delivery group, it consists of a replicate process for performing change data delivery, an input parameter file with the name <group name.prm> and a checkpoint table or a checkpoint file with the name <group name.cpr>. The group is also associated with its own discard and report files. The replicate process uses the target database SQL native calls to apply changes from the remote trail, maintaining transactional data integrity data consistency.

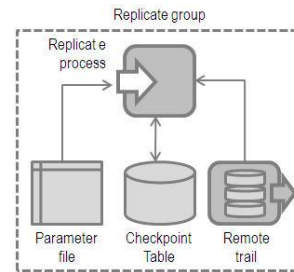


Figure 1-12 depicts the data streaming and the relationship between the three groups, assuming the optional secondary group is used. This diagram shows a serialized data streaming without parallel processing, for details on how Oracle GoldenGate implements parallelism, refer to the section implementing parallel processing.

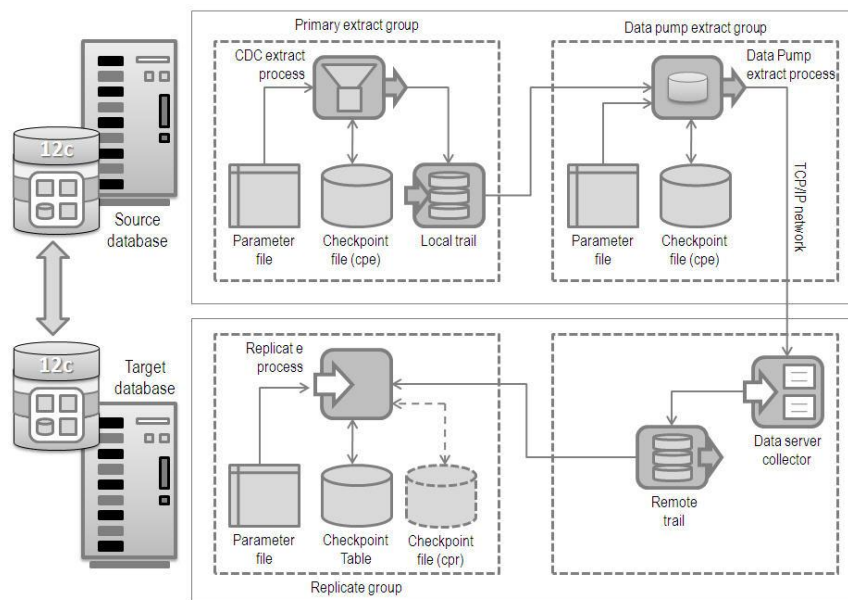


Figure 1-12: Process grouping and workflow

When using intermediate systems, the workflow tends to be loosely coupled, but the concept remains the same. When using integrated capture, the workflow is the same but the change data capture technique is different.

# Oracle GoldenGate Architecture explained

The Oracle GoldenGate instance composed of Oracle GoldenGate software location and an associated manager process. Figure 1-13 depicts Oracle GoldenGate architecture on bidirectional environment.

The **manager process**, running on the source and target systems for handling events and starting dynamic extract and replicate processes. Perform trail file purge management for maintaining an optimized disk utilization.

The **primary extract** referred to as the change data capture (CDC) process, reads the transaction logs and write to either a local trail or send data directly to the data sever collector of the remote system. Depending weather a data pump is used or not. The recommended practice is to use a data pump process for sending data to the remote data server collector, keeping the primary extract dedicated for change data capture only.

The **secondary extract** referred to as the data pump process, reads the local trail file and routes data to remote system. The data is received by a dedicated data collector server.

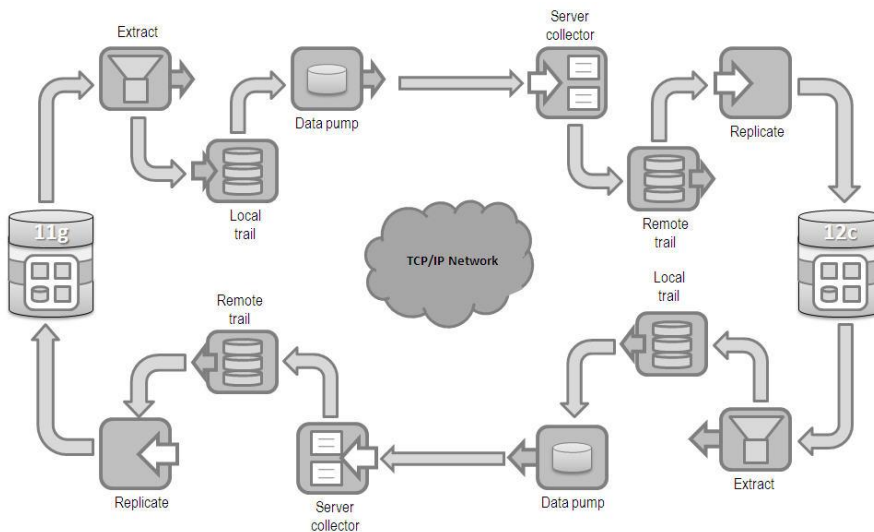


Figure 1-13: GoldenGate processes architecture

The **local trail** is a canonical file format, created by the primary extract, read by the database pump process or other Oracle GoldenGate utilities such as LOGDUMP.

The ***data collector server*** is a background process associated with either a primary extract or a secondary extract. The function of the data collector server is receiving, re-assembling and creating the remote trail on the target system.

The ***remote trail*** is a canonical file format, created by the data collector server. Purging of remote trail files is defined by PURGEAOLDEXTRACTS parameter.

The ***replicate process*** (replicat for short) reads the remote trail file and applying the transactions to the remote database, with data integrity and consistency.

## Network topologies and designs

Depending on business data acquisition requirements, the type of Oracle GoldenGate network topology is determined, and it's very important to be the right topology, because changing the topology at later stage requires changing the configuration as it effect data flow and processes. After the topology is selected, the components required to deploy the design are configured, defined, tested and finally deployed. Oracle GoldenGate network topologies listed hereunder followed by description and diagram.

- Unidirectional for query offloading
- Bidirectional for high availability
- Peer-to-peer for load balancing
- Broadcast for data distribution
- Consolidation for data warehouse
- Cascading for scalability

### Unidirectional for query offloading

This configuration allows data workflow in one direction, from source to target only. In the source system, Oracle GoldenGate extract groups capture and forward Oracle GoldenGate trails to the remote system. One or more replicate groups apply the transactions to the target database keeping the two databases synchronized. When using unidirectional setup for query offloading, change data capture (CDC) should limited to the database tables required by the collection of reports, rather than the complete database. Even though offloading to heterogeneous database such as Oracle MySQL may be an option, but it's wise to consider an offloading database engine that offer high-performance reporting capabilities. Oracle database OLAP option, materialized views and query rewrite supercharge database SQL drastically. Figure 1-14 depicts Oracle GoldenGate components involve for unidirectional setup.

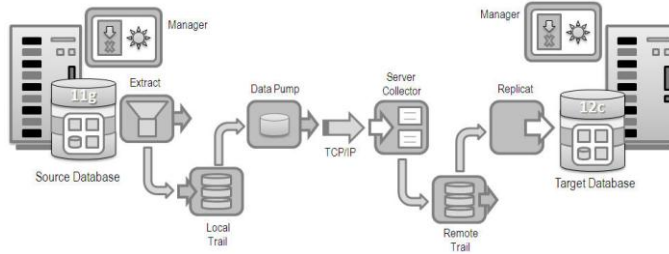


Figure 1-14: Unidirectional Oracle GoldenGate components

## Bidirectional network configuration

This configuration enables forward data flow from source to target, and backward data flow from target to source systems. Because Oracle GoldenGate extract and replicate groups are configured on both systems, it's important to handle data conflict as business case scenarios. Oracle GoldenGate supports common data conflict resolutions, for more details regarding data conflict resolutions, refer to chapter 4.

There are two types of bidirectional setup, it's all depending on the business requirements for implementing either option. Figure 1-15 shows the components and data flow for bidirectional configuration. A major advantage of using Oracle GoldenGate for high availability solution is system proximity. Having the source and target systems connected via TCP/IP eliminates the distance restriction. However; network stability and sufficient dedicated bandwidth are key factor to achieve near sub-second latency.

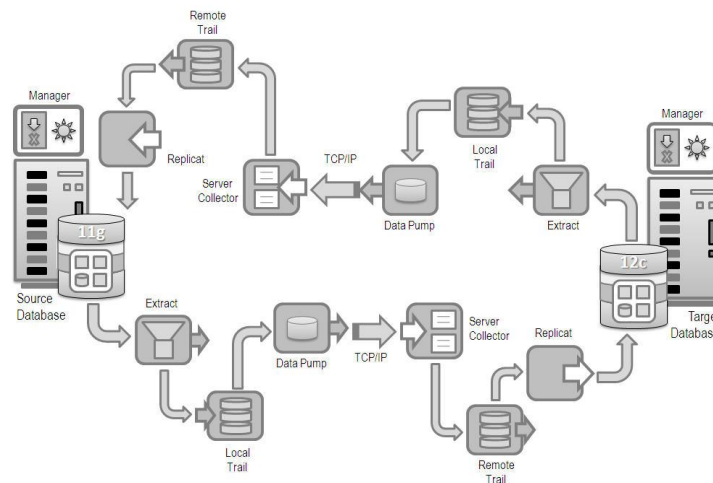


Figure 1-15: Bidirectional configuration

## Active-active

Oracle GoldenGate delivers real-time data replication with sub-second latency. Bidirectional active-active setup promotes high availability with customized application scalability, users are able to connect to either system for conducting transactions.

## Active-passive

When implementing Oracle GoldenGate for disaster recovery solution, it's configured as active-passive setup. The source and target systems are fully configured to support bidirectional data flow, but at all time, the data flow is active from either directions.

## Peer-to-peer configuration

Commonly used for master-to-master data replication. The configuration is used for load balancing and high availability. The two major distinctive features when using Oracle GoldenGate for such design, it can be deployed in heterogeneous environment, for example site A is an Oracle Database 12c, site B is MySQL and site C is Microsoft SQL Server, also supporting replication of subset of the database, this has a positive impact on disk storage and network bandwidth.

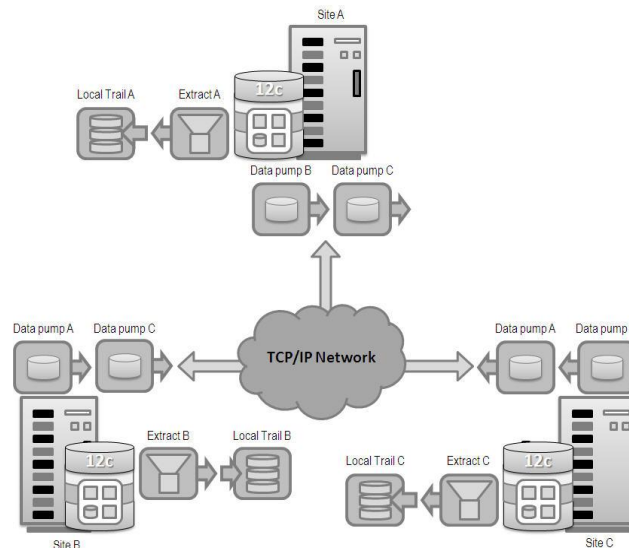


Figure 1-16: Master-to-master replication



Figure 1-16 show master-to-master for three site. Extending this topology by streaming data from a regional master to a local site achieves efficient transaction processing and higher data availability.

The number of sites on peer-to-peer configuration should be limited to 3 sites only. Otherwise, the configuration, data collision handling becomes extremely complex to build, and deploy.

## Broadcast configuration

Used for data distribution from the core business applications database to the enterprise data mart databases. This configuration predominately for read-once, deliver-many, after data is captured from the source transaction logs, it can be routed to any number of destinations (targets), regardless of their of their locations and platform types.

Data streaming is unidirectional from source to targets systems, applying data transformations on the data pump processes or on the targets replicate processes customizes the target databases to suite application specific data requirements. Business intelligence applications benefit from real-time data movement and transformation for real-time application support. An important added value from this configuration is security by ensuring physical data separation. The target databases may be small foot-prints of the source, this is very similar to the concept of virtual private database (VPD), but stored remotely. The data is pertaining to subset of related users and/or application for specific region. Figure 1-17 shows the data flow from source to targets.

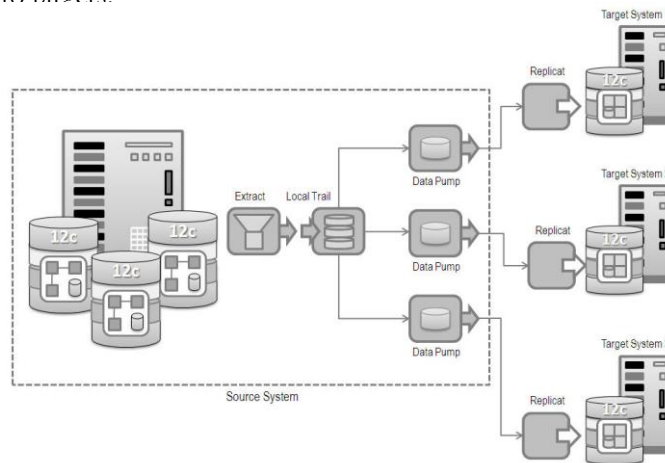


*Figure 1-17: Oracle GoldenGate broadcasting topology*

## Consolidation configuration

Used for the data warehouse, the enterprise data warehouse repository receives data from the enterprise online transactional systems. An E-LT process using Oracle data integrator (ODI) enabling real-time population of the star schemas for reporting and business intelligence. Figure 1-18 shows consolidation of three databases into the data warehouse database.

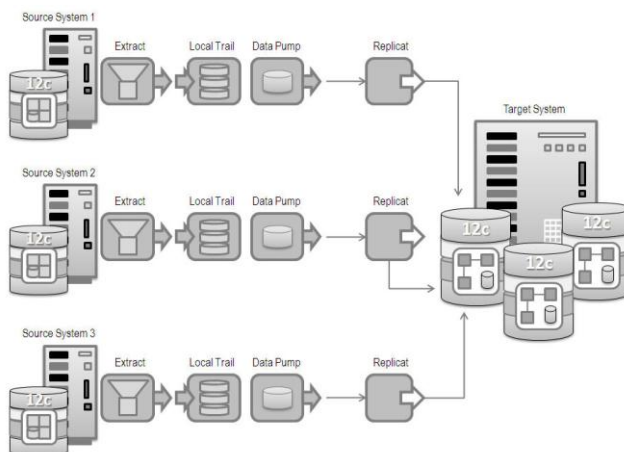


Figure 1-18: Consolidate for data warehouse repository

## Cascading configuration

Sites performing intensive data transformation on source and target systems, using an intermediate GoldenGate instance with sufficient processing resources enhances scalability by dedicating the source and target systems for capture and deliver functions only, leaving data transformation to occur within the intermediate system only. The intermediate system can be with or without a database. Figure 1-19 illustrates an intermediate system without a database, only the client software required for installing Oracle GoldenGate software.

The best practices recommends using multiple data pumps, each dedicated to specific target system, network failure of one system is isolated leaving other systems unaffected by the ABENDED data pump process.

In the case of columns mapping or data transformation involving an intermediate system, a source definition file and target definition file must be provided to the intermediate system.

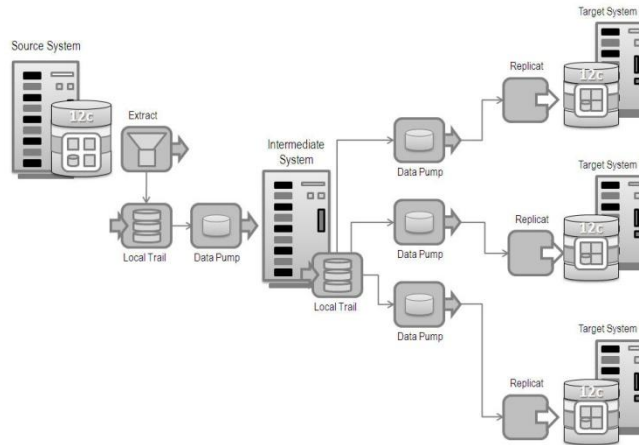


Figure 1-19: Cascade using intermediate system

## Oracle GoldenGate command-line interfaces

The Oracle GoldenGate Software command interface, or GGSCI performs administration and management tasks such as adding extracts, adding replicates, trails, login to the database and view reports. On Linux/UNIX operating systems, set the environment variable `LD_LIBRARY_PATH` before starting GGSCI.

```
$export LD_LIBRARY_PATH=$ORACLE_HOME/lib
$cd $GGS_HOME
$./ggsci
```

When using GGSCI for creating and modifying parameter files ensures saving the parameter files on the correct locations. You may also define a non-default preferred text editor. GGSCI is client application tool, it does not require additional software installation, it only uses Oracle database library. Figure 1-20 shows GGSCI on Oracle Enterprise Linux. When using GGSCI for MS SQL Server, run it as Windows administrator user and the manager process runs as a local Windows service.

Examples for using GGSCI interface:

- Manage manager process, performing start, stop and info commands
- Login to the database
- Adding and managing checkpoint table
- Adding and managing extract and replicate processes
- Creating and managing parameter files
- Manages extract and replicate processes
- Generates reports and displaying performance statistics

```
oracle@ggs-source:/u01/app/oracle/ggs - 92x29
$ cd $GGS_HOME
$ pwd
/u01/app/oracle/ggs
$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.0.0 17185003 OGGCORE_12.1.2.0.0_PLATFORMS_130924.1316_FBO
Linux, x64, 64bit (optimized), Oracle 11g on Sep 25 2013 00:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.

GGSCI (ggs-source) 1> █
```

*Figure 1-20: Oracle GoldenGate software command interface (GGSCI)*

Another important utility for troubleshooting Oracle GoldenGate instance error messages and performing content-search of trail files is the command-line interface LOGDUMP. Figure 1-21 shows the utility, ready to open a trail file.

The command line to run the utility:

```
$export LD_LIBRARY_PATH=$ORACLE_HOME/lib
$cd $GGS_HOME
$./logdump
```

```
oracle@ggs-source:/u01/app/oracle/ggs - 92x29
$ cd $GGS_HOME
$ pwd
/u01/app/oracle/ggs
$ ./logdump

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

Logdump 1 > █
```

*Figure 1-21: Oracle GoldenGate log file dump utility*

For detail regarding the LOGDUMP utility, refer to chapter 13.

## Obey files

The need to run batch job is essential for performing sequence of related tasks, such as running the sequence of commands to deploy an end-to-end Oracle GoldenGate instance. The GGSCI OBEY command reads and execute the commands listed on the text file, known as obey file.

Start by creating a text file, insert one GGSCI command per line and save the file. The example below creates the file info\_all.oby, then from GGSCI enter the file name as an input to the GGSCI OBEY command. The file name is normally has the file extension OBY, but it's not necessary, hence info\_all.oby and info\_all.txt are valid file names.

```
-- File: /home/oracle/oby/info_all.oby
--
start mgr
info all
start extract ES1E201
start extract EPMP01
info all
```

```
GGSCI (ggs-source) 7> OBEY /home/oracle/oby/info_all.oby
```

The GGSCI OBEY command allows nesting of obey files for invoking another obey file from within an obey file. To enable nesting, insert the parameter ALLOWNESTED at the beginning of the file before invoking and obey file as illustrated next.

```
-- File: /home/oracle/oby/info_all.oby
--      /home/oracle/oby/add_tran.oby
--
ALLOWNESTED
START mgr
INFO ALL
START EXTRACT es1e201
START EXTRACT epmp01
INFO ALL
OBEY /home/oracle/oby/add_tran.oby
```

```
GGSCI (ggs-source) 18> OBEY /home/oracle/oby/info_all.oby
```

```
GGSCI (ggs-source) 19> ALLOWNESTED
```

```
Nested OBEY scripts allowed.
```

GGSCI (ggs-source) 20> START mgr

Manager started.

GGSCI (ggs-source) 21> INFO ALL

| Program | Status  | Group   | Lag at Chkpt | Time Since Chkpt |
|---------|---------|---------|--------------|------------------|
| MANAGER | RUNNING |         |              |                  |
| EXTRACT | STOPPED | EPMP01  | 00:00:00     | 00:00:29         |
| EXTRACT | STOPPED | ES1E201 | 00:00:00     | 00:00:28         |

GGSCI (ggs-source) 22> START EXTRACT es1e201

Sending START request to MANAGER ...  
EXTRACT ES1E201 starting

GGSCI (ggs-source) 23> START EXTRACT epmp01

Sending START request to MANAGER ...  
EXTRACT EPMP01 starting

GGSCI (ggs-source) 24> INFO ALL

| Program | Status  | Group   | Lag at Chkpt | Time Since Chkpt |
|---------|---------|---------|--------------|------------------|
| MANAGER | RUNNING |         |              |                  |
| EXTRACT | RUNNING | EPMP01  | 00:00:00     | 00:00:32         |
| EXTRACT | RUNNING | ES1E201 | 00:00:01     | 00:00:00         |

GGSCI (ggs-source) 25> OBEY /home/oracle/oby/add\_tran.oby

\*\*\*\* Halting script [/home/oracle/oby/info\_all.oby], starting  
script [/home/oracle/oby/add\_tran.oby]...

GGSCI (ggs-source) 26> dblogin userid ggs\_admin, password oracle

Successfully logged into database.

GGSCI (ggs-source) 27> add trandata osm\$repapi.customers

Logging of supplemental redo data enabled for table  
OSM\$REPAPI.CUSTOMERS.  
TRANDATA for scheduling columns has been added on table  
'OSM\$REPAPI.CUSTOMERS'.  
GGSCI (ggs-source) 28>

```
GGSCI (ggs-source) 28> **** Terminating script
[/home/oracle/oby/add_tran.oby], resuming script
[/home/oracle/oby/info_all.oby]...
```

```
GGSCI (ggs-source) 28>
```

However; for executing an SQL\*Plus script and operating system commands, use GGSCI SHELL command within an obey file followed by the shell script file. For example, in certain situation, it may be needed to restart the database from within Oracle GoldenGate, which can be developed as follows:

- Create an obey file, db\_shut.oby, that has the lines:

```
SHELL /home/oracle/oby/db_shut.sh
```

- Prepare the database shutdown script and save it as db\_shut.sh. Ensure the file has execute permission.

```
#!/bin/bash
export ORACLE_SID=S1E2
export ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
sqlplus /nolog <<EOF
conn / as sysdba
startup force
exit
EOF
```

- Invoke the obey file from GGSCI command-line.

```
GGSCI (ggs-source) 1> OBEY /home/oracle/oby/db_shut.oby
```

When using GGSCI SHELL command to execute an operating system shell commands, these commands are executed with the access rights of the Oracle GoldenGate software owner, applying access rights of the database privileges and operating system security.

## Parameters files

There are two types of parameters, global parameters type apply to all Oracle GoldenGate instance processes, and runtime parameters type apply to specific process. When the same parameter specified as global and runtime, the runtime parameter takes precedence and override the global parameter. Parameter files are ASCII format, with one line per parameter or append the character ‘&’ to create multi-line parameter. The following syntax for the PURGEOLDEXTRACTS parameter produces the same results.

```
PURGEOLDEXTRACTS ./dirdat/*, USECHECKPOINTS, MINKEEPHOURS 1

PURGEOLDEXTRACTS ./dirdat/*, &
USECHECKPOINTS, &
MINKEEPHOURS 1
```

The global parameters are defined on the file GLOBALS (upper case without file extension) located at Oracle GoldenGate root directory. Global parameters are relatively static and rarely modified. Table 1-3 list common global parameters.

| Parameter        | Description                                                                                    |
|------------------|------------------------------------------------------------------------------------------------|
| CHECKPOINTTABLE  | Specifies instance-wide checkpoint table                                                       |
| ENABLEMONITORING | Required for enabling Oracle GoldenGate monitor                                                |
| MGRSERVNAME      | The name of the manager process when installed as local Windows service                        |
| CHARSET          | The character set to when reading the parameter files, overriding the default operating system |
| GGSCHEMA         | The name of the schema for supporting DDL replication between two Oracle databases             |
| SYSLOG           | Used for filtering the messages types recorded in ggserr.log file to override the default ALL. |

*Table 1-3: Common global parameters*

When changing global parameters on GLOBALS file from within GGSCI, it's required to exit and restart GGSCI utility for activating the newly added parameters.

Runtime parameters are specific to an extract or replicate process allowing more control for process runtime. The default location for run-time parameter files is ./dirprm/ directory of Oracle GoldenGate location. Using text editor such as 'vi' on Linux/UNIX, notepad on Windows or GGSCI to create a new or edit an existing parameter file, the group name specified at GGSCI command-line, which becomes the parameter file name, must matches the name followed the EXTRACT parameter. The process name, group name and parameter file name are not case sensitive.

The following listing is the parameter file for extract ES1E201, created using GGSCI command-line.

```
GGSCI (ggs-source) 2> EDIT PARAMS es1e201

EXTRACT ES1E201
SETENV (ORACLE_SID=S1E2)
SETENV (ORACLE_HOME="/u01/app/oracle/product/11.2.0/dbhome_1")
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@S1E2, PASSWORD oracle
```



```
EXTTRAIL ./dirdat/sa
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
```

## Configure the instance manager process

A GoldenGate instance can have one-and-only manager process. Which remains running to monitor activities and start dynamic processes. There are few important facts about the manager process.

- The only required parameter is `PORT`, which specifies a static TCP/IP listening port number.
- There can be one-and-only-one manager process per Oracle GoldenGate instance.
- The manager process status is either `RUNNING` or `STOPPED`.
- The manager process parameter file name and location must be `./dirprm/mgr.prm` only.
- The manager process supports up to 250 dynamic ports.
- Though the manager should remain running at all times, it may be stopped, change the parameter and restart without affecting currently running processes.

The following is an example of common manager process parameters. These parameters' values should be carefully determined as they are reflected on how Oracle GoldenGate instance manager process handles events.

```
-- mgr.prm
--
PORT 7811
DYNAMICPORTLIST 9001-9250
PURGEOLDEXTRACTS ./dirdat/*, USECHECKPOINTS, MINKEEPHOURS 1
AUTORESTART ER *, RETRIES 10, WAITMINUTES 5
LAGREPORTHOURS 2
LAGINFOMINUTES 5
LAGCRITICALMINUTES 10
```

This manager process is listening on TCP/IP port 7811. It searches and allocates from the range of ports 9001 to 9250 for dynamically starting processes.

The parameter `PURGEOLDEXTRACTS` allows the manager to perform trail file management. The option `USECHECKPOINTS` to delete trail files after being transferred to the target system and applied. The option `MINKEEPHOURS 1` instructs to delete the trail file until there has been no activity for one hour.

The manager will automatically start any extracts and replicates processes at startup and will attempt to restart any extracts and replicates with status **ABENDED** after waiting for five minutes, but only up to ten attempts.

The parameter **LAGREPORTHOURS 2** instructs the manager to report lag statistics every two hours for those processes that have five and ten minutes of latency using the parameters **LAGINFOMINUTES 5** and **LAGCRITICALMINUTES 10**. The message recorded as informational for lags of five minutes and critical for lag greater than ten minutes.

To find the basic details about the currently running manager, issue the **GGSCI** command:

```
GGSCI (ggs-source) 8> INFO mgr
```

```
Manager is running (IP port ggs-source.7811, Process ID 21649).
```

To view the manager process parameters, issue the **GGSCI** command:

```
GGSCI (ggs-source) 9> VIEW PARAMS mgr
```

```
PORT 7811
DYNAMICPORTLIST 9001-9250
PURGEOLDEXTRACTS ./dirdat/*, USECHECKPOINTS, MINKEEPHOURS 1
AUTORESTART ER *, RETRIES 10, WAITMINUTES 5
LAGREPORTHOURS 2
LAGINFOMINUTES 5
LAGCRITICALMINUTES 10
```

To issue stop and restart the manager process, enter the **GGSCI** commands:

```
GGSCI (ggs-source) 12> STOP mgr!
```

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

```
GGSCI (ggs-source) 13> START mgr
Manager started.
```

```
GGSCI (ggs-source) 14>
```

Even though the manager process may be stopped without effecting the currently running extracts and replicates processes, it should remain running at all time to handle runtime events and activities. A common case is when very quickly changing a

specific parameter value in the manager file, stopping and restarting the manager process immediately.

Refer to chapter 3 for more details regarding the parameters description.

## Configure change data capture

The change data capture is a function of the primary extract, during the capture operation the transaction and data remain in memory until the transaction is committed or rolled back. Committed transactions are written to the local trail in case a secondary extract (data pump) is configured or to the remote trail, otherwise. Best practices states configuring a data pump provides resiliency against network failures.

The configuration below instructs the primary extract to perform change data capture for the database tables CUSTOMERS and POLICIES.

```
EXTRACT es1e201
SETENV (ORACLE_SID=S1E2)
SETENV
(ORACLE_HOME="/u01/app/oracle/product/11.2.0/dbhome_1")
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@S1E2, PASSWORD oracle
EXTTRAIL ./dirdat/sa
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
```

The primary extract group name must matches the parameter file name es1e201.prm. The SETENV parameter access environment variable is required for operating the primary extract. The parameter USERID enables the primary extract to login to the database S1E2. The parameter EXTTRAIL instructs the primary extract to write to the local trail ./dirdat/sa. The TABLE parameter lists the tables for change data capture.

Oracle GoldenGate adapts its own universal format for exchanging data across heterogeneous systems. The parameter EXTTRAIL specifies the location and the name of the local trail files. The name is a two character prefix appended with 6 digit starting 000000 and ending with 999999, then recycling begin at 000000.

## Creating the primary extract

From GGSCI command-line use the ADD EXTRACT command to define the primary extract. The parameter file may not exist when adding the extract, but

required to start it. The ADD EXTRACT syntax below is used for defining the primary and secondary extracts.

```
ADD EXTRACT <group name>
{, SOURCEISTABLE |
, TRANLOG [<bsds name>] |
, INTEGRATED TRANLOG |
, VAM |
, EXTFILESOURCE <file name> |
, EXTTRAILSOURCE <trail name> |
, VAMTRAILSOURCE <VAM trail name>}
{, BEGIN {NOW | yyyy-mm-dd [:hh:mi:[ss[.cccccc]]]} |
, EXTSEQNO <seqno>, EXTRBA <relative byte address> |
, LOGNUM <log number>, LOGPOS <byte offset> |
, EOF |
, LSN <value> |
, EXTRBA <relative byte address> |
, EOF | LSN <value> |
, PAGE <data page>, ROW <row>}
[, THREADS <n>]
[, PASSIVE]
[, PARAMS <parameter file>]
[, REPORT <report file>]
[, DESC "<description>"]
```

The ADD EXTRACT command has three sections as highlighted above: The extract type, the capture begin timestamp or absolute location and the optional control parameters. For defining the primary extract ES1E201, we will use the basic parameters.

```
GGSCI (ggs-source) 25> ADD EXTRACT es1e201, TRANLOG, BEGIN  
now, THREADS 1  
EXTRACT added.
```

The above ADD EXTRACT command has successfully defined the extract group ES1E201. This extract performs change data capture from the transaction logs (online redo log files) indicated by the option TRANLOG. The timestamp to begin change data capture is the time of up on successful completion of the command indicated by option BEGIN NOW. The option THREADS 1 indicates the extract captures from a single thread on non-RAC Oracle database instance.

Using a secondary extract requires creating a local trail. The local trail ./dirdat/sa is associated with the primary extract ES1E201.

```
GGSCI (ggs-source) 26> ADD EXTTRAIL ./dirdat/sa, EXTRACT
es1e201, MEGABYTES 32
EXTTRAIL added.
```

After change data capture by the primary extract process written to the local trail, then the role of the secondary extract process (data pump) begins by routing data to one or more target systems. The following refers to the data pump process configuration.

```
EXTRACT epmp01
USERID ggs_admin@s1e2, PASSWORD oracle
PASSTHRU
RMTHOST ggs-target, MGRPORT 7812
RMTRAIL ./dirdat/ta
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
```

The data pump process epmp01 reads the local trail files. The parameter PASSTHRU indicate the source and target database tables structure are identical. NOPASSTHRU parameter is used when using the @RANGE function to support parallel processing or COLMAP parameter implemented for non-identical tables.

The parameter RMTHOST specifies the target system, the MGRPORT is the target system manager process.

The parameter RMTRAIL specifies the location and name of remote trail on the target system.

## Creating the secondary extract

Define the secondary group using ADD EXTRACT GGSCI command. The secondary extract reads the local trail and routes data across TCP/IP to remote data collector server.

```
GGSCI (ggs-source) 27> ADD EXTRACT epmp01, EXTTRAILSOURCE
./dirdat/sa
EXTRACT added.
```

The above ADD EXTRACT command define the secondary extract EPMP01, the extract reads the local trail ./dirdat/sa indicated by the option EXTTRAILSOURCE. The remote trail ./dirdat/ta associated the secondary extract defined using the ADD RMTRAIL command.

```
GGSCI (ggs-source) 28> ADD RMTRAIL ./dirdat/ta, EXTRACT
epmp01, MEGABYTES 32
```

RMTTRAIL added.

Use the command **START EXTRACT \*** for starting all non-started extracts. Then find the current status of the extracts by issuing **INFO ALL** command. This **INFO ALL** command shows the process status, and more importantly lagging and checkpoint details.

```
GGSCI (ggs-source) 34> START EXTRACT *
```

```
Sending START request to MANAGER ...  
EXTRACT EPMP01 starting
```

```
Sending START request to MANAGER ...  
EXTRACT ES1E201 starting
```

```
GGSCI (ggs-source) 35> INFO ALL
```

| Program | Status  | Group   | Lag at Chkpt | Time Since Chkpt |
|---------|---------|---------|--------------|------------------|
| MANAGER | RUNNING |         |              |                  |
| EXTRACT | RUNNING | EPMP01  | 00:00:00     | 00:00:13         |
| EXTRACT | RUNNING | ES1E201 | 00:00:01     | 00:00:00         |

```
GGSCI (ggs-source) 36>
```

Refer to chapter 3 for more details regarding extracts parameters.

## Configure change data delivery

The third and last step of the configuration is the change data delivery, the replicate process. The replicate process uses the target database native SQL calls to apply the transactions found on the remote trail. Reapplying the remote trail transactions is subject to the database constraint definition for primary keys and referential integrity.

The following is the change data delivery group parameter file.

```
REPLICAT rs1e201  
SETENV (ORACLE_SID=T1C1)  
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)  
USERID ggs admin@t2c1, PASSWORD oracle  
ASSUMETARGETDEFS  
DISCARDFILE ./dirrpt/rs1e201.dsc, PURGE  
MAP osm$repapi.customers, TARGET osm$repapi.customers;  
MAP osm$repapi.policies, TARGET osm$repapi.policies;
```

The replicate process RS1E201 connects to Oracle Database 12c, it references the list of mapped tables for applying transactions to the target database using its associated remote trail files. The ASSUMETARGETDEFS parameter is for homogenous environment when using identical source and target database tables structure. The MAP and the corresponding TARGET parameters are used to map the source with target table.

## Creating change data delivery process

The change data delivery is the replicate process reading from remote trail and applying transactions to the remote database using database native SQL calls. The ADD REPLICAT syntax is used to define change data delivery group.

```
ADD REPLICAT <group name>
{, SPECIALRUN |
, EXTFILE <file name> |
, EXTTRAIL <trail name>}
[, BEGIN {NOW | yyyy-mm-dd:hh:mm[:ss[.cccccc]]} |
, EXTSEQNO <seqno>, EXTRBA <rba>]
[, CHECKPOINTTABLE <owner.table> | NODBCHECKPOINT]
[, PARAMS <parameter file>]
[, REPORT <report file>]
[, DESC "<description>"]
```

The ADD REPLICAT command has three sessions: The replicate process type and data source, the apply begin timestamp and the optional control parameters.

Use the basic parameter to create the replicate group RS1E201, specifying the data source and the database checkpoint table name. The replicate process requires checkpoint database table, by default. Using the default option requires creating the checkpoint table before defining the replicate process RS1E201.

```
GGSCI (ggs-target) 4> ADD CHECKPOINTTABLE
ggs_admin.ggchkpttab
```

Successfully created checkpoint table ggs\_admin.ggchkpttab.

```
GGSCI (ggs-target) 16> ADD REPLICAT rs1e201, EXTTRAIL
./dirdat/ta, CHECKPOINTTABLE ggs_admin.ggchkpttab
REPLICAT added.
```

Use the command START REPLICAT to start the replicate process, then issue INFO ALL to confirm status of the replicate process.

```
GGSCI (ggs-target) 19> START REPLICAT *
```

```
Sending START request to MANAGER ...  
REPLICAT RS1E201 starting
```

```
GGSCI (ggs-target) 20> INFO ALL
```

| Program  | Status  | Group   | Lag at Chkpt | Time Since Chkpt |
|----------|---------|---------|--------------|------------------|
| MANAGER  | RUNNING |         |              |                  |
| REPLICAT | RUNNING | RS1E201 | 00:00:00     | 00:00:53         |

```
GGSCI (ggs-target) 21>
```

Refer to chapter 3 for more details regarding parameter description and database checkpoint table options.

## Using macros

Macros enable an easier and more efficient method for building Oracle GoldenGate instance parameter files. A macro defines a group of related parameters so that they can be referenced from within the parameter files, simplifying the configuration of large parameter files with repetitive parameters, commands and functions by reusing the group of parameter defined by the macro.

The definition of macros either *local* within the same process parameter file and invoked from within parameter file only, or *global* to Oracle GoldenGate instance by defining macros library and invoking the macros from any extract or replicate processes parameter files.

Prior to invoking macros, they must be defined. The definition of a macro similar to sub-programs, where a macro has a name beginning with # character, optionally followed by zero or more parameters. The macro body placed between the BEGIN and END statements. The syntax for defining macros follows.

```
MACRO #<macro name>  
PARAMS (#<parameter1>, #<parameter2>, #<parameter>)  
BEGIN  
    <Macro body>  
END;
```

After defining a macro, it can be used on the parameter files for:



- Manager process
- Primary and secondary extracts
- Replicate process
- GLOBALS parameter file

The best approach to work with macros is by creating macro libraries. Let's assume the parameters LOGIN and PASSTHRU are included for the majority of parameter files. Defining and invoking a macro replaces explicitly adding the those parameters to the parameter files. The parameter INCLUDE at the top of the parameter file enables invoking any of the macros defined within the macro library file.

- Start by defining the macro named #S1E2LOGIN inside the macro library file

```
$ vi /u01/app/oracle/ggs/dirprm/macrosqe2.mac
```

```
MACRO #S1E2LOGIN
BEGIN
    USERID ggs_admin@S1E2, PASSWORD oracle
    PASSTHRU
END;
```

- Invoke the macro from within the secondary extract EPMP01

```
INCLUDE /u01/app/oracle/ggs/dirprm/macrosqe2.mac
EXTRACT epmp01
#S1E2LOGIN()
RMTHOST ggs-target, MGRPORT 7812
RMTTRAIL ./dirdat/ta
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
```

## The modular architecture

Regardless of the design adapted to meet the business requirements for data acquisition, the level of database schema complexity, the data flow option, the database heterogeneity; The modular building blocks of Oracle GoldenGate is capable to produce a dynamic architecture for achieving real-time data movement with sub-second latency. The capture-and-apply processes operate as loosely or tightly coupled, loosely coupled in the sense the capture processes may be cascaded across multiple systems, where the primary extract process (capture) running on the source database system reading the transaction logs, and the secondary extract process (data pump) running on an intermediate system physically networked with the source system using TCP/IP. These capture-apply processes runs outside of the database management

system, hence Oracle GoldenGate delivers safe, high performance and scalable platform with minimum impact on existing system for supporting heterogeneous data movement.

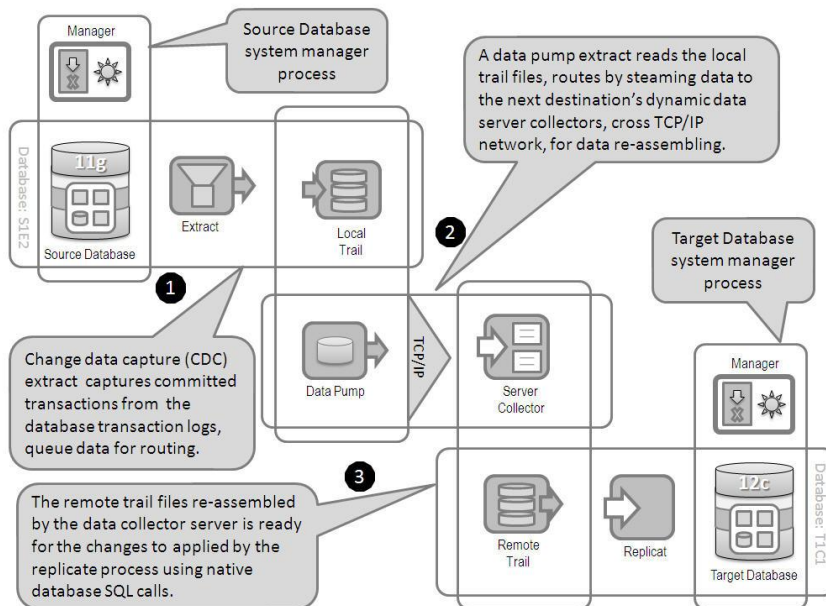
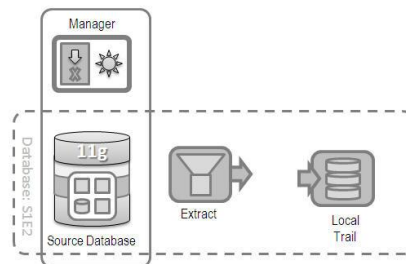


Figure 1-22: Oracle GoldenGate modular architecture

Figure 1-22 illustrates the building block for the three modular blocks:

- Change data capture phase using primary extract group
- Routing phase using secondary extract group
- Apply phase using change data delivery group

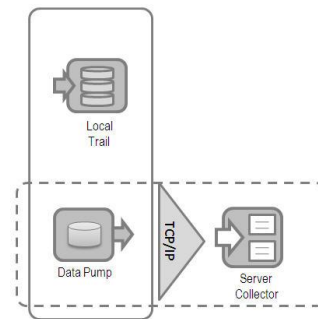
The capture phase, it involves one or more primary extract capturing *only committed* transactions as they become available on the transaction logs. This is occurring outside of the database, hence minimum impact sustained by the source database. For achieving a fault-tolerance non-interruptible data movement, the primary extract stages captured data in the form of advanced queues composed of local trail files ready for routing to one or more



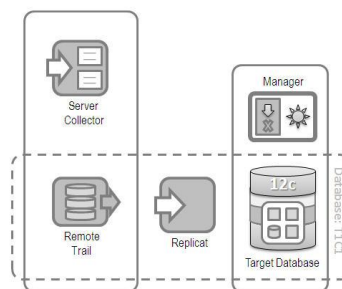
destinations (targets). The target systems could be the final destinations (sources) or an intermediate systems for achieving high-performance broadcasting network topology or intensive data transformation.

The source system manager process normally configured to manage local trail files purge task for sustaining an optimized disk space on the production system. It's also configurable to implement extract processes auto-start during system startup and restart for abnormally terminated extract processes. Trail files purge and extracts auto-start features are implemented for minimizing data movement discontinuity on sites requiring uninterruptable data processing.

The routing phase, it involves one or more secondary extracts called data pumps for routing transactions, *not trail files*, to one or more destinations. Having a data pump within the source system is optional, but very highly recommend to support fault-tolerance as explained on the section – Capture-route options. The receiving component is the data server collector for re-assembling the data and managing the remote queue. The data server collector is a dynamic server and associated with a data pump. The communication from the data pump extract to the data server collector occurs across TCP/IP network allowing the design for loosely coupled system where the source and target systems are geographically distributed.



The third, or last phase involves one or more change data apply, replicate processes know by Replicat. Replicate processes running on the target system use the database interface (SQL calls) for applying the transactions while maintaining transactions ordering for data integrity. The input of the replicate processes is the remote trail files created by the data collector server. High-performance apply is achieved using Oracle GoldenGate flexible parallelism architecture allowing multiple replicate processes to read from a single trail composed of trail files.



The manager process on the target system normally configured to manage remote trail files purge task for sustaining an optimized disk space on the target system. It's also configurable to implement replicate processes auto-start during system startup and restart for abnormally terminated replicate processes.

The flexibility of designing an operational Oracle GoldenGate instance is depicted logically on figure 1-23 followed by the interpretation.

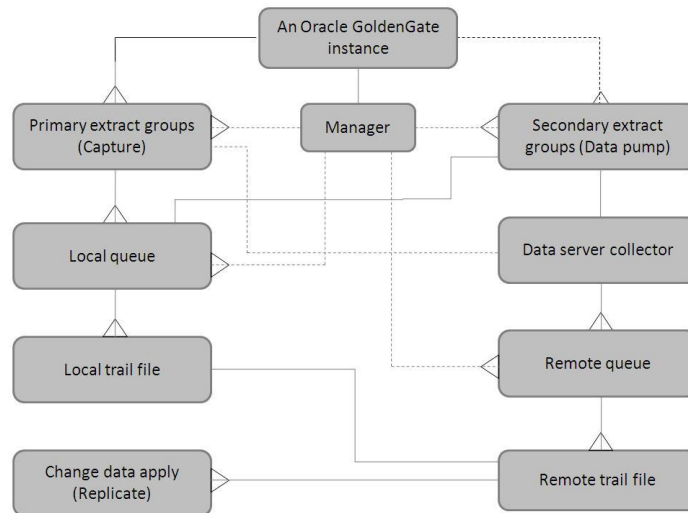


Figure 1-23: GoldenGate components relationship

An Oracle GoldenGate instance must have one-and-only-one manager. An *operational* Oracle GoldenGate instance must be composed of one-or-more primary extract groups (capture), but may have one-or-more secondary extract groups (data pump).

Each primary extract must create one-and-only-one trail, and each trail must be composed of one-or-more local trail files. Each secondary extract (data pump) is associated with one-and-only-one data server collector and each data server collector creates one-and-only-one remote trail. Each remote trail composed of one-or-more remote trail files.

An Oracle GoldenGate instance must be composed of one-or-more replicate processes, and each trail file is applied using the target database SQL native interface.

The manager process may manage one-or-more primary extracts, one-or-more secondary extracts and one-or-more trails.

## Capture-route options

There are two options available for deploying Oracle GoldenGate as visually compared in figure 1-24. The upper section shows a primary extract (capture) with a

data pump extract refers to as option-A and the lower section shows a primary extract (capture) without a data pump extract refers to as option-B. Best practices highly recommends implementing the primary extract with a data pump extract, option-A.

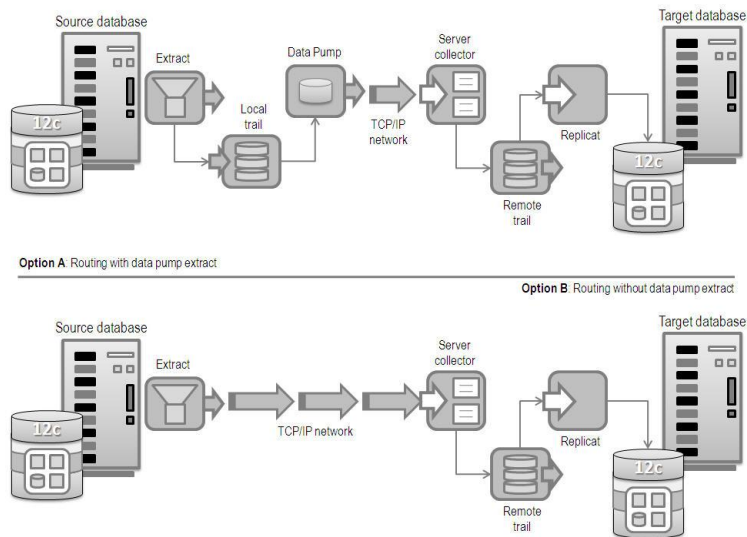


Figure 1-24: End-to-end configuration options

In the event of TCP/IP network disconnection from the source to target systems, option-A supports fault-tolerance because the primary extract continues to deliver change data capture (CDC) by writing to the local trail, only the data pump extract will be terminated temporary. Once TCP/IP network restored, the data pump is restarted and data continue sending data to the data server collector. Under the same circumstances, option-B will cause the primary extract process to terminate due to system resources unavailability as a result of unable to send data to the remote data server collector. Depending on the downtime timeframe, the transaction logs may not be available after the network is restored and the extract started. This scenario requires manual changes for reading the transaction logs from archived log files.

The storage overhead of writing to local trail files is minimized by the purge task automatically performed by the manager process and the parameters used to configure the purge. In most cases, under normally working extract-route-replicate for option-A the local trail files disk space is negligible and should not cause performance degradation. Option-B is selected in case where the source and target Oracle GoldenGate instances are running on the same host, moving data between two heterogeneous or homogenous databases.

## Commit sequence number

The commit sequence number, CSN identifies a particular point in time of the source transaction during the capture. It's constructed from the database, for Oracle database it's the same as the system change number (SCN), for MS SQL Server it's composed of three parts: the virtual log file number, the segment number within the virtual file and the entry number. Oracle GoldenGate uses CSN during apply to identify the point in time at which the transaction is committed for maintaining transaction consistency and data integrity.

## Checkpoints

Primary extract, data pump extract and replicate processes record their reads and writes positions along the data flow. Successfully created extract and replicate groups are associated with a checkpoint file. The checkpoint file name for an extract group is <group name.cpe>, the checkpoint file name for a replicate group is <group name.cpr>. For primary extracts performing change data capture save their writes and reads positions to a checkpoint file to prevent redundant extraction from the transaction logs. For data pump extracts performing routing to remote destination across TCP/IP save their reads and writes positions to prevent redundant extraction of the local trail. For replicates processes performing change data apply save their reads and writes positions to a checkpoint table or checkpoint file to prevent redundant extraction from the remote tail files and reapply to the target database.

Oracle GoldenGate checkpoint ensures fault tolerance in case of failure. The recovery process is similar to Oracle database automatic instance recovery.

For primary and data pump extracts checkpoint file is the only option. For replicates, there are two options; checkpoint file or checkpoint database table and checkpoint file combined, the default is the combined option. The name of the table either specified on GLOBALS file for all replicate groups using the parameter CHECKPOINTTABLE followed by the schema and the table name, or per replicate group specified when adding the replicate process using GGSCI command ADD REPLICATE. To override the default and use a checkpoint file for the replicate group, use the option NODBCHECKPOINT when adding the replicate process through GGSCI command ADD REPLICATE.

## Intermediate systems

For achieving an enterprise scalability in term of data transformation and routing capabilities, the use of Oracle GoldenGate instance as an intermediate system

receiving very active downstream of trail files from multiple source systems, then transforming and routing trail files to multiple remote systems. An intermediate system is configurable as:

- Non-database intermediate system
- Database intermediate system

Figure 1-25 depicts intermediate system, it's composed of an Oracle GoldenGate instance configured with data pump processes for transforming and routing data to target systems.

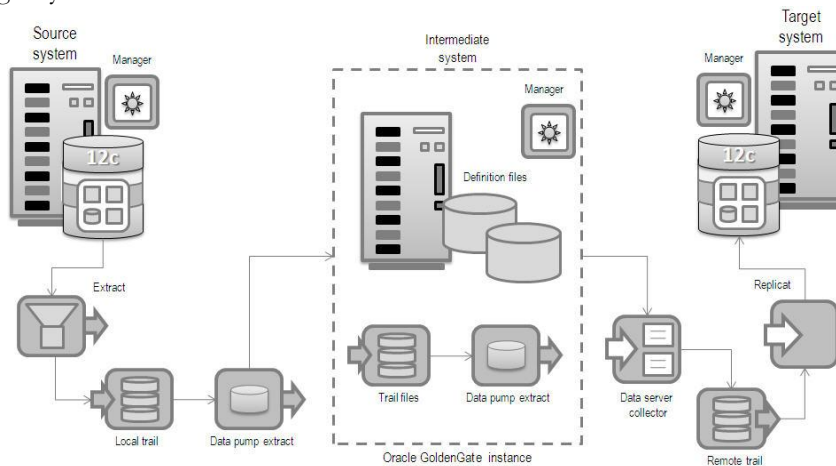


*Figure 1-25: Intermediate system for scalability*

In addition to achieving best scalability using an intermediate system, it also support enhanced security. Which is true in the case the target system receiving data from non-trusted source systems. With intermediate system in place, prior to routing the data to the target database, the data is verified, transformed then sent over secure channel to the target database. The intermediate system uses multiple data pump extracts to route data to multiple targets, and has data definition files for each target requiring data transformation before routing to the target database, the advantage of performing transformation within the intermediate system is to offload the target database from the overhead associated with intensive data transformation and filtering.

## Integrated capture

The purpose of the integrated capture is to offload the transaction logs processing by Oracle GoldenGate from the source database (production) to a downstream system.

Integrated capture is a specialized form of intermediate system continuously receiving the source database REDO as standby log files, where the integrated capture process on the downstream system performs change database capture (CDC) and forwarding the transaction data to the next downstream process such data pump, which in turn routes data to the target system. Figure 1-26 shows the architecture of the integrated capture environment.

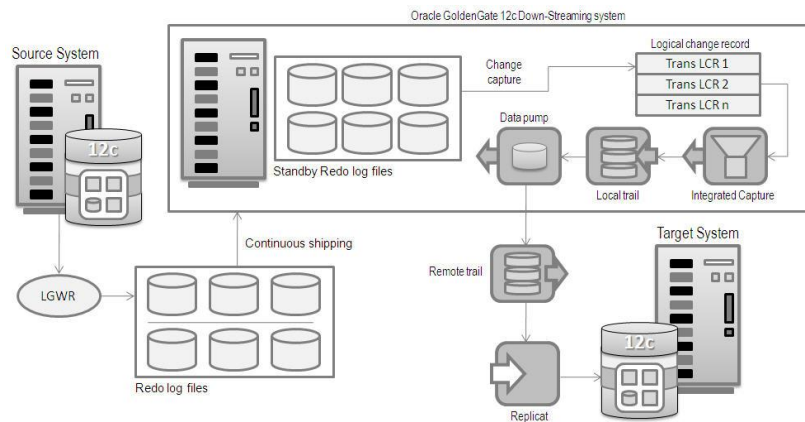


Figure 1-26: Integrated capture architecture

There are three options of integrated capture architecture, available with Oracle Database version 10.2 or higher.

- Option 1: Real-time same system integrated capture
- Option 2: Real-time separate system integrated capture
- Option 3: No-real-time separate system integrated capture

Real-time same system integrated capture requires installing Oracle GoldenGate 12c on source system, the integrated capture delivers remote trails from the source database REDO logs files.

Real-time separate system integrated capture, Oracle GoldenGate is not installed on the source system, but on a downstream system continuously receiving the source database REDO log files, where the integrated capture process delivers the remote trails on real-time.

No-real-time separate system integrated capture, Oracle GoldenGate is not installed on the source system, but on a downstream system continuously receiving the source database REDO log files, but Oracle GoldenGate integrated is operating on archive log only (ALO) mode readings the standby archived log file.



Refer to chapter 9 to learn more about Oracle GoldenGate integrated capture.

## Columns mapping

It's the norm having the structure of source database table differs from the target database table requiring handling using columns mapping techniques. This is applicable to tables with different column names, missing column names and different columns ordering. Use the TABLE or MAP statement COLMAP option to develop the mapping for each pair of columns.

The key elements of columns mapping are:

- Source definition file, generated using DEFGEN Oracle GoldenGate utility
- Mapping clause: COLMAP with USEDEFAULTS or COLMATCH

Let's illustrate columns mapping to handle different column names on the CUSTOMERS table for the two columns CUST\_ADDRESS and CUST\_ADDR.

The source database CUSTOMERS table

```
S1E2>DESC customers
Name                               Null?    Type
-----
CUST_NO                           NOT NULL NUMBER
CUST_NAME                         NOT NULL VARCHAR2 (128)
CUST_EMAIL                       NOT NULL VARCHAR2 (128)
CUST_MOBILE                      NUMBER
CUST_ADDRESS                    VARCHAR2 (128)
```

The target database CUSTOMERS table

```
T1C1>desc customers
Name                               Null?    Type
-----
CUST_NO                           NOT NULL NUMBER
CUST_NAME                         NOT NULL VARCHAR2 (128)
CUST_EMAIL                       NOT NULL VARCHAR2 (128)
CUST_MOBILE                      NOT NULL VARCHAR2 (16)
CUST_ADDR                      VARCHAR2 (128)
```

## Definition file utility

For heterogeneous data replication and replicating dissimilar tables on homogenous environment, a definition file created using Oracle GoldenGate DEFGEN utility is required. The utility generates the a source file describing the structure of the source to enable Oracle GoldenGate handles the internal data type mapping.

The steps to use the DEFGEN utility for generating a source definition file and apply it to the replicate process parameter file follows.

- Prepare the parameter file defgen.prm. Save the parameter file on /dirprm directory.

```
DEFSFILE ./dirdef/oracle.def, PURGE
USERID ggs_admin, PASSWORD oracle
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
```

- Run the utility using the parameter file. The utility creates the definition file specified by the parameter DEFSFILE.

```
$cd $GGS_HOME
$ ./defgen PARAMFILE ./dirprm/defgen.prm REPORTFILE
./dirrpt/defgen.rpt
```

- Using ASCII mode, transfer the source definition file oracle.def to the target system into the directory ./dirdef
- Modify the replicate parameter file to include the parameter SOURCEDEFS, and remove the parameter ASSUMETARGETDEFS.

```
REPLICAT rs1e201
SETENV (ORACLE_SID=T1C1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@t1c1, PASSWORD oracle
DISCARDFILE ./dirrpt/rs1e201.dsc, PURGE
SOURCEDEFS ./dirdef/oracle.def
MAP osm$repapi.customers, TARGET osm$repapi.customers, &
COLMAP (USEDEFAULTS, cust_addr = cust_address);
MAP osm$repapi.policies, TARGET osm$repapi.policies;
```

When using the DEFGEN utility to produce the source definition file, the character set of the source and target systems presumably matching. However; to guarantee the

definition file generated compatible with the target system character set, specify the CHARSET parameter.

#### **CHARSET UTF-8**

```
DEFMSFILE ./dirdef/oracle.def PURGE CHARSET UTF-8  
USERID ggs_admin@S1E2, PASSWORD oracle  
TABLE osm$repapi.customers;  
TABLE osm$repapi.policies;
```

## **Columns mapping clauses**

There are two techniques for handling columns mapping, using the parameters COLMAP or COLMATCH. Depending on how dissimilar the two tables one would be more preferred to use than the other.

The two methods are illustrated using the replicate process TS1E201.

### **COLMAP clause**

Use COLMAP on cases where the columns have completely different names, or where one table has large number of column with only few columns differences. The benefit of COLMAP is using the USEDEFAULT option which causes implicit mapping of identical columns, followed by explicit column-to-column mapping process as illustrated next.

```
REPLICAT rs1e201  
SETENV (ORACLE_SID=T1C1)  
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)  
USERID ggs_admin@t1c1, PASSWORD oracle  
DISCARDFILE ./dirrpt/rs1e201.dsc, PURGE  
SOURCEDEFS /u01/app/oracle/ggs_2/dirdef/oracle.def  
MAP osm$repapi.customers, TARGET osm$repapi.customers, &  
COLMAP (USEDEFAULTS, cust_addr = cust_address);  
MAP osm$repapi.policies, TARGET osm$repapi.policies;
```

Notice, column mapping syntax allows the target column placed only on the left side of the assignment statement.

### **COLMATCH clause**

Use COLMATCH on cases where tables have similar structure with different column names. COLMATCH clause uses either the NAMES or the SUFFIX/PREFIX options.

The NAMES option explicitly maps two columns names, while the SUFFIX/PREFIX relies on substring mapping to dynamically determine columns mapping. SUFFIX is used for ignoring columns suffix and PREFIX is used for ignoring column prefix.

The clause COLMATCH PREFIX PK\_ maps any target column having a PK\_ prefix to any source column of similar name. For example, PK\_POL\_NO with POL\_NO.

COLMATCH using the NAMES option illustrated on the replicate parameter file.

```
REPLICAT rs1e201
SETENV (ORACLE_SID=T1C1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@T1E2, PASSWORD oracle
DISCARDFILE ./dirrpt/rs1e201.dsc, PURGE
SOURCEDEFS ./dirdef/oracle.def
COLMATCH NAMES CUST_ADDR = CUST_ADDRESS
MAP osm$repapi.customers, TARGET osm$repapi.customers;
MAP osm$repapi.policies, TARGET osm$repapi.policies;
```

Refer to chapter 10 to learn more about column mapping.

## Built-in functions

Functions are mostly used for performing data transformation, required by the target database table structure and applications. Oracle GoldenGate supports categories of functions for characters, numbers, dates and general usages. The same principle of database and user-defined functions applies to Oracle GoldenGate functions in term of:

- Accepting of zero or more input parameters
- Allowing functions nesting
- Returning a value

Oracle GoldenGate functions invoked using the character '@' followed immediately by the function name. Functions are may be found in the parameter file of primary extract (capture), secondary extract (data pump) or replicate process.

## Numbers and character functions

Table 1-4 lists number and character built-in functions.

| Function Name | Description                                                                                                              |
|---------------|--------------------------------------------------------------------------------------------------------------------------|
| @COMPUTE      | Returns the result of arithmetic expressions into a target column. The return value is of type string                    |
| @NUMBIN       | Converts a binary string into number                                                                                     |
| @NUMSTR       | Converts a string to number                                                                                              |
| @STRCAT       | Concatenates sequence of strings                                                                                         |
| @STRCMP       | Compares two strings to determine whether they are equal, or the first string is less or greater than the second string. |
| @STREQ        | Tests two strings to see if they equal or not. Return 1 for equal strings, returns 0 for not equal strings.              |
| @STREXT       | Returns substring from a string indicated by a begin and an end positions                                                |
| @STRFIND      | Returns the first occurrence of a substring within a string                                                              |
| @STRLEN       | Returns the length of a string                                                                                           |
| @STRTRIM      | Removes the leading spaces from a string                                                                                 |
| @STRNCAT      | Concatenates one or more strings up to the specified number of characters per string                                     |
| @STRNCMP      | Compares two strings up to specific number of characters                                                                 |
| @STRRTRIM     | Trims the trailing spaces in a string                                                                                    |
| @STRSUB       | Substitutes one string for another within a string                                                                       |
| @STRTRIM      | Trims both leading and trailing spaces in a string                                                                       |
| @STRUP        | Converts the string to uppercase                                                                                         |

*Table 1-4: Number and character built-in functions*

Examples of using the character strings functions @STRUP, @STRFIND and @STRCAT follows.

- Converting the customer address to upper case for the target column.

```
MAP osm$repapi.customers, TARGET osm$repapi.customers, &
COLMAP (USEDEFAULTS,
        cust_addr = @STRUP(cust_address));
```

- Validate the email address contains the '@' character:

```
MAP osm$repapi.customers, TARGET osm$repapi.customers, &
COLMAP (USEDEFAULTS,
        cust_addr = cust_address,
        cust_email = @STRFIND (CUST_EMAIL, "@") > 0);
```

- Construct an upper case email by concatenating the name with specific domain. The upper case email achieved by nesting the two function @STRCAT and @STRUP.

```
MAP osm$repapi.customers, TARGET osm$repapi.customers,
COLMAP (USEDEFAULTS,
        cust_email=@STRUP(@STRCAT(cust_name,
                                   '@precisetrace.com')));
```

## Date functions

Table 1-5 lists Oracle GoldenGate date related functions.

| Function Name | Description                                                                                |
|---------------|--------------------------------------------------------------------------------------------|
| @DATE         | Returns the date from specific input format, and output the date to specific output format |
| @DATEDIFF     | Returns the difference between two dates                                                   |
| @DATENOW      | Returns the current date and time value                                                    |

Table 1-5: Oracle GoldenGate built-in date functions

An example of using the date function @DATENOW:

- Use the current date for the column CUST\_NAME.

```
MAP osm$repapi.customers, TARGET osm$repapi.customers, &
COLMAP (USEDEFAULTS, &
        cust_addr = cust_address,
        cust_date = @DATENOW());
```

## General functions

Table 1-6 lists Oracle GoldenGate general purpose functions.

| Function Name | Description                                                                                                                       |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------|
| @BINARY       | Ensures source binary data remains binary in the target column when the source column is defined as a character column data type. |
| @BINTOHEX     | Converts a binary string to hexadecimal string value                                                                              |
| @GETENV       | Returns Oracle GoldenGate environment variable value.                                                                             |
| @GETVAL       | Extracts parameter value from a stored procedure when using                                                                       |

|           |                                                                                                                         |
|-----------|-------------------------------------------------------------------------------------------------------------------------|
|           | SQLEXEC clause, requires the name procedure and parameter name.                                                         |
| @HEXTOBIN | Converts from hexadecimal to binary string value                                                                        |
| @RANGE    | Used to provide parallel processing for enabling multiple secondary extract or replicate processes read the same trail. |
| @TOKEN    | Used to retrieves environment variable value for column mapping.                                                        |

Table 1-6: Oracle GoldenGate general purpose built-in functions

Example of using general function @TOKEN:

- Retrieve Oracle GoldenGate environment variable for hostname, operating system user and database name.

```
MAP osm$repapi.customers, TARGET osm$repapi.customers,
COLMAP (USEDEFAULTS, &
        cust_addr = cust_address,
        cust_date = @DATENOW(),
        cust_hostname = @TOKEN ('TK_HOST'),
        cust_osname = @TOKEN ('TK_OSUSER'),
        cust_dbname = @TOKEN ('TK_DBNAME'));
```

## Data selection and filtering

Data selection and filtering are fundamentals to data integration, Oracle GoldenGate provides the capabilities to perform data selection and filtering to meet applications requirements. There are several advantages for performing data selection and filtering from within GoldenGate, the most appealing is data security.

### Data security

Implemented as a form of physical private database (PPD), where data routing to the target databases may be designed based on application context. A sophisticated use cases is implementing physical private database based on partitioned source table. Figure 1-27 illustrates limiting data access by implementing physical private database based on the table partitioning design.

The CUSTOMERS table is partitioned by range. The partitioning column is CUST\_NO created as shown below.

```
SQL> CREATE TABLE customers (
2  cust_no      NUMBER NOT NULL PRIMARY KEY,
3  cust_name    VARCHAR2(128) NOT NULL,
```

```

4  cust_email    VARCHAR2(128) NOT NULL,
5  cust_mobile   NUMBER,
6  cust_address  VARCHAR2(128))
7  PARTITION BY RANGE (cust_no)
8    (PARTITION p100000 VALUES LESS THAN (100001),
9     PARTITION p200000 VALUES LESS THAN (200001),
10    PARTITION p300000 VALUES LESS THAN (300001),
11    PARTITION p999999 VALUES LESS THAN (MAXVALUE) );

```

Table created.

SQL>

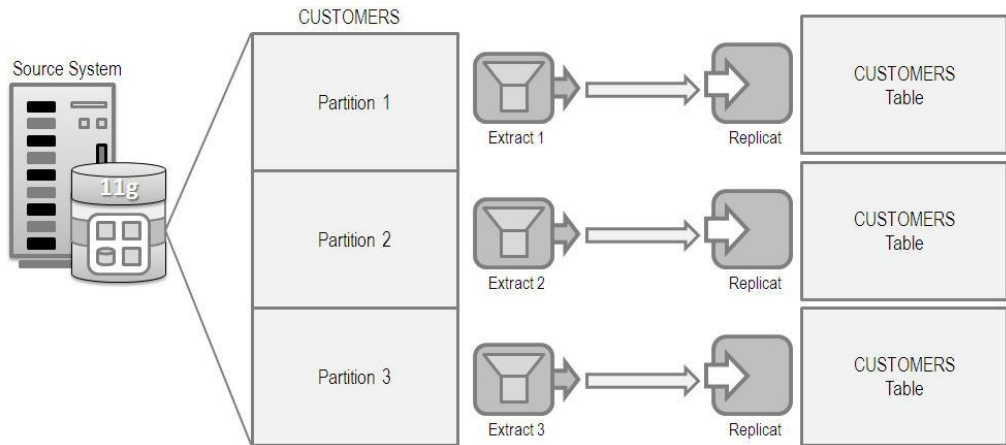


Figure 1-27: Physical private database (PPD)

For each partition, a dedicated primary extract performs change data capture using selection clause, creating its own local trail. Next, a data pump extract routes the data to a data server collector which creates a remote trail. The replicate process applies the transactions to the target database CUSTOMERS table.

The above example is more realistic when implemented using interval or list partitioning methods. The combination of Oracle GoldenGate selection, filtering, tokens and application context contribute for building secure data model.

## Data selection

Oracle GoldenGate selection is conceptually similar to the WHERE clause found on SQL. The purpose is to limit data replication to specific criteria. The selection can be developed as part of the primary extract, data pump extract or replicate process. The



following illustrates data selection on the CUSTOMERS and POLICIES tables for customers with policy no greater than 100000 through the primary extract ES1E201.

```
EXTRACT es1e201
SETENV (ORACLE_SID=S1E2)
SETENV (ORACLE_HOME="/u01/app/oracle/product/11.2.0/dbhome_1")
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@S1E2, PASSWORD oracle
EXTTRAIL ./dirdat/sa
TABLE osm$repapi.customers, WHERE (cust_no > 100000);
TABLE osm$repapi.policies, WHERE (cust_no > 100000);
```

The selection clause should be defined for all related tables to avoid referential integrity violation when applying the transactions to the target database. Hence, it's necessary to construct the where clause to the CUSTOMERS table and POLICIES table as well.

## Data filtering

Data filtering is suitable for complex data selection by utilizing Oracle GoldenGate built-in functions. For the replicate process, the FILTER parameter is developed as part of the MAP statement.

The example below illustrates the use of the FILTER parameter with the function @STRFIND to filter out email addresses that does not contains the character '@'.

```
REPLICAT rs1e201
SETENV (ORACLE_SID=T1C1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@T1C1, PASSWORD oracle
DISCARDFILE ./dirrpt/rs1e201.dsc, PURGE
SOURCEDEFS ./dirdef/oracle.def
MAP osm$repapi.customers, TARGET osm$repapi.customers, &
COLMAP (USEDEFAULTS,
        cust_addr = cust_address),
FILTER (@STRFIND (cust_email, "@") > 0);
MAP osm$repapi.policies, TARGET osm$repapi.policies;
```

## Data transformation

The need to transform tables columns values is very essentials for implementing an online operational data warehouse repository for business intelligence applications.

Data transformation predominately applies to:

- Heterogeneous environment requirements
- The target database application uses rigid data types of different format
- Data warehouse repository summaries

For more complex operational data store (ODS), one form of data warehouse repository, data transformation is often essential. The transformation may be configured as part of the primary extract, secondary extract, the delivery process or a combination of any of them. Transformations achieved using Oracle GoldenGate offload the database and enable transparent application support . A simple yet common transformation example is the conversion of number data type to formatted character string data type as shown below:

Source table column data type: NUMBER  
Source input value: 5085797993

Target table column data type: VARCHAR2(16)  
Target output value: (508) 579-7993

Nesting Oracle GoldenGate built-in functions @STRCAT, @STREXT and @STRNUM convert the number to a formatted string. Starting with @STRNUM to convert the number to string, @STREXT to extract substrings of input string and then use @STRCAT to concatenate the substrings.

This data transformation is developed as part of replicate process configuration. Because of mismatched column names for the customer address, column mapping is also included.

```

REPLICAT rs1e201
SETENV (ORACLE_SID=T1C1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@T1C1, PASSWORD oracle
DISCARDFILE ./dirrpt/rs1e201.dsc, PURGE
SOURCEDEFS ./dirdef/oracle.def
MAP osm$repapi.customers, TARGET osm$repapi.customers, &
COLMAP (USEDEFAULTS,
        cust_addr = cust_address,
        cust_mobile = @STRCAT('(',
        @STREXT(@STRNUM(cust_mobile), 1, 3), ')',
        ' ',
        @STREXT(@STRNUM(cust_mobile), 4, 6),
        '-',
        @STREXT(@STRNUM(cust_mobile), 7, 10)));
MAP osm$repapi.policies, TARGET osm$repapi.policies;

```

# Tokens

Oracle GoldenGate tokens are attributes to stores environment variable values, which are available on the header of trail record header. Using Oracle GoldenGate built-in function @TOKEN within the parameter files retrieves the token value.

Table 1-6 lists pre-defined Oracle GoldenGate tokens.

| Token Name   | Description                                                |
|--------------|------------------------------------------------------------|
| TK_HOST      | Stores the host name                                       |
| TK_GROUP     | Stores the operating system group name                     |
| TK_OSUSER    | Stores the operating system user name                      |
| TK_DOMAIN    | Stores the hostname domain                                 |
| TK_COMMIT_TS | Stores the current commit transaction sequence number      |
| TK_POS       | Stores the current read position                           |
| TK_RBA       | Stores the relative byte address (RBA)                     |
| TK_TABLE     | Stores the table name                                      |
| TK_OPTYPE    | Stores the operation types (INSERT, UPDATE, DELETE or DDL) |

Table 1-6: Oracle GoldenGate pre-defined tokens

Another type of tokens are user-token. These are defined on the source system and moved to target system for event handling, providing extendibility for attributes that are not pre-defined by Oracle GoldenGate.

The replicate RS1E201 parameter file illustrates a basic implementation of token by associating each row with Oracle GoldenGate environment pre-defined tokens values that can be used by other applications, such as auditing.

```
T1C1>DESC customers
Name                                     Null?    Type
-----
CUST_NO                                NOT NULL NUMBER
CUST_NAME                              NOT NULL VARCHAR2(128)
CUST_EMAIL                             NOT NULL VARCHAR2(400)
CUST_MOBILE                             NOT NULL VARCHAR2(16)
CUST_ADDR                               VARCHAR2(128)
HOST                                   VARCHAR2(128)
GG_GROUP                               VARCHAR2(128)
OSUSER                                 VARCHAR2(128)
DOMAIN                                 VARCHAR2(128)
BA_IND                                 VARCHAR2(128)
```

|                  |                       |
|------------------|-----------------------|
| <b>COMMIT_TS</b> | <b>VARCHAR2 (128)</b> |
| <b>POS</b>       | <b>VARCHAR2 (128)</b> |
| <b>RBA</b>       | <b>VARCHAR2 (128)</b> |
| <b>TABlename</b> | <b>VARCHAR2 (128)</b> |
| <b>OPTYPE</b>    | <b>VARCHAR2 (128)</b> |

```

REPLICAT rs1e201
SETENV (ORACLE_SID=T1C1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@T1C1, PASSWORD oracle
DISCARDFILE ./dirrpt/rs1e201.dsc, PURGE
SOURCEDEFS ./dirdef/oracle.def
MAP osm$repapi.customers, TARGET osm$repapi.customers, &
COLMAP (USEDEFAULTS,
        cust_addr = cust_address,
        host      = @TOKEN ('TK_HOST'),
        gg_group  = @TOKEN ('TK_GROUP'),
        osuser    = @TOKEN ('TK_OSUSER'),
        domain    = @TOKEN ('TK_DOMAIN'),
        ba_ind    = @TOKEN ('TK_BA_IND'),
        commit_ts = @TOKEN ('TK_COMMIT_TS'),
        pos       = @TOKEN ('TK_POS'),
        rba       = @TOKEN ('TK_RBA'),
        tablename = @TOKEN ('TK_TABLE'),
        optype    = @TOKEN ('TK_OPTYPE'));
MAP osm$repapi.policies, TARGET osm$repapi.policies;

```

Refer to chapter 10 to learn about advanced implementation of how to store environment variable into Oracle database instance memory, SGA.

## Programs call-out

Oracle GoldenGate extends its functionalities through call-out to database stored procedures and C programs. For database stored procedures, the call-out enabled through SQLEXEC parameter, call-out to C programs developed using user-exist.

To invoke a database stored procedures, extends the TARGET clause to include the SQLEXEC option. The parameter SPNAME specifies the <owner>.<procedure name> and PARAMS specifies the list of formal parameters.

```

MAP osm$repapi.customers, TARGET osm$repapi.customers, &
COLMAP (USEDEFAULTS,
        cust_addr = cust_address),
SQLEXEC (SPNAME osm$repapi.load_cust_pic,

```

```
PARAMS (pcust_no = cust_no));
```

Refer to chapter 10 for an end-to-end demonstration on how to invoke database stored procedure from within Oracle GoldenGate. The demonstration uses of DBMS\_FILE\_TRANSFER Oracle database built-in package to enable moving an image file of type BFILE from source system to target system. The purpose of the demonstration to show the technique for handling unsupported data types such as BFILE.

## Configuration options

Post-deployment of Oracle GoldenGate instance often requires environment enhancements. This section is limited to the basic enhancements only.

We will discuss enabling the following:

- Secure parameter files
- Data compression
- Scheduler
- Batch SQL

## Secure parameter files

Extract and replicate processes parameter files often requires login to the database. Best practices should avoid using clear-text password, instead use encrypted password. Two methods are available: default method and named method.

We will illustrate using the default method to encrypt and use the encrypted password.

- From the GGSCI command-line, run the ENCRYPT command using the default algorithm, BLOWFISH.

```
GGSCI (ggs-source) 34> ENCRYPT PASSWORD oracle, ENCRYPTKEY default
```

```
Using Blowfish encryption with DEFAULT key.
```

```
Encrypted password: AACAAAAAAAAAAAAAGAIFAAUDVHCFUGFIYF
```

```
Algorithm used: BLOWFISH
```

```
GGSCI (ggs-source) 35>
```

- Test the encrypted password

```
GGSCI (ggs-source) 39> DBLOGIN USERID ggs_admin@S1E2,
PASSWORD AACAAAAAAAAAAGAIFAAUDVHCFUGFIYF, ENCRYPTKEY
DEFAULT
```

Successfully logged into database.

```
GGSCI (ggs-source) 36>
```

- Copy the encrypted password into the extract and replicate parameter files.

```
EXTRACT es1e201
SETENV (ORACLE_SID=S1E2)
SETENV (ORACLE_HOME="/u01/app/oracle/product/11.2.0/dbhome_1")
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@S1E2, PASSWORD
AACAAAAAAAAAAGAIFAAUDVHCFUGFIYF, ENCRYPTKEY DEFAULT
EXTTRAIL ./dirdat/sa
TABLE osm$repapi.customers, WHERE (cust_no > 100000);
TABLE osm$repapi.policies, WHERE (cust_no > 100000);
```

## Data compression

Moving data between source and target systems highly benefits from trail files compression, the default is non-compressed. Compression is transparent and enabled by adding the compression parameter option to the data pump process.

The following illustrates using Oracle GoldenGate compression option of type zlib. Automatic decompression is performed by the data collector server.

```
EXTRACT epmp01
USERID ggs_admin@S1E2, PASSWORD oracle
PASSTHRU
RMTHOST localhost, MGRPORT 7810, COMPRESS, &
COMPRESSTHRESHOLD 512
RMTRAIL /u01/app/oracle/ggs_2/dirdat/ta
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
```

## Job scheduler

Oracle GoldenGate has its own job scheduler sub-system. It enables executing SQL statements and PL/SQL stored sub-programs. However, for complex event scheduling, consider using the database scheduler DBMS\_SCHEDULER API. Oracle GoldenGate scheduler uses simplified expression to trigger the jobs.

An application area for using Oracle GoldenGate scheduler is to implement self-monitoring tool by scheduling the replicate processes to send an email every 10 minutes to inform the Oracle GoldenGate administrator it's on RUNNING status.

```
SQL> CREATE PROCEDURE email_rep_status(p1 VARCHAR2) IS
  2 BEGIN
  3     smtp_send(p1||' IS RUNNING...');
  4 END;
  5 /
```

Procedure created.

SQL>

Now, add the parameter SQLEXEC to each replicate process parameter file. Stop and restart replicate processes.

```
REPLICAT rs1e201
SETENV (ORACLE_SID=T1C1
)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@T1C1, PASSWORD oracle
SQLEXEC "CALL email_rep_status('RS1E201')" EVERY 10 MINUTES
DISCARDFILE ./dirrpt/rs1e201.dsc, PURGE
SOURCEDEFS ./dirdef/oracle.def
MAP osm$repapi.customers, TARGET osm$repapi.customers, &
COLMAP (USEDEFAULTS,
        cust_addr = cust_address),
SQLEXEC (SPNAME osm$repapi.load_cust_pic,
        PARAMS (pcust_no = cust_no));
MAP osm$repapi.policies, TARGET osm$repapi.policies;
```

An informational message is sent to the process report file whenever every time the statement executed. Nice and simple tool!

```
2013-07-14 22:23:25 INFO      OGG-00888  SQL statement
executed successfully on interval.
```

## Batch SQL

The replicate process parameter BATCHSQL increases the apply throughput by classifying and building in-memory arrays for similar SQL statements to be applied as one database operation.

A typical environment is an OLTP system that applies transactions differing on bind variable values only. Oracle GoldenGate supercharges the replicate processes using batch SQL technique. By default, replicate processes apply transactions one at a time, with BATCHSQL the replicate process applies transactions on batch mode.

The replicate process parameter BATCHSQL follows.

```
REPLICAT rs1e201
SETENV (ORACLE_SID=T1C1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@T1C1, PASSWORD oracle
BATCHSQL
DISCARDFILE ./dirrpt/rs1e201.dsc, PURGE
SOURCEDEFS ./dirdef/oracle.def
MAP osm$repapi.customers, TARGET osm$repapi.customers, &
COLMAP (USEDEFAULTS,
        cust_addr = cust_address);
MAP osm$repapi.policies, TARGET osm$repapi.policies;
```

Refer to chapter 13 to learn more about BATCHSQL settings.

## Automatic storage management (ASM) support

Oracle Database 10g release 2 introduced Oracle automatic storage management (ASM), since then Oracle corporation has been enhancing ASM to provides high-performance file system management supporting all Oracle database file types. Figure 1-28 depicts Oracle ASM instance showing the data flow from Oracle GoldenGate extract on an ASM environment.

Oracle GoldenGate change data capture requires ASM authentication for accessing the transaction logs. The extract parameter TRANLOGOPTIONS handles authentication to the ASM instance.

```
TRANLOGOPTIONS ASMUSER SYS@<ASM_instance>, ASMPASSWORD
<password> [<algorithm> ENCRYPTKEY {<keyname> | DEFAULT}]
```

The parameter TRANLOGOPTIONS is database type dependent, and may appear several times within the same parameter file.



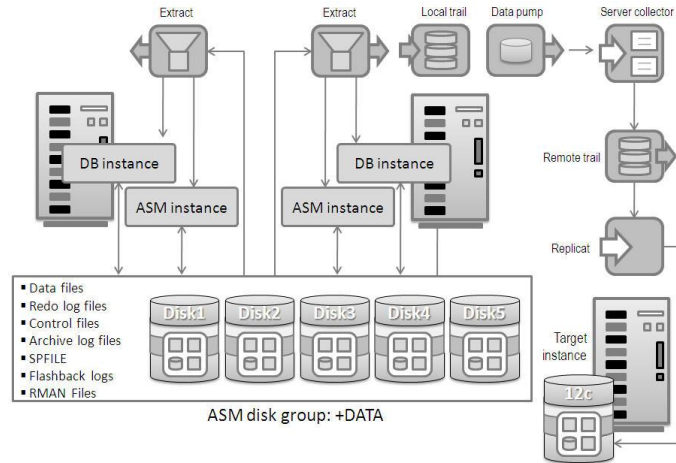


Figure 1-28: Oracle ASM for GoldenGate

To enable Oracle GoldenGate capture from an ASM instance transaction logs, it requires configuring the network for the ASM instance. The steps to prepare Oracle GoldenGate for an ASM instance follows.

Step 1: Ensure the database listener is configured to handle connection, and the TNS alias is configured. The network configuration is for the grid infrastructure network.

```
SID_LIST_LISTENER =
(
  SID_LIST =
  (
    SID_DESC =
      (GLOBAL_DBNAME = S2E2.precisetrace.com)
      (ORACLE_HOME = /u01/app/oracle/product/11.2.0/dbhome_1)
      (SID_NAME = S2E2)
    )
  (SID_DESC =
      (ORACLE_HOME = /u01/app/oracle/product/11.2.0/grid)
      (SID = +ASM)
    )
  )
)

LISTENER =
(
  DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = ggs-source) (PORT = 1521))
  )
)

ADR_BASE_LISTENER = /u01/app/oracle

ASM =
(
  DESCRIPTION =
    (ADDRESS_LIST =
```

```

        (ADDRESS = (PROTOCOL = TCP) (HOST = ggs-source) (PORT = 1521))
    )
    (CONNECT_DATA =
        (SID = +ASM)
    )
)

```

Step 2: Test connectivity using SQL\*Plus as SYSASM users. The test must be successful before proceeding to step 3.

```

$ . oraenv
ORACLE_SID = [S2E2] ? +ASM
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus /nolog

SQL*Plus: Release 11.2.0.3.0 Production on Fri May 31 11:08:35 2013

Copyright (c) 1982, 2011, Oracle. All rights reserved.

SQL> conn / as SYSASM
Connected.
SQL>

```

Step 3: Modify the primary extract parameter file to include the TRANSLOGOPTIONS parameter.

```

EXTRACT es1e201
SETENV (ORACLE_SID=S1E2)
SETENV (ORACLE_HOME="/u01/app/oracle/product/11.2.0/dbhome_1")
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@S1E2, PASSWORD AACAAAAAAAAAAGAIFAAUDVHCUGFIYF,
ENCRYPTKEY DEFAULT
TRANLOGOPTIONS ASMUSER sys@ASM, ASMPASSWORD oracle_4U
EXTTRAIL ./dirdat/sa
TABLE osm$repapi.customers, WHERE (cust_no > 100000);
TABLE osm$repapi.policies, WHERE (cust_no > 100000);

```

When using Oracle GoldenGate on an ASM instance, ensure the maximum number of concurrently processes parameter is sufficient to handle the addition of GoldenGate extracts, default value is 50. When using Oracle GoldenGate parallel processing, set the value to support the total number of processes defined by the @RANGE function. A typical parameter value setting is 300.

## Parallel processing

Oracle GoldenGate supports high-performance parallel processing. The degree of parallelism controlled using the @RANGE function. Figure 1-29 delivers 1-3-6 degree of parallelism. The extract ES1E201 perform change data capture from the list

of tables. The data pump extract groups EPMP01, EPMP02, and EPMP03 parallelize extracting from the local trail `./dirdat/sa` into degree of 3, each data pump extract has its associated data server collector, which in turn re-assembles the data and writes to its own remote trail. Ultimately, using parallel processing maintains stable sub-second latency and lagging.

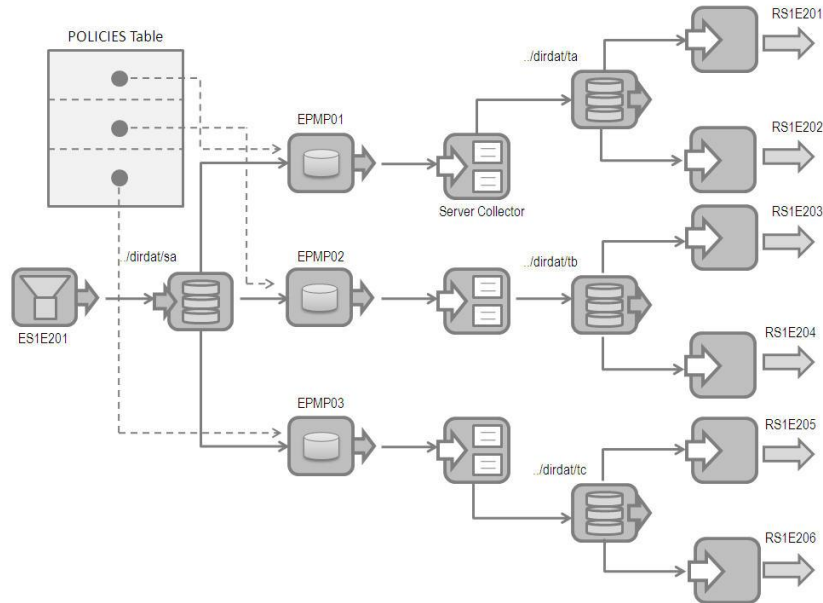


*Figure 1-29: Oracle GoldenGate Parallel processing*

The replicate groups RS1E201, RS1E202, RS1E203, RS1E204, RS1E205 and RS1E206 parallelize its associated remote trail extract into degree of 2, hence a total of 6 replicates processes are used to apply the transaction, captured by the extract ES1E201, to the target database.

## Implement parallelism

Parallel processing is a key for maximizing system scalability and throughput. Oracle GoldenGate built-in function `@RANGE` implemented to deploy parallel processing for increasing throughput on large volume transaction tables for online synchronization, keeping the lagging minimum.

Specify each range in a `FILTER` clause in a `TABLE` or `MAP` statement. `@RANGE` function is very similar to hash database partitioning, it's safe and scalable preserving data integrity by guaranteeing that the row is processed by related process groups.

@RANGE computes a hash value of the columns specified in the input. If no columns are specified, the KEYCOLS clause of the TABLE or MAP statement is used to determine the columns to hash, if a KEYCOLS clause exists. Otherwise, the primary key columns are used, which the default.

The function takes three arguments, the current process number, the degree of parallelism and partitioning column(s), which optional. The general syntax:

```
@RANGE (<range>, <total ranges> [, <column>] [, <column>] [, ...])
```

## Memory resource requirement

Prior to deploying high-degree parallel processing, calculate and provision the memory requirements for all processes. The required memory size is dynamic and depends on the transaction volume processed by each process. Table 1-x shows the calculation for the above parallel processing design using the operating system Linux process status (ps) and process mapping (pmap) commands. Table 1-x illustrates the memory consumption for the above design while the source and target system engaged on real-time data replication for a low volume transaction.

Start by determining the Linux process ID, then use the process for on pmap (process map) operating system command.

```
GGSCI (ggs-target) 28> SHELL ps -edf | grep RS1E201

oracle      6427   6584    0 05:25 pts/3      00:00:00 sh -c ps -edf |
grep RS1E201
oracle      6429   6427    0 05:25 pts/3      00:00:00 grep RS1E201
oracle      8943   4964    0 04:22 ?          00:00:03
/u01/app/oracle/ggs/replicat PARAMFILE
/u01/app/oracle/ggs/dirprm/rs1e201.prm REPORTFILE
/u01/app/oracle/ggs/dirrpt/RS1E201.rpt PROCESSID RS1E201 USESUBDIRS

GGSCI (ggs-target) 29> shell pmap -x 6427 | grep total

total kB   204252      -      -      -

GGSCI (ggs-target) 30>
```

Follow the above syntax to calculate the memory for all processes as shown in table 1-7.

| GoldenGate process<br>Name | GoldenGate process<br>Type | Memory<br>size (KB) |
|----------------------------|----------------------------|---------------------|
|----------------------------|----------------------------|---------------------|

|                   |                           |             |
|-------------------|---------------------------|-------------|
| ES1E201           | Primary extract           | 204252      |
| EPMP01            | Second extract, data pump | 112396      |
| EPMP02            | Second extract, data pump | 111808      |
| EPMP03            | Second extract, data pump | 111812      |
| RS1E201           | Replicate process         | 131476      |
| RS1E202           | Replicate process         | 131476      |
| RS1E203           | Replicate process         | 131128      |
| RS1E204           | Replicate process         | 131124      |
| RS1E205           | Replicate process         | 131128      |
| RS1E206           | Replicate process         | 131476      |
| <b>Total (GB)</b> |                           | <b>1.27</b> |

*Table 1-7: Oracle GoldenGate process memory utilization*

## Parallel processing implementation

The following shows the parameter files and the GGSCI ADD EXTRACT, ADD EXTTRAIL and ADD RMTTRAIL commands executed to deploy the above parallel processing design shown in figure 1-30.

The source system setup creates the primary extract group ES1E201 and the secondary extract groups EPMP01, EPMP02 and EPMP03. The parameter files are listed below. Because of the FILTER clause, the parameter NOPASSTHRU is required.

### -- es1e201.prm

```
EXTRACT es1e201
SETENV (ORACLE_SID=S1E2)
SETENV (ORACLE_HOME="/u01/app/oracle/product/11.2.0/dbhome_1")
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@S1E2, PASSWORD AACAAAAAAAAAAGAIFAAUDVHCFUGFIYF,
ENCRYPTKEY DEFAULT
EXTTRAIL ./dirdat/sa
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
```

### -- epmp01.prm

```
EXTRACT epmp01
USERID ggs_admin@S1E2, PASSWORD AACAAAAAAAAAAGAIFAAUDVHCFUGFIYF,
ENCRYPTKEY DEFAULT
NOPASSTHRU
RMTHOST ggs-target, MGRPORT 7812
RMTTRAIL ./dirdat/ta
TABLE osm$repapi.customers, FILTER (@RANGE(1, 3));
TABLE osm$repapi.policies, FILTER (@RANGE(1, 3));
```

```
-- epmp02.prm
```

```
EXTRACT epmp02
USERID ggs_admin@S1E2, PASSWORD AACAAAAAAAAAAGAIFAAUDVHCFUGFIYF,
ENCRYPTKEY DEFAULT
NOPASSTHRU
RMTHOST ggs-target, MGRPORT 7812
RMTTRAIL ./dirdat/tb
TABLE osm$repapi.customers, FILTER (@RANGE(2, 3));
TABLE osm$repapi.policies, FILTER (@RANGE(2, 3));
```

```
-- epmp03.prm
```

```
EXTRACT epmp03
USERID ggs_admin@S1E2, PASSWORD AACAAAAAAAAAAGAIFAAUDVHCFUGFIYF,
ENCRYPTKEY DEFAULT
NOPASSTHRU
RMTHOST ggs-target, MGRPORT 7812
RMTTRAIL ./dirdat/tc
TABLE osm$repapi.customers, FILTER (@RANGE(3, 3));
TABLE osm$repapi.policies, FILTER (@RANGE(3, 3));
```

Now, run the ADD EXTRACT, ADD EXT\*TRAIL and ADD RMT\*TRAIL commands.

```
GGSCI (ggs-source) 5> ADD EXTRACT es1e201, TRANLOG, BEGIN
now, THREADS 1
EXTRACT added.
```

```
GGSCI (ggs-source) 6> ADD EXTTRAIL ./dirdat/sa, EXTRACT
es1e201, MEGABYTES 32
EXTTRAIL added.
```

```
GGSCI (ggs-source) 7> ADD EXTRACT epmp01, EXTTRAILSOURCE
./dirdat/sa
EXTRACT added.
```

```
GGSCI (ggs-source) 8> ADD RMTTRAIL ./dirdat/ta, EXTRACT
epmp01, MEGABYTES 32
RMTTRAIL added.
```

```
GGSCI (ggs-source) 9> ADD EXTRACT epmp02, EXTTRAILSOURCE
./dirdat/sa
EXTRACT added.
```

```
GGSCI (ggs-source) 10> ADD RMTTRAIL ./dirdat/tb, EXTRACT
epmp02, MEGABYTES 32
RMTTRAIL added.
```

```

GGSCI (ggs-source) 11> ADD EXTRACT epmp03, EXTTRAILSOURCE
./dirdat/sa
EXTRACT added.

GGSCI (ggs-source) 12> ADD RMTTRAIL ./dirdat/tc, EXTRACT
epmp03, MEGABYTES 32
RMTTRAIL added.

```

The target system setup creates the replicate process RS1E201, RS1E201, RS1E201, RS1E201, RS1E201 and RS1E201. The parameter files prepared to successfully start the replicate processes.

#### -- rs1e201.prm

```

REPLICAT rs1e201
SETENV (ORACLE_SID=T1C1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@tlc1, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rs1e201.dsc, PURGE
MAP osm$repapi.customers, TARGET osm$repapi.customers,
FILTER (@RANGE(1, 2));
MAP osm$repapi.policies, TARGET osm$repapi.policies,
FILTER (@RANGE(1, 2));

```

#### -- rs1e202.prm

```

EPLICAT rs1e202
SETENV (ORACLE_SID=T1C1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@tlc1, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rs1e202.dsc, PURGE
MAP osm$repapi.customers, TARGET osm$repapi.customers,
FILTER (@RANGE(2, 2));
MAP osm$repapi.policies, TARGET osm$repapi.policies,
FILTER (@RANGE(2, 2));

```

#### -- rs1e203.prm

```

REPLICAT rs1e203
SETENV (ORACLE_SID=T1C1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@tlc1, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rs1e203.dsc, PURGE
MAP osm$repapi.customers, TARGET osm$repapi.customers,
FILTER (@RANGE(1, 2));
MAP osm$repapi.policies, TARGET osm$repapi.policies,
FILTER (@RANGE(1, 2));

```

#### -- rs1e204.prm

```
REPLICAT rs1e204
SETENV (ORACLE_SID=T1C1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@t1c1, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rs1e204.dsc, PURGE
MAP osm$repapi.customers, TARGET osm$repapi.customers,
FILTER (@RANGE(2, 2));
MAP osm$repapi.policies, TARGET osm$repapi.policies,
FILTER (@RANGE(2, 2));
```

#### -- rs1e205.prm

```
REPLICAT rs1e205
SETENV (ORACLE_SID=T1C1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@t1c1, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rs1e205.dsc, PURGE
MAP osm$repapi.customers, TARGET osm$repapi.customers,
FILTER (@RANGE(1, 2));
MAP osm$repapi.policies, TARGET osm$repapi.policies,
FILTER (@RANGE(1, 2));
```

#### -- rs1e206.prm

```
REPLICAT rs1e206
SETENV (ORACLE_SID=T1C1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@t1c1, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rs1e206.dsc, PURGE
MAP osm$repapi.customers, TARGET osm$repapi.customers,
FILTER (@RANGE(2, 2));
MAP osm$repapi.policies, TARGET osm$repapi.policies,
FILTER (@RANGE(2, 2));
```

Let's start by creating the checkpoint table. This is the default checkpoint location for replicate processes. The ADD CHECKPOINTTABLE command requires successful login to the database.

```
GGSCI (ggs-target) 2> DBLOGIN USERID ggs_admin@t1c1, PASSWORD
oracle
Successfully logged into database.

GGSCI (ggs-target) 3> ADD CHECKPOINTTABLE
```



```
No checkpoint table specified. Using GLOBALS specification  
(ggs_admin.chkpttab)...
```

```
Successfully created checkpoint table ggs_admin.chkpttab.
```

```
GGSCI (ggs-target) 4>
```

Now, creating replicate processes RS1E201, RS1E202, RS1E203, RS1E204, RS1E205 and RS1E206.

```
GGSCI (ggs-target) 6> ADD REPLICAT rs1e201, EXTTRAIL  
./dirdat/ta  
REPLICAT added.
```

```
GGSCI (ggs-target) 7> ADD REPLICAT rs1e202, EXTTRAIL  
./dirdat/ta  
REPLICAT added.
```

```
GGSCI (ggs-target) 8> ADD REPLICAT rs1e203, EXTTRAIL  
./dirdat/tb  
REPLICAT added.
```

```
GGSCI (ggs-target) 9> ADD REPLICAT rs1e204, EXTTRAIL  
./dirdat/tb  
REPLICAT added.
```

```
GGSCI (ggs-target) 10> ADD REPLICAT rs1e205, EXTTRAIL  
./dirdat/tc  
REPLICAT added.
```

```
GGSCI (ggs-target) 11> ADD REPLICAT rs1e206, EXTTRAIL  
./dirdat/tc  
REPLICAT added.
```

Start and verify the source system processes status

```
GGSCI (ggs-source) 40> START er *
```

```
Sending START request to MANAGER ...  
EXTRACT EPMP01 starting
```

```
Sending START request to MANAGER ...  
EXTRACT EPMP02 starting
```

```
Sending START request to MANAGER ...  
EXTRACT EPMP03 starting
```

```
Sending START request to MANAGER ...
EXTRACT ES1E201 starting
```

```
GGSCI (ggs-source) 41> INFO ALL
```

| Program | Status  | Group   | Lag at Chkpt | Time Since Chkpt |
|---------|---------|---------|--------------|------------------|
| MANAGER | RUNNING |         |              |                  |
| EXTRACT | RUNNING | EPMP01  | 00:00:00     | 00:00:12         |
| EXTRACT | RUNNING | EPMP02  | 00:00:00     | 00:00:11         |
| EXTRACT | RUNNING | EPMP03  | 00:00:00     | 00:00:10         |
| EXTRACT | RUNNING | ES1E201 | 00:00:00     | 00:00:09         |

```
GGSCI (ggs-source) 43>
```

Start and verify the target system processes status.

```
GGSCI (ggs-target) 11> START ER *
```

```
Sending START request to MANAGER ...
REPLICAT RS1E201 starting
```

```
Sending START request to MANAGER ...
REPLICAT RS1E202 starting
```

```
Sending START request to MANAGER ...
REPLICAT RS1E203 starting
```

```
Sending START request to MANAGER ...
REPLICAT RS1E204 starting
```

```
Sending START request to MANAGER ...
REPLICAT RS1E205 starting
```

```
Sending START request to MANAGER ...
REPLICAT RS1E206 starting
```

```
GGSCI (ggs-target) 12> INFO ALL
```

| Program  | Status  | Group   | Lag at Chkpt | Time Since Chkpt |
|----------|---------|---------|--------------|------------------|
| MANAGER  | RUNNING |         |              |                  |
| REPLICAT | RUNNING | RS1E201 | 00:00:00     | 00:00:09         |
| REPLICAT | RUNNING | RS1E202 | 00:00:00     | 00:00:09         |
| REPLICAT | RUNNING | RS1E203 | 00:00:00     | 00:00:08         |
| REPLICAT | RUNNING | RS1E204 | 00:00:00     | 00:00:08         |
| REPLICAT | RUNNING | RS1E205 | 00:00:00     | 00:00:08         |
| REPLICAT | RUNNING | RS1E206 | 00:00:00     | 00:00:08         |

## Generating Reports

Oracle GoldenGate generates various types of reports for troubleshooting and monitoring purposes. These report are available from GGSCI command-line.

Use the view GGSCI command to view process report.

```
GGSCI (ggs-target) 13> VIEW REPORT rs1e201
```

GGSCI STATS command reports process statistics. For example, display the @RANGE statistics using wild-card character. The STATS indicates parallel processing has been distributed almost evenly across the replicate processes.

```
GGSCI (ggs-target) 15> STATS REPLICAT rs1e2*
```

Sending STATS request to REPLICAT RS1E201 ...

Start of Statistics at 2013-11-05 08:52:00.

Replicating from OSM\$REPAPI.CUSTOMERS to OSM\$REPAPI.CUSTOMERS:

```
*** Total statistics since 2013-11-05 08:23:19 ***
Total inserts                19494.00
Total updates                 0.00
Total deletes                 0.00
Total discards                0.00
Total operations              19494.00
```

```
*** Daily statistics since 2013-11-05 08:23:19 ***
Total inserts                19494.00
Total updates                 0.00
Total deletes                 0.00
Total discards                0.00
Total operations              19494.00
```

```
*** Hourly statistics since 2013-11-05 08:23:19 ***
Total inserts                19494.00
Total updates                 0.00
Total deletes                 0.00
Total discards                0.00
Total operations              19494.00
```

```
*** Latest statistics since 2013-11-05 08:23:19 ***
Total inserts                19494.00
Total updates                 0.00
Total deletes                 0.00
```

|                  |          |
|------------------|----------|
| Total discards   | 0.00     |
| Total operations | 19494.00 |

## Monitoring processes lagging

The difference between the commit and the apply determines the amount of lagging. The following show the lag for replicate process RS1E201.

```
GGSCI (ggs-target) 16> LAG REPLICAT rs1e201
```

```
Sending GETLAG request to REPLICAT RS1E201 ...
```

```
Last record lag: 12 seconds.
```

```
At EOF, no more records to process.
```

The column Lag at Chkpt column of the INFO ALL command display the lagging of the replicate process.

```
GGSCI (ggs-target) 17> INFO ALL
```

| Program  | Status  | Group   | Lag at Chkpt | Time Since Chkpt |
|----------|---------|---------|--------------|------------------|
| MANAGER  | RUNNING |         |              |                  |
| REPLICAT | RUNNING | RS1E201 | 00:00:00     | 00:00:07         |
| REPLICAT | RUNNING | RS1E202 | 00:00:00     | 00:00:07         |
| REPLICAT | RUNNING | RS1E203 | 00:00:00     | 00:00:06         |
| REPLICAT | RUNNING | RS1E204 | 00:00:00     | 00:00:06         |
| REPLICAT | RUNNING | RS1E205 | 00:00:00     | 00:00:06         |
| REPLICAT | RUNNING | RS1E206 | 00:00:00     | 00:00:06         |

```
GGSCI (ggs-target) 18>
```

## Events logging system

Oracle GoldenGate is integrated with Linux/UNIX syslog messaging system and Windows Event Viewer for events logging. There are three files for managing runtime errors, namely: ggserr.log, tcperr and ggMessage.dat all located on Oracle GoldenGate home directory. The file ggserr.log is for logging currently running instance activities, tcperr defines rules for handling TCP/IP errors and actions and ggMessage.dat is an obfuscated data file for storing Oracle GoldenGate error messages specific to the instance version.

By default Oracle GoldenGate writes all levels of messages to ggserr.log file, which is crucial for newly deployed instance for monitoring. After the instance runs operational and highly stable, logging level can be adjusted to filter out less important messages send to the log file. The logging level is determined by Oracle GoldenGate SYSLOG parameter. Setting SYSLOG at GLOBALS parameter file allows Oracle

GoldenGate to write and maybe filter out messages for all instance processes, setting SYSLOG at mgr.prm parameter files allows Oracle GoldenGate to write and maybe filter out the manager process related messages only. Setting SYSLOG on both locations causes the manager to override the GLOBALS setting for the manager process, this allows the use of separate settings between the manager and other processes. SYSLOG parameter takes the following arguments for level settings.

## **ALL**

Writes all Oracle GoldenGate messages. This setting is the same as combining INFO, WARN, ERROR arguments. This is the default level of logging messages. This argument cannot be combined with other arguments.

## **NONE**

Suppresses all Oracle GoldenGate messages. This is not a recommended setting and it should be used for non-production environment only. This argument cannot be combined with other arguments.

## **INFO**

Writes messages of type INFO (information) only. This argument can be combined with WARN (warning) and ERROR arguments. Information messages include activities such as stopping and starting extracts and replicates processes.

## **WARN**

Write messages of type WARN (warning). This argument can be combined with INFO (information) and ERROR arguments.

## **ERROR**

Write messages of type ERROR. This argument can be combined with INFO (information) and WARN (warning) arguments.

Because the log file ggserr.log is not cyclic, periodically archiving the log file is recommend to avoid the file growing linearly. When ggserr.log is deleted, Oracle GoldenGate re-creates the logging file ggserr.log once it's trying to write the very next message. The following Linux/UNIX shell program archives ggserr.log file to /tmp directory when ggserr.log exceeds 10MB (10485760 bytes).

```
#-----  
#--      DISCLAIMER:
```

```

#--
#--      This script is provided for educational purposes only.
#--      The script has been tested on an Oracle GoldenGate instance
#--      and run as intended. However; prior to using the script on
#--      production systems, a dynamic test should be conducted prior
#--      to deployment.
#--
#--      Copyright (c) Rampant TechPress. 2014. All Rights Reserved.
#-----
#
#!/bin/bash
GGS_HOME="/u01/app/ogg/11.2.0"
if [ -f $GGS_HOME/ggserr.log ]; then
    FS=$( stat -c %s $GGS_HOME/ggserr.log)
    if [ $FS -gt 10485760 ]; then
        mv $GGS_HOME/ggserr.log /tmp
        echo "$GGS_HOME/ggserr.log was last archived on:" `date` >>
/tmp/ggserr.txt
    fi
else
    echo "$GGS_HOME/ggserr.log does not exist:" `date` >>
/tmp/ggserr.txt
fi

```

For scheduled archiving on Linux/UNIX platforms, save the above shell program to a file, for example ggserr.sh, set the execute permission, add the file ggserr.sh to /etc/crontab using the command crontab. However; Adapting Oracle GoldenGate monitor notification system enables immediate response to all types events.

The Syntax for SYSLOG parameter is shown next. The default setting is SYSLOG ALL, which writes INFO (information), WARN (warning) and ERROR messages.

```
SYSLOG {[ALL | NONE] | [, INFO] [, WARN] [, ERROR]}
```

## Windows Event Viewer

Oracle GoldenGate on MS Windows for SQL Server writes GoldenGate messages to ggserr.log file and to Windows Event Viewer, if enabled. The event levels is controlled by setting the parameter SYSLOG as explained above. To sort Oracle GoldenGate related messages, from Administrative Events click on the source column and scroll down to GGS ER.

Oracle GoldenGate is enabled to utilize Windows Event Viewer by running Oracle GoldenGate install program with the addevents option, this is normally performed after the installation. Then from Oracle GoldenGate home directory, copy the files category.dll and ggsmmsg.dll libraries to Windows SYSTEM32 directory for more

detailed Windows specific messages. Figure 1-30 shows an ERROR level message from Windows Event Viewer, the same message is written to ggserr.log file.

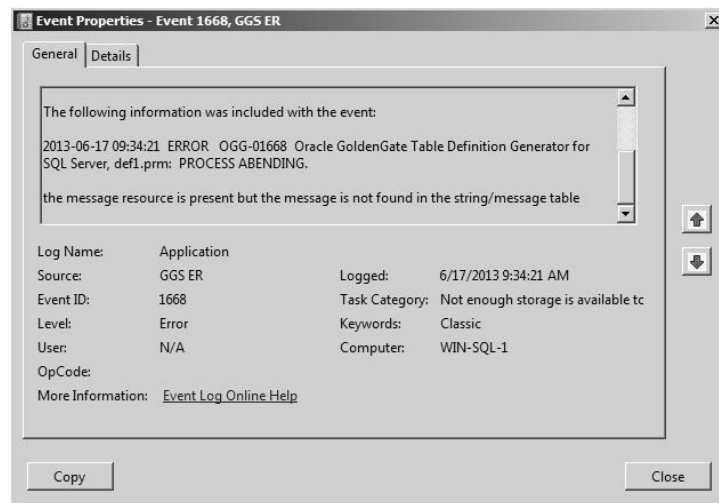


Figure 1-30: MS Windows event viewer for Oracle GoldenGate

## Windows events handling

MS Windows allows you to attach a task to an event level, providing an automated event handling for Oracle GoldenGate log messages. For example, events of level Information and Warning can be regularly cleared out to avoid the Event Viewer from growing quickly, events of level Error are associated with an action for immediate notification and action, as explained next.

Right-clicking on the event to start the Create Basic Task Wizard. Figure 1-31 associate an event of level Error with the action Send an email. Send an E-mail task uses the inputs: From, To, Subject, Text, Attachment, and SMTP server. An alternative to using the Create Basic Task Wizard is to create a customized archiving batch program to run using the Windows scheduler created by the AT command. AT command schedules operating system commands and batch programs (utilities) to run at a specific time and date, the computename argument refer to the local host (default) or remote host. Before using the AT command, Windows schedule service must be running.

Failure to manage ggserr.log file and Windows Event Viewer may result on:

- Performance related issues for large file size of ggserr.log
- Disk storage related issues for excessive events messaging

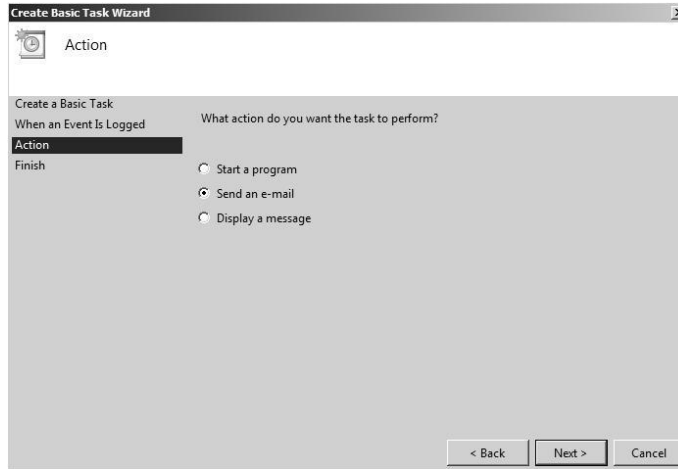


Figure 1-31: Managing and associating an event using an action

## TCP/IP network errors

Because Oracle GoldenGate uses TCP/IP network to communicate between source and target systems processes, TCP/IP network related errors response are adjustable to suite the network throughput. Modify TCP/IP values found at tcperr file to customizes Oracle GoldenGate TCP/IP related error message, such as time-out interval and dynamic port allocation. The following table shows the default TCP/IP error response settings found at tcperr file.

```
#
# TCP/IP error handling parameters
#
# Default error response is abend
#
```

| # Error       | Response | Delay (csecs) | Max Retries |
|---------------|----------|---------------|-------------|
| ECONNABORTED  | RETRY    | 1000          | 10          |
| #ECONNREFUSED | ABEND    | 0             | 0           |
| ECONNREFUSED  | RETRY    | 1000          | 12          |
| ECONNRESET    | RETRY    | 500           | 10          |
| ENETDOWN      | RETRY    | 3000          | 50          |
| ENETRESET     | RETRY    | 1000          | 10          |
| ENOBUFS       | RETRY    | 100           | 60          |
| ENOTCONN      | RETRY    | 100           | 10          |
| EPIPE         | RETRY    | 500           | 10          |
| ESHUTDOWN     | RETRY    | 1000          | 10          |
| ETIMEDOUT     | RETRY    | 1000          | 10          |
| NODYNPORTS    | RETRY    | 100           | 10          |



Carefully modify `tcpcrr`, and test using the same network throughput before deploying on production. As unexpected behavior may result due to incorrect settings of delay and retries values when moving from intranet for testing to internet for production TCP/IP network with firewall settings.

## Database design considerations

The majority of enterprise applications require data movement originating from heterogeneous databases, from within and outside of the organization. Because Oracle GoldenGate maintains transaction commit order of the source database, the source database layout influences the level of flexibility when implementing Oracle GoldenGate. Data movement is applications transparent eliminating any type of changes. The source system sustains minimum overhead as Oracle GoldenGate is optimized for change data capture from the transaction logs. However; Oracle GoldenGate implementers need to consider the following:

### Partitioned schemas

A partitioned schema is forms data model where the schema is self-contained, mostly found on specialized applications with limited number of related database tables. In such environment Oracle GoldenGate achieves the highest level of flexibility during the implementation and deliver sub-seconds latency for data movement, due to no transformation requirements. Partitioned schema helps eliminates data conflict when deploying Oracle GoldenGate on bidirectional topology for active-active environment. For example, in the case of using Oracle sequence as an artificial primary key value on partitioned schema design, associating the generated sequence number with a unique identifier such as the `INSTANCE_NAME` context attribute value always ensures a unique value across the source and target databases, thus avoiding data conflict on bidirectional replication as illustrated next.

For the source database (S1E2),

```
SQL> conn osm$repapi/oracle@S1E2
Connected.
SQL> CREATE SEQUENCE pol_no_seq;
```

Sequence created.

```
SQL> SELECT CONCAT(pol_no_seq.nextval,
2  CONCAT('-', SYS_CONTEXT('USERENV', 'INSTANCE_NAME')) ) POL_NO
3  FROM dual;
```

POL\_NO

```
-----  
7-S1E2
```

```
SQL>
```

For the target database (T1C1),

```
SQL> conn osm$repapi/oracle@T1C1
```

```
Connected.
```

```
SQL> CREATE SEQUENCE pol_no_seq;
```

```
Sequence created.
```

```
SQL> SELECT CONCAT(pol_no_seq.nextval,  
2  CONCAT('-', SYS_CONTEXT('USERENV', 'INSTANCE_NAME')) ) POL_NO  
3  FROM dual;
```

```
POL_NO  
-----
```

```
7-T1C1
```

```
SQL>
```

## Data types

Depending on the database type, certain data types are not supported. It always possible to migrate unsupported data types to similar functionality data types. For Oracle database, the list of unsupported data type is limited to: ANYDATA, ANYDATASET, ANYTYPE, BFILE, BINARY\_INTEGER, MLSLABEL, ORDDICOM, PLS\_INTEGER, TIMEZONE\_ABBR, URITYPE and UROWID.

For MS SQL Server, the list of unsupported data types is limited to: SQL\_Variant

## Referential integrity constraints

Oracle GoldenGate perform transactional change data capture from the transaction logs, preserving the transaction commit order. Even though the ordering of tables found at the extract parameter file is irrelevant. However; the ordering of tables found at the replicate parameter file requires proper listing. When the target database tables constraints are enabled, the order of tables must consider referential integrity constraints among tables, otherwise the replicate processes ABENDED due to constraint violation. This is a challenging task for complex database schemas definition when replicating the entire database.

## Three steps demonstration setup

The objectives of presenting an end-to-end configuration is to demonstrate how Oracle GoldenGate fundamental setup is concise and easy when used for as-is data movement such as homogenous replication, which absolutely true when compared with Oracle Steams configuration. The details of the steps are explained in throughout the book.

This fundamental setup can be used for the purpose of quick demonstration or as a reference for time-to-time GoldenGate users. Figure 1-32 shows the topology and the processes and trail files names used throughout the configuration. The source system is running Oracle Database 11g, while the target system is running Oracle Database 12c. The operating system for source and target systems is Oracle Enterprise Linux 5 update 6.

Refer to chapter 2 for details regarding downloading, installing and database pre-requisite of Oracle GoldenGate.

The components used throughout the configuration of the source and target systems are:

- Source database SID: S1E2
- Primary extract process: ES1E201
- Local trail: ./dirdat/sa
- Secondary extract (data pump) process: EPMP01
- Remote trail: ./dirdat/ta
- Replicate process: RS1E201
- Target database SID: T1C1

The unidirectional setup illustrated is using a data pump extract option in a non-ASM Oracle database environment. The database tables used for the demonstration listed below. These table are identical on both systems since no columns mapping required.

- OSM\$REPAPI.CUSTOMERS
- OSM\$REPAPI.POLICIES

Because of referential integrity enabled on the target system, the CUSTOMER is mapped before the POLICIES table. Failure to do so results on referential integrity violation returning an Oracle database error code ORA-02291 integrity constraint violation. In case the target database is for read-only database such as data warehouse, referential integrity constraints may remains disabled.

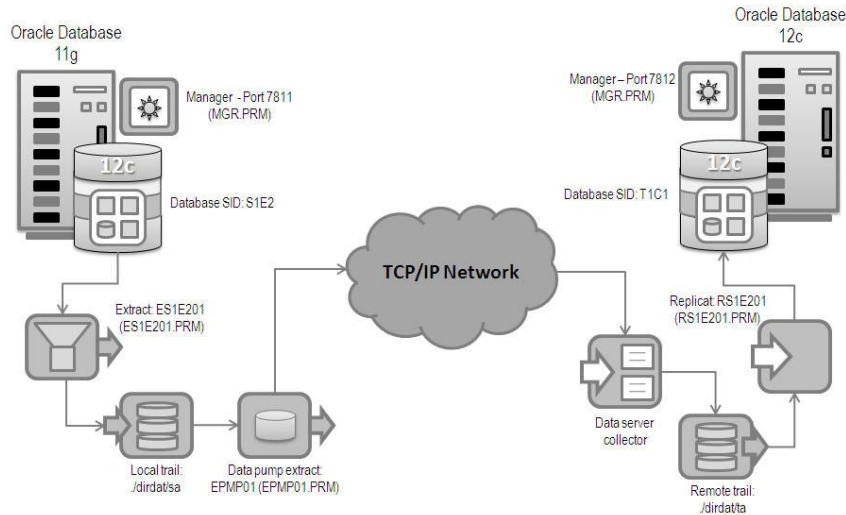


Figure 1-32: Fundamental setup components

## Step 1: Primary extract process configuration

The primary extract ES1E201 performs change data capture (CDC) from the transaction logs to keep the source and the target databases synchronized delivering sub-second latency. It's linked with a local trail 'sa' starting at sa000000 and recycling at sa999999. The runtime parameter file is ES1E201.PRM. The primary extract uses in-memory technique to capture currently running transactions, until it's committed and sent to the trail file, or rolled back and discarded by the primary extract. While the primary extract is running, a checkpoint process keep records it read and write position to the checkpoint file. In an event of failure, the primary extract avoid redundant capture of transactions already processed. Oracle GoldenGate uses its own commit sequence number (CSN) to guarantee the transaction integrity and data consistency when applied at the target database by the down-streaming replicate.

### -- Parameter file ES1E201.PRM

```
EXTRACT es1e201
SETENV (ORACLE_SID=S1E2)
SETENV (ORACLE_HOME="/u01/app/oracle/product/11.2.0/dbhome_1")
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@S1E2, PASSWORD AACAAAAAAGAIFAAUDVHCFUGFIYF,
ENCRYPTKEY DEFAULT
EXTTRAIL ./dirdat/sa
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
```

```
GGSCI (ggs-source) 7> ADD EXTRACT es1e201, TRANLOG, BEGIN
```

```
now, THREADS 1
```

```
EXTRACT added.
```

```
GGSCI (ggs-source) 8> ADD EXTTRAIL ./dirdat/sa, EXTRACT
```

```
es1e201, MEGABYTES 32
```

```
EXTTRAIL added.
```

## Step 2: Secondary extract process configuration (optional)

The secondary extract process is the data pump, EPMP01. It reads the local trail files then on real-time routes data to the remote data server collector. This data pump uses passthru option because source and target tables structure are identical. The runtime parameters RMTHOST and MGRPORT specify the remote system and Oracle GoldenGate manager listening port number. The remote trail files re-assembled by the data server collector are denoted by RMTTRAIL with the absolute or relative path followed by the name, a two characters prefix for the trail files name starting with ta000000 and recycling at ta999999.

```
-- Parameter file EPMP01.PRM
```

```
EXTRACT epmp01
```

```
USERID ggs_admin@s1e2, PASSWORD AACAAAAAAAAAAGAIFAAUDVHCFUGFIYF,
```

```
ENCRYPTKEY DEFAULT
```

```
PASSTHRU
```

```
RMTHOST ggs-target, MGRPORT 7812
```

```
RMTTRAIL ./dirdat/ta
```

```
TABLE osm$repapi.customers;
```

```
TABLE osm$repapi.policies;
```

```
GGSCI (ggs-source) 13> ADD EXTRACT epmp01, EXTTRAILSOURCE
```

```
./dirdat/sa
```

```
EXTRACT added.
```

```
GGSCI (ggs-source) 14> ADD RMTTRAIL ./dirdat/ta, EXTRACT
```

```
epmp01, MEGABYTES 32
```

```
RMTTRAIL added.
```

```
GGSCI (ggs-source) 15> info all
```

| Program | Status  | Group   | Lag      | Time Since Chkpt |
|---------|---------|---------|----------|------------------|
| MANAGER | RUNNING |         |          |                  |
| EXTRACT | STOPPED | EPMP01  | 00:00:00 | 00:02:27         |
| EXTRACT | STOPPED | ES1E201 | 00:00:00 | 00:08:29         |

```
GGSCI (ggs-source) 16> start extract *
```

```

Sending START request to MANAGER ...
EXTRACT EPMP01 starting

Sending START request to MANAGER ...
EXTRACT ES1E201 starting

GGSCI (ggs-source) 17> info all

```

| Program | Status  | Group   | Lag      | Time Since Chkpt |
|---------|---------|---------|----------|------------------|
| MANAGER | RUNNING |         |          |                  |
| EXTRACT | RUNNING | EPMP01  | 00:00:00 | 00:02:41         |
| EXTRACT | RUNNING | ES1E201 | 00:07:55 | 00:00:02         |

```

GGSCI (ggs-source) 18>

```

### Step 3: Replicate process configuration

The last down-streaming group of the configuration is the replicate process. The replicate process applies transactions to the target database with data consistency and integrity. The replicate relies on the commit sequence number (CSN) for ensuring transactions commit order of the source database. When the target database is solely for the purpose of data warehouse repository to support business intelligence application, then it's required to enable the primary key constraints and indexes, the referential integrity constraints should remain disabled for read-only application. However; when the target database is for load-balancing active-active bidirectional environment, referential integrity constraints must be enabled to ensure data consistency, data integrity and data conflict resolutions.

```
-- Parameter file RS1E201.PRM
```

```

REPLICAT rs1e201
SETENV (ORACLE_SID=T1C1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin@t1c1, PASSWORD oracle
handlecollisions
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rs1e201.dsc, PURGE
MAP osm$repapi.customers, TARGET osm$repapi.customers;
MAP osm$repapi.policies, TARGET osm$repapi.policies;

```

```

GGSCI (ggs-target) 6> ADD REPLICAT rs1e201, EXTTRAIL
./dirdat/ta

GGSCI (ggs-target) 7> info all

```

| Program  | Status  | Group   | Lag      | Time Since Chkpt |
|----------|---------|---------|----------|------------------|
| MANAGER  | RUNNING |         |          |                  |
| REPLICAT | STOPPED | RS1E201 | 00:00:00 | 00:00:05         |

```
GGSCI (ggs-target) 9> start replicat *
```

Sending START request to MANAGER ...  
REPLICAT RS1T201 starting

```
GGSCI (ggs-target) 10> info all
```

| Program  | Status  | Group   | Lag      | Time Since Chkpt |
|----------|---------|---------|----------|------------------|
| MANAGER  | RUNNING |         |          |                  |
| REPLICAT | RUNNING | RS1E201 | 00:00:00 | 00:00:00         |

## Oracle GoldenGate on multitenant container database

Deploying Oracle GoldenGate 12c on an Oracle Database 12c container database (CDB) inherits multitenant database operating requirements for pluggable database (PDB) as source and target databases. Figure 1-33 depicts Oracle GoldenGate for capture from T2C1PDB01 pluggable database.

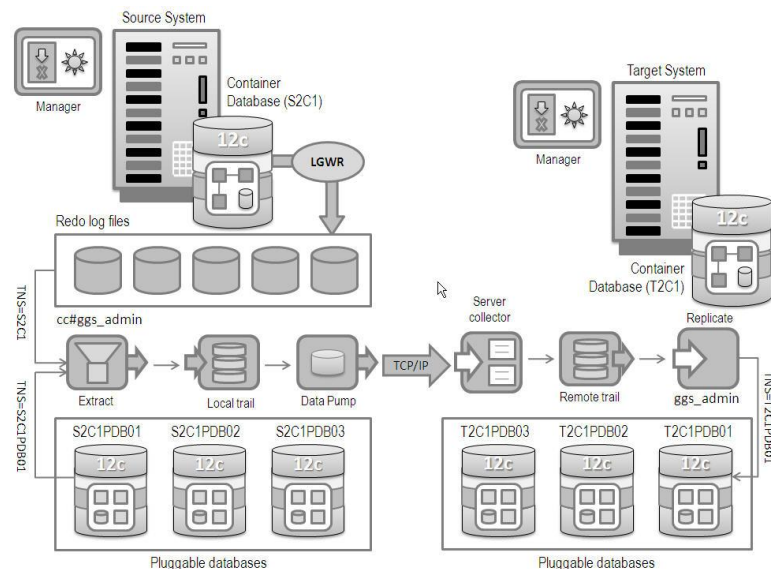


Figure 1-33: Pluggable database Support

## Service name

A pluggable database (PDB) has a service name associated with it, the service name matches the pluggable database name. In order to connect from within Oracle GoldenGate to the specific pluggable database, the network configuration is required. The following is portion of the TNSNAMES.ORA for connecting to the T2C1PDB01 pluggable database as Oracle GoldenGate target database.

```
# $ORACLE_HOME/network/admin/tnsnames.ora
#
T1C1PDB01 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = ggs-target) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = T1C1PDB01)
    )
  )
```

Test the connection to the pluggable database before creating the local type database user. The built-in function SYS\_CONTEXT returns the session current container database name.

```
$ . oraenv
ORACLE_SID = [T2C1] ?
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Mon Sep 16 03:36:59 2013

Copyright (c) 1982, 2013, Oracle. All rights reserved.

SQL> connect system/oracle_4U@t2c1pdb01
Connected.
SQL> SELECT sys_context('USERENV', 'CON_NAME')
       2 FROM dual;

SYS_CONTEXT('USERENV','CON_NAME')
-----
T2C1PDB01

SQL>
```



## Source database user

In a multitenant database, the transaction logs (online redo log files) are part of root level database. Oracle GoldenGate capture process must uses a common user for change data capture from the transaction logs.

```
SQL> show con_name

CON_NAME
-----
CDB$ROOT
SQL> create user c##ggs_admin identified by oracle;

User created.

SQL> grant dba to c##ggs_admin;

Grant succeeded.

SQL>
```

## Target database user

GoldenGate change data delivery process (replicate) requires a local database user for pluggable database (PDB). The user login to the pluggable database to perform database SQL calls to apply changes.

An error is returned when using common user at the root level database.

```
SQL> show con_name

CON_NAME
-----
T2C1PDB01
SQL> create user ggs_admin identified by oracle;

User created.

SQL> grant dba to ggs_admin;

Grant succeeded.
```

The above GRANT command grants DBA privilege to the local user ggs\_admin, which is a highly powerful database access rights and should not the practice on production environment. Refer to chapter 2 to learn more about source and target database privileges required by Oracle GoldenGate extracts and replicate processes.

## Checkpoint table

The database schema, `ggs_admin`, for the checkpoint table specified when adding the replicate process via `ADD REPLICAT` command or globally on `GLOBAL` parameter file uses the pluggable database local schema, explicitly connecting to the pluggable database using the network alias `T2C1PDB01` from `GGSCI` command line.

```
GGSCI (ggs-target) 1> dblogin userid ggs_admin@t2c1pdb01, password
oracle
Successfully logged into database T2C1PDB01.

GGSCI (ggs-target) 2> add checkpointtable

No checkpoint table specified. Using GLOBALS specification
(ggs_admin.chkpttab)...
Logon catalog name T2C1PDB01 will be used for table specification
T2C1PDB01.ggs_admin.chkpttab.

Successfully created checkpoint table T2C1PDB01.ggs_admin.chkpttab.

GGSCI (ggs-target) 3>
```

## Change data delivery

The pluggable database is determined by the `USERID` parameter of the replicate process parameter file. The `USERID` parameter resolves the pluggable database service name. Now, there are two replicate processes `RS1E201` and `RPDB01`, both use the same remote trail `./dirdat/ta`. `RS1E201` applies transactions to non-CDB database `T1C1`, while `RPDB01` applies transactions to the pluggable database `T2C1PDB01`.

```
-- Parameter file: rpdb01.prm
--

replicat rpdb01
setenv (ORACLE_SID=T2C1)
setenv (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
userid ggs_admin@t2c1pdb01, password oracle
assumtargetdefs
discardfile ./dirrpt/rpdb01.dsc, purge
map osm$repapi.customers, target osm$repapi.customers;
map osm$repapi.policies, target osm$repapi.policies;

GGSCI (ggs-target) 3> ADD REPLICAT rpdb01, EXTTRAIL ./dirdat/ta
REPLICAT added.

GGSCI (ggs-target) 5> START REPLICAT rpdb01
```

Sending START request to MANAGER ...  
REPLICAT RPDB01 starting

GGSCI (ggs-target) 6> INFO ALL

| Program  | Status  | Group   | Lag at Chkpt | Time Since Chkpt |
|----------|---------|---------|--------------|------------------|
| MANAGER  | RUNNING |         |              |                  |
| REPLICAT | RUNNING | RPDB01  | 00:00:00     | 00:00:27         |
| REPLICAT | STOPPED | RS1E201 | 00:00:00     | 00:13:50         |

GGSCI (ggs-target) 7>

## Summary

This chapter has covered Oracle GoldenGate fundamentals. The topics presented provided the basis for applying Oracle GoldenGate to an application area requiring design specific configuration to meet the business requirements. Areas where Oracle GoldenGate commonly deployed are:

- Query offloading to reporting instance
- Business continuity for high-availability solutions
- Zero downtime migration and upgrade
- Data warehouse and operational business intelligence
- Data integration services

Oracle GoldenGate topologies and architecture flexibility extends the deployed application area to suite business requirements. For example, query offloading architecture is deployable as one-to-one or many-to-one configuration, depending on the number of data sources and data consolidation requirements. Chapter 4 discusses the technique from a generic prospective and from applying it to Oracle R12 E-Business suite, an important area for the majority of Oracle customers.

Heterogeneity is a key differentiator for Oracle GoldenGate for supporting an enterprise environment running various database management system. The chapter has covered Oracle GoldenGate 12c for Oracle database MS SQL Server, these platforms commonly found at enterprise data processing environment, moving customers data from the workgroup computing servers running MS Windows to EXADATA machine for data consolidation and business intelligence.

The chapter also discussed very specialized application using Oracle GoldenGate 12c adapters, which are necessary components for extending the capabilities to any form of integration services, and this is a major needs for enterprise customers moving data from the mainframe to Linux/UNIX platforms. With Oracle GoldenGate 12c, it's all about being heterogeneous and real-time data movement.

The next chapter will get you started in preparation for deploying Oracle GoldenGate 12c to meet your business requirements. It covers installation and preparation of Oracle database 11g, Oracle Database 12c and Microsoft MS SQL Server for Oracle GoldenGate 12c. Those who need hands-on need to download the virtual machines from <http://www.oracle-ggs.com> to support the workshops' examples.

Good luck, and let's get started with Oracle GoldenGate 12c, *the Oracle bridge*.

# Enterprise Data Warehouse from an Oracle Standby Database

## B

## Overview

Sometimes, due mainly to security reasons, installation of additional software in the production system is not an option, when this is the case, the alternative is to empower Oracle auto-managed standby database as the source of Oracle GoldenGate 12c change data capture (CDC).

There are two modes to deploy and operate Oracle GoldenGate 12c on an auto-managed standby database:

- Oracle GoldenGate 12c running on *Active Data Guard* mode
- Oracle GoldenGate 12c running on *Archive Only Log (AOL)* mode

## Which Mode to Choose?

When the need for real-time change data capture is essential, then Oracle GoldenGate 12c running *Data Guard* is the appropriate mode. In this case the redo-transport service sends the transactions logs on SYNC mode, from which Oracle GoldenGate 12c synchronously performs change data capture and write to the extract trail. However; when the non-real-time meets the business and technical requirements, then Oracle GoldenGate 12c running on *Archive Only Log (AOL)* is the appropriate mode. The two modes are mutually exclusive, only one mode configurable at a time.

The *Archive Only Log (AOL)* mode has two options for change data capture (CDC):

- Change data capture from the primary archive log files
- Change data capture from the standby archive log files

However; both modes (Data Guard and AOL) demand enablement of the primary database supplemental logging, and table-level supplemental logging which optionally run remotely using GGSCI – GoldenGate Software Command Interface.

# Architecture

The approach of empowering Oracle managed-standby database has revolutionized the change data capture technology. Figure 1 depicts the components of Oracle GoldenGate 12c running data guard mode.

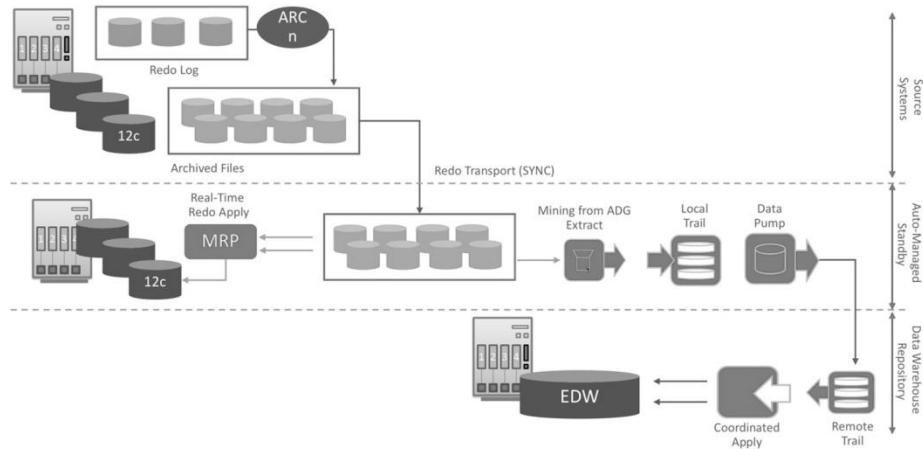


*Figure 1: Oracle GoldenGate 12c on Data Guard Mode*

The core system is configured with redo-transport service to transport the transaction logs to the remote standby database. Since the standby database is configured for real-time apply, the Managed Recovery Process (MRP) applies the transaction on real-time, hence the primary database (source) and the standby database are on synchronized.

The Extract is configured to mine the online standby redo logs continuously, creating the extract trails, which are consumed by the data pump process. The data pump process through TCP/IP sends the data to the remote system running Oracle GoldenGate 12c instance. The remote trails created, consumed by the Replicat process and applied to the target database – EDW.

Similarly, for Archive Only Log (AOL) we need to specify the location and format of the archive log files up on arrival of the archive log file. The only drawback of the AOL is the change data capture is not real-time. This is because the shipment of the archive log file happens internally, that when the log switch occurs, the online redo log file sent to the archive destination, and forwarded to the remote standby database to be applied. For consistent archive log transport, configure log switch through the database parameter `LOG_CHECKPOINT_INTERVAL`, this ensures the change data capture occurs consistently with pre-determined interval.

# Enable Change Data Capture Method

The two change data capture methods from standby database configurable using its own set of parameters, hence they are mutually exclusive.

## Archive Log Only Mode

This method is for non-real-time change data capture option. The extract waits for the arrival of archive log file, which trigger change data capture. The Archive Log Only (AOL) mode is configuring by setting the following Extract parameters.

```
tranlogoptions ARCHIVEDLOGONLY  
tranlogoptions ALTARCHIVELOGDEST <destination>  
tranlogoptions ALTARCHIVEDLOGFORMAT <format>
```

Extract configuration example:

```
tranlogoptions ARCHIVEDLOGONLY  
tranlogoptions ALTARCHIVELOGDEST /u01/app/oracle/arch/DXDB/archivelog  
tranlogoptions ALTARCHIVEDLOGFORMAT DXDB_%t_%s_%r.arc
```

Ensure the ALTARCHIVELOGDEST is not located on the fast recovery area (FRA), as the location of the archive log files is a dynamic FRA subdirectories with the name derived as the current system date, which is not recognized by the Oracle GoldenGate 12c primary extract.

## Active Data Guard Mode

This method is for real-time change data capture. The data guard mode enabled by setting one-and-only Extract parameter as shown below.

```
tranlogoptions minefromactivedg
```

The parameter `minefromactivedg` enables real-time change data capture from the online standby redo log files. The standby database is running on active data guard mode with real-time apply, which continuously synchronizes the primary database and the standby database.

With active data guard mode, the locations of the archive log files are irrelevant, because as stated earlier the change data capture processed from the standby online redo log files.

# Data Guard Configuration

Figure 2 shows the configuration of the primary database (DHDB) and the physical standby database (DXDB). The configuration is part of the data guard broker and configured for automatic failover and reinstatement. The Oracle Database 12c active data guard is enabled for real-time applies.

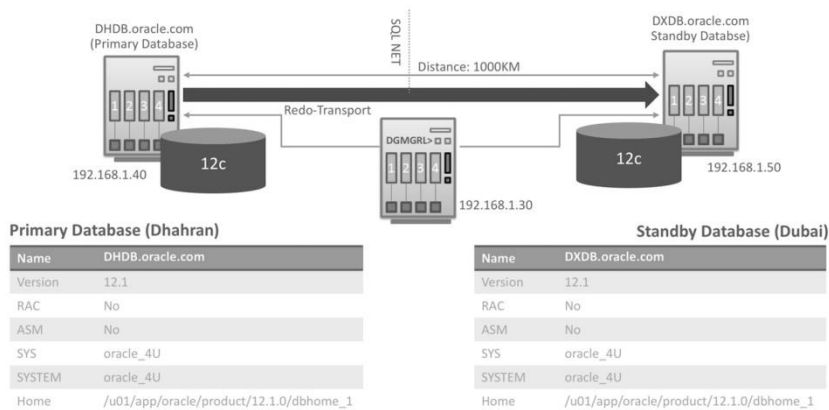


Figure 2: Primary Database and Standby Database Configuration

```
DGMGRL> connect sys/oracle
Connected as SYSDB.
DGMGRL> show configuration

Configuration - DG_CONF

  Protection Mode: MaxPerformance
  Databases:
    DHDB - Primary database
    DXDB - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

DGMGRL>
```

As shown above, the DGMGRL> SHOW CONFIGURATION command shows a summary of the data guard configuration. Prior to deploying Oracle GoldenGate 12c on active data guard, the data guard configuration status must be SUCCESS as illustrated. As shown in the data guard broker configuration, the data guard protection mode is operating on MaxPerformance (default). The available three data protection modes are: *Maximum Protection*, *Maximum Availability* and *Maximum Performance*.



## Switchover from Primary (DHDB) to Standby (DXDB)

The switchover from the primary to standby database is performed to ensure the active data guard configuration is fully operational.

```
DGMGRL> switchover to 'DXDB';

Performing switchover NOW, please wait...
Operation requires a connection to instance "DXDB" on database "DXDB"
Connecting to instance "DXDB"...
Connected as SYSDBA.
New primary database "DXDB" is opening...
Operation requires startup of instance "DHDB" on database "DHDB"
Starting instance "DHDB"...
ORACLE instance started.
Database mounted.
Database opened.

Switchover succeeded, new primary is "DXDB"

DGMGRL>
```

## Switchover from Primary (DXDB) to Standby (DHDB)

Restore the configuration to the original primary database – DHDB.

```
DGMGRL> switchover to 'DHDB';

Performing switchover NOW, please wait...
Operation requires a connection to instance "DHDB" on database "DHDB"
Connecting to instance "DHDB"...
Connected as SYSDBA.
New primary database "DHDB" is opening...
Operation requires startup of instance "DXDB" on database "DXDB"
Starting instance "DXDB"...
ORACLE instance started.
Database mounted.
Database opened.

Switchover succeeded, new primary is "DHDB"

DGMGRL>
```

## Database Role

The database role on the primary system must show PRIMARY. Login as SYSDBA and verify the database role on the primary database.

```
SQL> select database_role from v$database;

DATABASE_ROLE
```

```
-----  
PRIMARY  
SQL>
```

Login as SYSDBA on the standby database; verify the database role as PHYSICAL STANDBY.

```
SQL> select database_role from v$database;  
  
DATABASE_ROLE  
-----  
PHYSICAL STANDBY  
  
SQL>
```

## GoldenGate Administrator

The GoldenGate administrator is a dedicated database user for performing database operations specific to Oracle GoldenGate 12c. Since this is part of Oracle Data Guard configuration, the administrator database users will be created on the primary database and EDW database only. By creating the user on the primary database it's automatically created on the standby.

Follow the steps below to create the administrative user on the Primary database:

```
SQL> conn sys/oracle@DHDB AS SYSDBA  
Connected.  
SQL> create tablespace OGG_DAT;  
  
Tablespace created.  
  
SQL> grant dba to ogg_admin identified by oracle;  
  
Grant succeeded.  
  
SQL> alter user ogg_admin default tablespace OGG_DAT  
2 quota unlimited on OGG_DAT;  
  
User altered.  
  
SQL>
```

Follow the steps below to create the administrative user on the target database (EDW):

```
SQL> conn sys/oracle_4U@EDW AS SYSDBA  
Connected.  
SQL> create tablespace OGG_DAT;
```

```

Tablespace created.

SQL> grant dba to ogg_admin identified by oracle;

Grant succeeded.

SQL> alter user ogg_admin default tablespace OGG_DAT
      2 quota unlimited on OGG_DAT;

User altered.

SQL>

```

## Installations

Oracle GoldenGate 12c installations performed on the following locations:

- The primary database system (optional)
- The standby database system
- The target database system

The installation on the primary database is avoided; instead a remote GGSCI connection from standby or target system is established to generate table-level supplemental logging.

Generally, there are two methods of installation: Oracle Universal Installer and silent mode. The section empowers silent mode installation method. The silent method uses a response file that need to be created and specified when running the installer.

## Installation on the Standby Database System

The installation performed using silent mode, which requires a response file.

```
[oracle@stdby01 ~]$ cat oggcore1.rsp
```

```

oracle.install.responseFileVersion=rsfmt ogginstall response schema v12 3 0
INSTALL_OPTION=ORA12c
SOFTWARE_LOCATION=/u01/app/oracle/product/12.3.0/gghome 1
START_MANAGER=true
MANAGER_PORT=7840
DATABASE_LOCATION=/u01/app/oracle/product/12.1.0/dbhome_1
INVENTORY_LOCATION=/u01/app/oraInventory
UNIX_GROUP_NAME=oinstall

```

Begin the installation using the runInstaller program shown below. The installer program is invoked from a customized shell script – runInsaller.sh.

```
$runInstaller -silent -responsefile /home/oracle/oggcore1.rsp -showProgress  
-force -waitforcompletion
```

```
[oracle@stdby01 ~]$ ./runInstall.sh  
  
DISPLAY turned off Successfully  
  
The Silent install option...  
  
oracle.install.responseFileVersion=rsfmt_ogginstall_response_schema_v12_3_0  
INSTALL_OPTION=ORA12c  
SOFTWARE_LOCATION=/u01/app/oracle/product/12.3.0/gghome_1  
START_MANAGER=true  
MANAGER_PORT=7840  
DATABASE_LOCATION=/u01/app/oracle/product/12.1.0/dbhome_1  
INVENTORY_LOCATION=/u01/app/oraInventory  
UNIX_GROUP_NAME=oinstall  
  
/stage/ogg1230/fbo ggs Linux x64 shiphome/Disk1/runInstaller -silent -  
responsefile /home/oracle/oggcore1.rsp -nowelcome -showProgress -force -  
waitforcompletion  
  
Press Enter to Begin Oracle GoldenGate Silent Install.....  
Do not interrupt the installation while in-progress.....  
It will take approximately 5-10 minutes. Press Enter to begin....  
Starting Oracle Universal Installer...  
  
Checking Temp space: must be greater than 120 MB.   Actual 27793 MB  
Passed  
Checking swap space: must be greater than 150 MB.   Actual 9964 MB   Passed  
Preparing to launch Oracle Universal Installer from /tmp/OraInstall2018-03-  
21 03-16-32AM. Please wait ...You can find the log of this install session  
at:  
  /u01/app/oraInventory/logs/installActions2018-03-21_03-16-32AM.log  
  
Prepare in progress.  
..... 9% Done.  
  
Prepare successful.  
  
Copy files in progress.  
..... 31% Done.  
..... 44% Done.  
..... 65% Done.  
..... 70% Done.  
..... 89% Done.  
.....  
Copy files successful.  
  
Link binaries in progress.  
  
Link binaries successful.  
  
Setup files in progress.  
  
Setup files successful.  
  
Setup Inventory in progress.
```

```

Setup Inventory successful.

Finish Setup successful.
The installation of Oracle GoldenGate Core was successful.
Please check '/u01/app/oraInventory/logs/silentInstall2018-03-21_03-16-32AM.log' for more details.

Running Configuration Assistants in progress.

Configuring and starting manager... in progress.
..... 100% Done.

Configuring and starting manager... successful.
Successfully Setup Software.
[oracle@stdby01 ~]$

```

## Start GoldenGate Software Command Interface (GGSCI)

Start the GGSCI and test login to the primary database and standby database.

```

[oracle@stdby01 ~]$ cd $OGG_HOME
[oracle@stdby01 gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE_12.3.0.1.0_PLATFORMS_170721.0154_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (stdby01.oracle.com) 1> dblogin userid ogg_admin@DHDB, password oracle
Successfully logged into database.

GGSCI (stdby01.oracle.com as ogg_admin@DHDB) 2> dblogin userid
ogg_admin@DXDB, password oracle
Successfully logged into database.

GGSCI (stdby01.oracle.com as ogg_admin@DXDB) 3>

```

Verify the Oracle GoldenGate 12c instance

```

[oracle@stdby01 gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE 12.3.0.1.0 PLATFORMS 170721.0154_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

```

```

GGSCI (stdby01.oracle.com) 1> info mgr

Manager is running (IP port stdby01.oracle.com.7840, Process ID 9075).

GGSCI (stdby01.oracle.com) 2> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING

GGSCI (stdby01.oracle.com) 3> show

Parameter settings:

SET DEBUG      OFF

Current directory: /u01/app/oracle/product/12.3.0/gghome 1

Editor: vi

Reports (.rpt)           /u01/app/oracle/product/12.3.0/gghome_1/dirprt
Parameters (.prm)        /u01/app/oracle/product/12.3.0/gghome_1/dirprm
Replicat Checkpoints (.cpr) /u01/app/oracle/product/12.3.0/gghome_1/dirchk
Extract Checkpoints (.cpe) /u01/app/oracle/product/12.3.0/gghome_1/dirchk
Process Status (.pcs)    /u01/app/oracle/product/12.3.0/gghome_1/dirpcs
SQL Scripts (.sql)       /u01/app/oracle/product/12.3.0/gghome_1/dirsq
Database Definitions (.def) /u01/app/oracle/product/12.3.0/gghome_1/dirdef
Dump files (.dmp)        /u01/app/oracle/product/12.3.0/gghome_1/dir
Masterkey wallet files (.wlt) /u01/app/oracle/product/12.3.0/gghome_1/dirwlt
Credential store files (.crd) /u01/app/oracle/product/12.3.0/gghome_1/dir

GGSCI (stdby01.oracle.com) 4>

```

## Installation on the Target (EDW) Database System

Install Oracle GoldenGate 12c on the target (EDW) system. The installation is done using the silent mode.

```
[oracle@dgobserver ~]$ cat oggcore1.rsp
```

```

oracle.install.responseFileVersion=rsfmt_ogginstall_response_schema_v12_3_0
INSTALL_OPTION=ORA12c
SOFTWARE_LOCATION=/u01/app/oracle/product/12.3.0/gghome_1
START_MANAGER=true
MANAGER_PORT=7830
DATABASE_LOCATION=/u01/app/oracle/product/12.1.0/dbhome_1
INVENTORY_LOCATION=/u01/app/oraInventory
UNIX_GROUP_NAME=oinstall
[oracle@dgobserver ~]$

```

Begin the installation using the runInstaller program shown below. The installer program is invoked from a customized shell script – runInsaller.sh.

```
$runInstaller -silent -responsefile /home/oracle/oggcore1.rsp -showProgress  
-force -waitforcompletion
```

```
[oracle@dgobserver ~]$ ./runInstall.sh  
  
DISPLAY turned off Successfully  
  
The Silent install option...  
  
oracle.install.responseFileVersion=rsfmt_ogginstall_response_schema_v12_3_0  
INSTALL_OPTION=ORA12c  
SOFTWARE_LOCATION=/u01/app/oracle/product/12.3.0/gghome 1  
START_MANAGER=true  
MANAGER_PORT=7840  
DATABASE_LOCATION=/u01/app/oracle/product/12.1.0/dbhome_1  
INVENTORY_LOCATION=/u01/app/oraInventory  
UNIX_GROUP_NAME=oinstall  
  
/stage/ogg1230/fbo ggs Linux x64 shiphome/Disk1/runInstaller -silent -  
responsefile /home/oracle/oggcore1.rsp -nowelcome -showProgress -force -  
waitforcompletion  
  
Press Enter to Begin Oracle GoldenGate Silent Install.....  
Do not interrupt the installation while in-progress.....  
It will take approximately 5-10 minutes. Press Enter to begin.....  
Starting Oracle Universal Installer...  
  
Checking Temp space: must be greater than 120 MB. Actual 27183 MB  
Passed  
Checking swap space: must be greater than 150 MB. Actual 9981 MB Passed  
Preparing to launch Oracle Universal Installer from /tmp/OraInstall2018-03-  
21_03-32-33AM. Please wait ...You can find the log of this install session  
at:  
/u01/app/oraInventory/logs/installActions2018-03-21_03-32-33AM.log  
  
Prepare in progress.  
..... 9% Done.  
  
Prepare successful.  
  
Copy files in progress.  
..... 31% Done.  
..... 44% Done.  
..... 65% Done.  
..... 70% Done.  
..... 89% Done.  
.....  
Copy files successful.  
  
Link binaries in progress.  
  
Link binaries successful.  
  
Setup files in progress.  
  
Setup files successful.
```

```

Setup Inventory in progress.

Setup Inventory successful.

Finish Setup successful.
The installation of Oracle GoldenGate Core was successful.
Please check '/u01/app/oraInventory/logs/silentInstall2018-03-21 03-32-33AM.log' for more details.

Running Configuration Assistants in progress.

Configuring and starting manager... in progress.
..... 100% Done.

Configuring and starting manager... successful.
Successfully Setup Software.
[oracle@dgobserver ~]$

```

## Starting GoldenGate Software Command Interface (GGSCI)

Start the GGSCI and test login to the EDW database, show the instance directory structure.

```

[oracle@dgobserver ~]$ cd $OGG_HOME
[oracle@dgobserver gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE_12.3.0.1.0_PLATFORMS_170721.0154_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (dgobserver.oracle.com) 1> dblogin userid ogg_admin, password oracle
Successfully logged into database.

GGSCI (dgobserver.oracle.com as ogg_admin@EDW) 2> info mgr
Manager is running (IP port dgobserver.oracle.com.7840, Process ID 13131).

GGSCI (dgobserver.oracle.com as ogg_admin@EDW) 3> show

Parameter settings:

SET DEBUG          OFF

Current directory: /u01/app/oracle/product/12.3.0/gghome_1

Editor: vi

Reports (.rpt)           /u01/app/oracle/product/12.3.0/gghome_1/dirrpt
Parameters (.prm)        /u01/app/oracle/product/12.3.0/gghome_1/dirprm
Replicat Checkpoints (.cpr) /u01/app/oracle/product/12.3.0/gghome_1/dirchk

```



```

Extract Checkpoints (.cpe)      /u01/app/oracle/product/12.3.0/gghome_1/dirchk
Process Status (.pcs)          /u01/app/oracle/product/12.3.0/gghome_1/dirpcs
SQL Scripts (.sql)             /u01/app/oracle/product/12.3.0/gghome_1/dirsq
Database Definitions (.def)     /u01/app/oracle/product/12.3.0/gghome_1/dirdef
Dump files (.dmp)              /u01/app/oracle/product/12.3.0/gghome_1/dirdmp
Masterkey wallet files (.wlt)   /u01/app/oracle/product/12.3.0/gghome_1/dirwlt
Credential store files (.crd)   /u01/app/oracle/product/12.3.0/gghome_1/dircrd

```

```
GGSCI (dgobserver.oracle.com as ogg_admin@EDW) 4>
```

## Enable Database Supplemental Logging

A prerequisite of Oracle GoldenGate 12c is to enable the database supplemental logging. The supplemental logging is enabled at the database level using SQL command, and the table-level using GGSCI ADD TRANDATA command. Because Oracle GoldenGate 12c is not installed on the primary database system, GGSCI connects to it remotely as demonstrated below.

Login as SYSDBA on the primary database; enable the minimum database supplemental logging.

```

SQL> conn / as sysdba
Connected.
SQL> alter database add supplemental log data;

Database altered.

SQL> select supplemental_log_data_min
       2 from v$database;

SUPPLEME
-----
YES

SQL> alter system switch logfile;

System altered.

SQL>

```

On the standby system, start GGSCI interface, login to the primary and enable table-level supplemental logging.

```

[oracle@stdby01 gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE_12.3.0.1.0_PLATFORMS_170721.0154_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

```

```
GGSCI (stdby01.oracle.com) 1> dblogin userid ogg_admin@DHDB, password oracle
Successfully logged into database.
```

```
GGSCI (stdby01.oracle.com as ogg_admin@DHDB) 2> ADD TRANDATA osm_repapi.*
```

```
2018-03-21 03:54:04 INFO OGG-15131 Logging of supplemental redo log
data is already enabled for table OSM_REPAPI.CUSTOMERS.
```

```
2018-03-21 03:54:04 INFO OGG-15135 TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.CUSTOMERS.
```

```
2018-03-21 03:54:04 INFO OGG-15131 Logging of supplemental redo log
data is already enabled for table OSM_REPAPI.DISCOUNT_TYPES.
```

```
2018-03-21 03:54:04 INFO OGG-15135 TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.DISCOUNT_TYPES.
```

```
2018-03-21 03:54:06 WARNING OGG-06439 No unique key is defined for table
OSM_DOCUMENTS. All viable columns will be used to represent the key, but may
not guarantee uniqueness. KEYCOLS may be used to define the key.
```

```
2018-03-21 03:54:06 INFO OGG-15130 No key found for table
OSM_REPAPI.OSM_DOCUMENTS. All viable columns will be logged.
```

```
2018-03-21 03:54:06 INFO OGG-15131 Logging of supplemental redo log
data is already enabled for table OSM_REPAPI.OSM_DOCUMENTS.
```

```
2018-03-21 03:54:06 INFO OGG-15135 TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.OSM_DOCUMENTS.
```

```
2018-03-21 03:54:06 INFO OGG-15131 Logging of supplemental redo log
data is already enabled for table OSM_REPAPI.POLICIES.
```

```
2018-03-21 03:54:06 INFO OGG-15135 TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.POLICIES.
```

```
2018-03-21 03:54:06 INFO OGG-15131 Logging of supplemental redo log
data is already enabled for table OSM_REPAPI.POLICY_TYPES.
```

```
2018-03-21 03:54:06 INFO OGG-15135 TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.POLICY_TYPES.
```

```
2018-03-21 03:54:06 INFO OGG-15135 TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.RANDOM_DOMAIN.
```

```
2018-03-21 03:54:06 INFO OGG-15131 Logging of supplemental redo log
data is already enabled for table OSM_REPAPI.RANDOM_FIRST.
```

```
2018-03-21 03:54:06 INFO OGG-15131 Logging of supplemental redo log
data is already enabled for table OSM_REPAPI.RANDOM_LAST.
```

```
2018-03-21 03:54:06 INFO OGG-15135 TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.RANDOM_LAST.
```

```
2018-03-21 03:54:06 INFO OGG-15131 Logging of supplemental redo log
data is already enabled for table OSM_REPAPI.RANDOM_MIDDLE.
```

```

2018-03-21 03:54:06 INFO      OGG-15135  TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.RANDOM_MIDDLE.

GGSCI (stdby01.oracle.com as ogg_admin@DHDB) 3>

```

## Overhead of Enabling Supplemental Logging

An Oracle GoldenGate 12c database prerequisite is enablement of supplemental logging at the database-level and table-level. There are several levels of database supplemental logging, GoldenGate 12c primary extract process requires the minimum logging enabled by the ALTER command while connected AS SYSDBA.

```
SQL> alter database add supplemental log data;
```

The table-level supplemental logging is enabled using GGSCI ADD TRANDATA command while connected to the source database using the GoldenGate administrator.

```
GGSCI (stdby01.oracle.com as ogg_admin@DHDB) 2> ADD TRANDATA osm_repapi.*
```

Figure 3 shows the system resources utilization. The left System Monitor is without any form of supplemental logging, the right System Monitor with database supplemental logging and table-level supplemental logging are enabled.

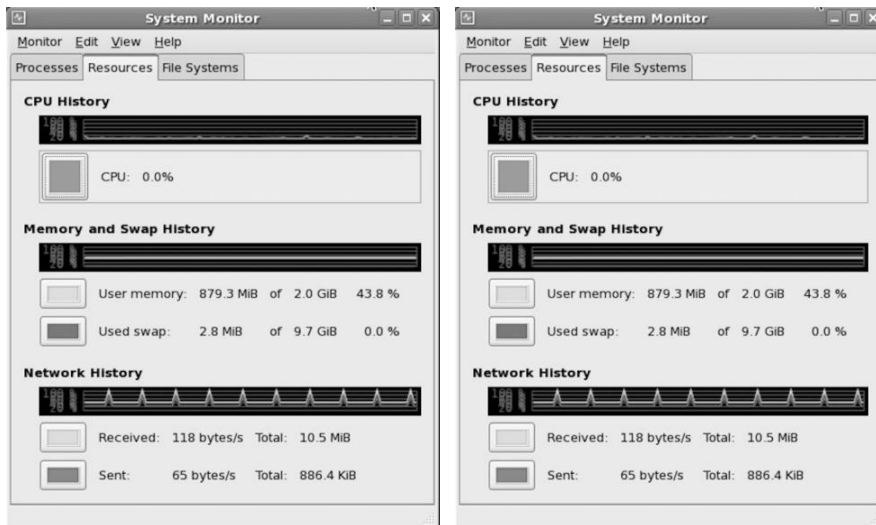


*Figure 3: System Resource Utilization with (left)/without (right) Supplemental Logging*  
Figure 3 diagrams clearly show the overhead associated with supplemental logging is relatively low, no spark on CPU and Memory, hence it should not cause performance

related issues. Figure 4 analyzes the behavior related to supplemental logging, in regard to log switch and archive log file frequencies.

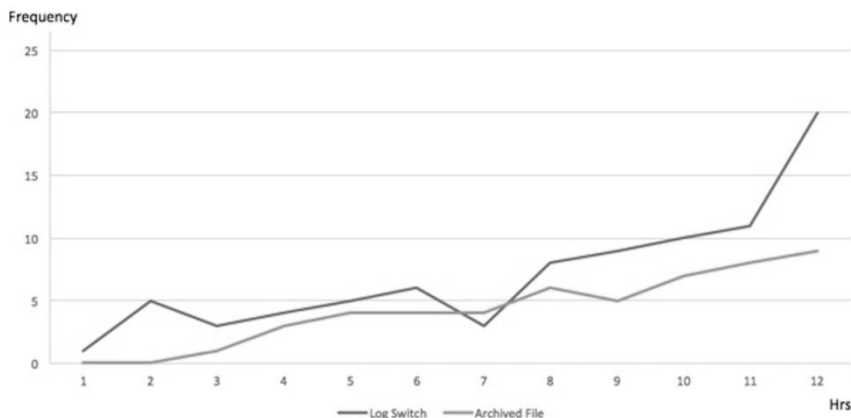


Figure 4: Log Switch and Archive Log File Patterns

## Configure and Define the Primary Extract

Because the two operating modes (data guard and AOL) are mutually exclusive, this section configures and defines the change data capture for data guard mode. The data guard mode delivers a real-time change data capture from an active data guard instance, receiving the redo-transport from the primary database.

### Active Data Guard Mode

The following configure and define the primary extract for active data guard mode, enabled by the parameter:

```
tranlogoptions minefromactivedg
```

Prepare the parameter file for the primary extract – EDXDB01.

```
GGSCI (stdby01.oracle.com as ogg_admin@DHDB) 4> edit params EDXDB01
```

```
EXTRACT EDXDB01
SETENV (ORACLE_SID = DXDB)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
SETENV (TNS_ADMIN = "/u01/app/oracle/product/12.1.0/dbhome_1/network/admin")
USERID ogg_admin@DXDB, PASSWORD oracle
tranlogoptions minefromactivedg
EXTTRAIL ./dirdat/sa
```

```
TABLE osm_repapi.*;
```

Define the primary extract – EDXDB01, and then link it to the local extract trail ./dirdir/sa. Notice; though the database is a single instance, the parameter THREADS specified as 2, this is because there are two members of the redo log groups.

```
GGSCI (stdby01.oracle.com as ogg_admin@DHDB) 5> add extract EDXDB01,
tranlog, begin now, threads 2
EXTRACT added.

GGSCI (stdby01.oracle.com as ogg_admin@DHDB) 6> add exttrail ./dirdat/sa,
extract EDXDB01
EXTTRAIL added.

GGSCI (stdby01.oracle.com as ogg_admin@DHDB) 7>
```

## Archive Only Log mode

The following configure the primary extract for active only log (AOL) mode. The parameters related to AOL mode are:

```
tranlogoptions ARCHIVEDLOGONLY
tranlogoptions ALTARCHIVELOGDEST /u01/app/oracle/arch/DXDB/archivelog
tranlogoptions ALTARCHIVEDLOGFORMAT DXDB_%t_%s_%r.arc
```

```
EXTRACT EDXDB01
SETENV (ORACLE_SID = DXDB)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN AMERICA.AL32UTF8)
SETENV (TNS_ADMIN = "/u01/app/oracle/product/12.1.0/dbhome_1/network/admin")
USERID ogg_admin@DXDB, PASSWORD oracle
tranlogoptions ARCHIVEDLOGONLY
tranlogoptions ALTARCHIVELOGDEST /u01/app/oracle/arch/DXDB/archivelog
tranlogoptions ALTARCHIVEDLOGFORMAT DXDB_%t_%s_%r.arc
EXTTRAIL ./dirdat/sa
TABLE osm_repapi.*;
```

## Define and Configure the Secondary Extract

Prepare the parameter file for the secondary extract (data pump) – PDXDB01. The data pump sends the trail transaction to the remote Oracle GoldenGate 12c instance as referenced by RMTHOST and MGRPORT parameters.

```
GGSCI (stdby01.oracle.com as ogg_admin@DHDB) 7> edit params PDXDB01
```

```
EXTRACT PDXDB01
```

```

SETENV (ORACLE_SID = DXDB)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@DXDB, PASSWORD oracle
RMTHOST 192.168.1.30, MGRPORT 7830
RMTRAIL ./dirdat/ta
TABLE osm_repapi.*;

```

Defined the secondary extract (PDXDB01) and link it to the remote trail ./dirdat/ta.

```

GGSCI (stdby01.oracle.com as ogg_admin@DHDB) 8> add extract PDXDB01,
extrailsources ./dirdat/sa
EXTRACT added.

GGSCI (stdby01.oracle.com as ogg_admin@DHDB) 9> add rmttrail ./dirdat/ta,
extract PDXDB01
RMTRAIL added.

GGSCI (stdby01.oracle.com as ogg_admin@DHDB) 10>

```

## Configure and Define Apply

Because the default checkpoint for the Replicat is a database table, the checkpoint table must be created first.

Configure and define the checkpoint table, the name of the parameter file is GLOBAL with the default location as the Oracle GoldenGate 12c root directory.

```

[oracle@dgobserver gghome_1]$ pwd
/u01/app/oracle/product/12.3.0/gghome_1
[oracle@dgobserver gghome_1]$ vi GLOBALS

```

```

checkpointtable ogg_admin.chkpttab

```

Start GGSCI, login to the target database (EDW), and execute the command ADD CHECKPOINTTABLE.

```

[oracle@dgobserver gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE_12.3.0.1.0_PLATFORMS_170721.0154_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (dgobserver.oracle.com) 1> dblogin userid ogg_admin, password oracle

```

```

Successfully logged into database.

GGSCI (dgobserver.oracle.com as ogg_admin@EDW) 2> add checkpointtable

No checkpoint table specified. Using GLOBALS specification
ogg_admin.chkpttab...

Successfully created checkpoint table ogg_admin.chkpttab.

GGSCI (dgobserver.oracle.com as ogg_admin@EDW) 3>

```

Prepare the configuration parameters file for the Replicat RDXDB. Since its coordinated apply Replicat, with the length of maximum name 5 alphanumeric characters.

```

GGSCI (dgobserver.oracle.com as ogg_admin@EDW) 3> edit params RDXDB

```

```

REPLICAT RDXDB
SETENV (ORACLE_SID = EDW)
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@EDW, PASSWORD oracle
REPERERROR (DEFAULT, DISCARD)
DISCARDFILE /u01/app/oracle/product/12.3.0/gghome 1/dirrpt/RDXDB01.dsc,
PURGE
map osm_repapi.*, target osm_repapi.*;

```

Define the parallel Replicat as parallel, reading the remote trails stored at ./dirdat/ta.

```

GGSCI (dgobserver.oracle.com as ogg_admin@EDW) 4> add replicat RDXDB,
parallel, exttrail ./dirdat/ta
REPLICAT (Parallel) added.

GGSCI (dgobserver.oracle.com as ogg_admin@EDW) 5>

```

## Start the Oracle GoldenGate Extracts on Source

Start the primary and secondary extracts process (EDXDB01 and PDXDB01).

```

GGSCI (stdby01.oracle.com as ogg_admin@DHDB) 21> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
EXTRACT      STOPPED     EDXDB01    00:00:00     00:00:02
EXTRACT      STOPPED     PDXDB01    00:00:00     00:00:01

GGSCI (stdby01.oracle.com as ogg_admin@DHDB) 22> start *

Sending START request to MANAGER ...

```

```

EXTRACT EDXDB01 starting

Sending START request to MANAGER ...
EXTRACT PDXDB01 starting

GGSCI (stdby01.oracle.com as ogg_admin@DHDB) 23> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
EXTRACT      RUNNING     EDXDB01    00:00:00      00:00:07
EXTRACT      RUNNING     PDXDB01    00:00:00      00:00:06

GGSCI (stdby01.oracle.com as ogg_admin@DHDB) 24>

```

## Start the Oracle GoldenGate Replicat on Target (EDW)

The integrated Replicat is started; it applies pending transaction stored on the remote trails.

```

GGSCI (dgobserver.oracle.com as ogg_admin@EDW) 11> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
REPLICAT     STOPPED     RDXDB      00:00:00      00:07:06

GGSCI (dgobserver.oracle.com as ogg_admin@EDW) 12> start *

Sending START request to MANAGER ...
REPLICAT RDXDB starting

GGSCI (dgobserver.oracle.com as ogg_admin@EDW) 13> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
REPLICAT     RUNNING     RDXDB      00:00:00      00:07:11

GGSCI (dgobserver.oracle.com as ogg_admin@EDW) 14>

```

## Perform Workload Testing on Data Guard mode

Start the workload simulator on the primary database by invoking CALL\_TRANS\_LOAD database procedure.

```

SQL> conn osm_repapi/oracle

```



```
Connected.
SQL> exec call_trans_load;
```

Change directory to ./dirdat and list the trail files.

```
[oracle@stdby01 dirdat]$ ls -al
total 92
drwxr-x--- 2 oracle oinstall 4096 Mar 21 04:47 .
drwxr-xr-x 29 oracle oinstall 4096 Mar 21 04:45 ..
-rw-r----- 1 oracle oinstall 1431 Mar 21 04:47 sa000000000
-rw-r----- 1 oracle oinstall 1493 Mar 21 04:47 sa000000001
-rw-r----- 1 oracle oinstall 70171 Mar 21 04:51 sa000000002
[oracle@stdby01 dirdat]$ ls -al
total 96
drwxr-x--- 2 oracle oinstall 4096 Mar 21 04:47 .
drwxr-xr-x 29 oracle oinstall 4096 Mar 21 04:45 ..
-rw-r----- 1 oracle oinstall 1431 Mar 21 04:47 sa000000000
-rw-r----- 1 oracle oinstall 1493 Mar 21 04:47 sa000000001
-rw-r----- 1 oracle oinstall 74962 Mar 21 04:51 sa000000002
[oracle@stdby01 dirdat]$
```

Wait for a short while, re-display the list of trail files, and notice the file size is incrementing, meaning the transactions are successfully shipping to the remote instance.

```
[oracle@dglobserver dirdat]$ ls -al
total 124
drwxr-x--- 2 oracle oinstall 4096 Mar 21 04:47 .
drwxr-xr-x 28 oracle oinstall 4096 Mar 21 04:38 ..
-rw-r----- 1 oracle oinstall 113397 Mar 21 04:51 ta000000000
[oracle@dglobserver dirdat]$ ls -al
total 128
drwxr-x--- 2 oracle oinstall 4096 Mar 21 04:47 .
drwxr-xr-x 28 oracle oinstall 4096 Mar 21 04:38 ..
-rw-r----- 1 oracle oinstall 117438 Mar 21 04:52 ta000000000
[oracle@dglobserver dirdat]$
```

This is a real-time apply, the row count is incrementally displayed.

```
SQL> select count(*) from customers;

COUNT (*)
-----
1927781

SQL> /

COUNT (*)
-----
1927792

SQL>
```

## Perform Workload Testing on AOL mode

Edit the primary extract parameter file to enable the AOL mode as shown in Table 1. Ensure the parameter `ALTARCHIVELOGDEST` is not located on the Fast Recovery Area (FRA), as the archived log files are created on cyclic sub-directories named with the system date.

| Parameter                         | Description                                                                              |
|-----------------------------------|------------------------------------------------------------------------------------------|
| <code>ARCHIVEDLOGONLY</code>      | The mode of the change data capture, This parameter enables AOL mode.                    |
| <code>ALTARCHIVELOGDEST</code>    | Specifies the location of the archive log files as found on the database parameter file. |
| <code>ALTARCHIVEDLOGFORMAT</code> | Specifies the format of the archive log file as found on the database parameter file.    |

*Table 1: AOL Mode Parameters Settings*

Edit the primary extract parameter file to disable data guard mode, and enable AOL mode.

```
GGSCI (stdby01.oracle.com) 4> edit params EDXDB01
```

```
EXTRACT EDXDB01
SETENV (ORACLE_SID = DXDB)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN AMERICA.AL32UTF8)
SETENV (TNS_ADMIN = "/u01/app/oracle/product/12.1.0/dbhome_1/network/admin")
USERID ogg admin@DXDB, PASSWORD oracle
-- tranlogoptions minefromactivedg
tranlogoptions ARCHIVEDLOGONLY
tranlogoptions ALTARCHIVELOGDEST /u01/app/oracle/arch/DXDB/archivelog
tranlogoptions ALTARCHIVEDLOGFORMAT DXDB_%t_%s_%r.arc
EXTTRAIL ./dirdat/sa
TABLE osm_repapi.*;
```

Stop the primary extract. Also, alter the extract change data capture timestamp to now, this advanced the extract RBA and re-start the primary extract. Finally verify the extract status.

```
GGSCI (stdby01.oracle.com) 5> stop *

Sending STOP request to EXTRACT EDXDB01 ...
Request processed.

Sending STOP request to EXTRACT PDXDB01 ...
Request processed.

GGSCI (stdby01.oracle.com) 6> alter extract EDXDB01 begin now
EXTRACT altered.
```

```

GGSCI (stdby01.oracle.com) 7> start *

Sending START request to MANAGER ...
EXTRACT EDXDB01 starting

Sending START request to MANAGER ...
EXTRACT PDXDB01 starting

GGSCI (stdby01.oracle.com) 8> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
-----
MANAGER      RUNNING
EXTRACT      RUNNING      EDXDB01      00:00:00      00:00:07
EXTRACT      RUNNING      PDXDB01      00:00:00      00:00:06

GGSCI (stdby01.oracle.com) 8>

```

Invoke the workload simulator `CALL_TRANS_LOAD` with default parameters.

```

SQL> conn osm_repapi/oracle
Connected.
SQL> exec call_trans_load;

```

From another session display the current row count.

```

SQL> conn osm_repapi/oracle
Connected.
SQL> select count(*) from customers;

COUNT (*)
-----
1928051

SQL>

```

Manually force a log switch to ship the current online redo log file to the standby database system archive log file location.

```

SQL> alter system switch logfile;

System altered.

SQL>

```

Re-display the current row count, as it shows it advances after applying the transactions of the arrived archived log file.

```

SQL> conn osm_repapi/oracle
Connected.
SQL> select count(*) from customers;

  COUNT (*)
-----
  1928051

SQL> /

  COUNT (*)
-----
  1928281

SQL>

```

## How to Control Change Data Capture for AOL mode

Use the following methods control the interval for change data capture running on AOL mode. The purpose is to achieve consistent interval of change data capture as business rule, workflow data feed or data warehouse refresh.

- Proper Sizing of the Online Redo Log Groups
- Setting Database Parameters
- Use of DBMS\_SCHEDULER

### Proper Sizing of the Online Redo Log Groups

The log switch occurs when the checkpoint reaches the end of the current redo log group. When the transactions volume is consistent, the log switch interval occurs at nearly constant interval. However; when the transactions volume is unpredictable, the log switch interval varies, hence it's not an accurate interval.

### Setting Database Parameters

There are two parameters directly signal the log switch event as described next.

#### **Log\_checkpoint\_interval**

Specifies the frequency of checkpoints in terms of the number of redo log file blocks that can exist between an incremental checkpoint and the last block written to the redo log. This number refers to physical operating system blocks, not database blocks.

Regardless of this value, a checkpoint always occurs when switching from one online redo log file to another. Therefore, if the value exceeds the actual redo log file size,

checkpoints occur only when switching logs. Checkpoint frequency is one of the factors that influence the time required for the database to recover from an unexpected failure.

Specifying a value of 0 for the interval disables interval-based checkpoints

## Log\_checkpoint\_timeout

Specifies (in seconds) the amount of time that has passed since the incremental checkpoint at the position where the last write to the redo log (sometimes called the tail of the log) occurred. This parameter also signifies that no buffer will remain dirty (in the cache) for more than integer seconds.

Specifying a value of 0 for the timeout disables time-based checkpoints

## Use of DBMS\_SCHEDULER

In the case the log switch event is unpredictable, then scheduling the event using

**Step 1:** Create the shell script to force log switch event.

```
#!/bin/bash
export ORACLE_SID=DHDB
export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
$ORACLE_HOME/bin/sqlplus<<eof
connect / as sysdba
alter system switch logfile;
eof
```

**Step 2:** Create and enable the scheduled job to force log switch every 2 minutes (for test only). In real-world it should be based on reasonable business/technical requirement, a reasonable interval value should be 20-30 minutes.

```
SQL> begin
2   dbms_scheduler.create_job(
3     job_name=>'ForceLogSwitch',
4     job_type=>'EXECUTABLE',
5     job_action=>'/home/oracle/ForceAOLCDC.sh',
6     repeat_interval=>'FREQ=MINUTELY; interval=2; bysecond=0;',
7     enabled=>TRUE
8   );
9 end;
10 /

PL/SQL procedure successfully completed.

SQL>
```

**Step 3:** Display current row count from CUSTOMERS table. The row count is based on the last apply.

```
SQL> select count(*) from customers;

COUNT (*)
-----
1929891

SQL>
```

**Step 4:** Wait for two minutes for the log switch event to trigger, re-display the current row count, the row count changes because the Replicat has applied the transaction from the last archive log file(s).

```
SQL> /

COUNT (*)
-----
1932721

SQL>
```

**Step 5:** Disable the job by calling the next DBMS\_SCHEDULER.DISABLE session.

```
SQL> exec dbms_scheduler.disable('ForceLogSwitch');

PL/SQL procedure successfully completed.

SQL>
```

# Summary

A disaster recovery site of the production is an obligation, and sometimes is mandated. Oracle GoldenGate 12c is shipped with active data guard; hence it's an attractive solution for the data guard as the source of change data capture. Oracle Database 12c has two options to deploy a disaster recovery solution:

- *Oracle Data Guard*
- *Oracle Active Data Guard*

The first option, Oracle Data Guard, is used for change data capture using Archive Only Log (AOL) mode, where the change data capture (CDC) is triggered up on the arrival of the archive log file the standby database, allowing non-real-time CDC that based on the elapsed interval of the primary database log switch event. The second option, Oracle Active Data Guard mode, where the change data capture occurs in real-time from the transaction logs sent synchronously to the online standby redo log files. The option is distinguished by the way the database started.

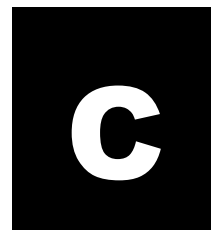
Oracle GoldenGate 12c installed on the standby database system and the target (EDW) system. The installation of Oracle GoldenGate 12c on the primary system is optional. However; because of a one-time GGSCI command – ADD TRANDATA – must be execute on the primary database, the GGSCI connection is remotely established through SQLNET TNS configuration.

Definition and configuration of the primary and the secondary extracts are executed on the standby database system, and the apply process is executed on the target EDW system. The active data guard and archive only log (AOL) modes are mutually exclusive; hence only one mode is configured to support the desired change data capture (CDC) method. Comparing the two modes is outline below:

| Active Data Guard Mode                | Archive Only Log (AOL) Mode                |
|---------------------------------------|--------------------------------------------|
| ADG is bundled with Oracle GoldenGate | Oracle-to-Oracle                           |
| Oracle-to-Oracle                      | Non-Real-Time CDC                          |
| Real-Time change data capture         | Primary or standby archive log file        |
| Integrated Capture and Apply          | RAC restriction of matching size           |
| The primary must have access to GGSCI | Require dedicated SQLNET connection        |
| Set THREADS attributes to 2           | Set THREADS attributes to 2                |
| FRA is configurable                   | Do not use FRA for archive log destination |

The data guard and AOL change data capture mode doesn't replace the downstream technology. Oracle GoldenGate 12c deployed on downstream is strategic architecture, its part of Oracle GoldenGate 12c best practice for Oracle-to-Oracle data replication.

# Migrating the Enterprise Data Warehouse to Oracle Cloud Infrastructure as a Service (IaaS)



## Overview

The journey to Oracle cloud is fascinating, shifting the workload to Oracle cloud IaaS is an objective that sets on the table of many decision makers. The ultimate reasons are cost reduction and elastic platform that expand and shrink based on business needs. The Enterprise Data Warehouse (EDW) is a perfect candidate for the migration to Oracle Cloud Infrastructure as a Service (IaaS).

This appendix demonstrates the shift and lift of an on-premise EDW to Oracle cloud IaaS. In order to support business continuity, the adaption of an online solution for the shift and left to Oracle cloud IaaS is developed. To achieve this, the online shift and lift empowers Oracle GoldenGate 12c downstream technology and Oracle Recovery Manager (RMAN) to transparently move the EDW to Oracle Cloud IaaS.

Nonetheless, to make the shift and lift more applicable to the vast majority of Oracle customers, the move to Oracle cloud IaaS is demonstrated as a two-stop journey, the first stop of the journey is the migration to Oracle cloud IaaS, and the second stop is EDW upgrade from Oracle Database 11g to Oracle Database 12c. After the EDW is upgraded to Oracle Database 12c Release 2, then applying the database features and options transforms the EDW to be fit-for-purpose, which is a high-performance data warehouse engine for intense analytical applications.

The automation of any complex process minimizes the time to design, develop and build by eliminating all kind of human errors during the design, development and build phases. The migration and upgrade of EDW to Oracle cloud IaaS is entirely automated using the author's tool – Near Zero-Downtime Manager (NZDT). The NZDT manager is developed to support three functionalities:

- Migration and Upgrade to Oracle cloud IaaS
- Migration to Oracle cloud IaaS
- Upgrade to Oracle Database 12c



# Architecture

The *online* migration to Oracle Cloud IaaS is a long journey when developed and executed manually. Figure 1 shows a highly automatable process for the migration of EDW to Oracle Cloud IaaS. The components and the process are explained from a high-level and detailed on the next sessions.

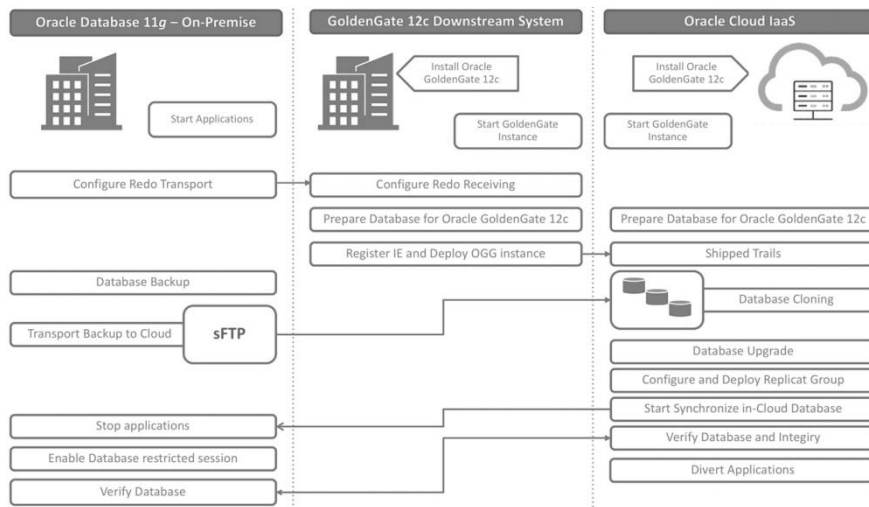


Figure 1: Migration of EDW to Oracle Cloud IaaS

## Installation of Oracle GoldenGate 12c

The migration empowers Oracle GoldenGate 12c downstream technology; hence the software installed on the downstream system and Oracle Cloud IaaS only, and no installation on the on-premise database (source) running Oracle Database 11g.

## Start the Application

The workload simulator designed to generate workload sustainable during the migration project. Refer to Appendix F to learn more about the workload simulator.

## Verify Oracle GoldenGate 12c Instances

Oracle GoldenGate 12c downstream and target instances already started by the installer. The downstream and target instances are running on the designated manager port and have their own root directory.

## Configure Redo-Transport

The redo-transport service ships the transactions logs from the on-premise database to the downstream database mining server. On the downstream the transactions logs processed for generating the Logical Change Records (LCRs) followed by the primary integrated extract (IE) performing change data capture (CDC).

## Prepare the Database for Oracle GoldenGate 12c

Oracle GoldenGate 12c pre-requisites are executed on the source, downstream and target database. This include: Creating administrative database user on source, downstream and target; enabling the supplemental logging on source database. Creating a dedicated tablespace for Oracle GoldenGate is recommended, it ease the administration and management of Oracle GoldenGate metadata.

## Register and Deploy the Primary Integrated Extract

The integrated extract (IE) empowers the Oracle Database 12c mining server to generate the LCRs, deliver high-performance change data capture (CDC), and create the local extract trail. The Extract local trails are shipped to the target system running on Oracle Cloud IaaS.

## Online Database Backup

The online backup is for the purpose of cloning the on-premise database on Oracle Cloud IaaS. It's executed using Oracle Recovery Manager (RMAN). The backup include the database files, archived redo log files and the control file.

## Transport Backup to Oracle Cloud IaaS

Using secure File Transfer Procedure (sFTP), the backup transported to Oracle Cloud IaaS, using slice, ship and verify technique to ensure the backup sets are fully shipped. The Oracle Cloud IaaS system has two Oracle binaries: Oracle Database 11g (matching the release of the source) and Oracle Database 12c for database upgrade.

## Database Clone

The clone is executed using Oracle Recovery Manager (RMAN). Using the backup shipped to Oracle Cloud IaaS, the database cloned in preparation of database upgrade using the Database Upgrade Assistant (DBUA).

## Database Upgrade

Prior to database upgrade, upgrade pre-requisites are executed on the cloned database. This is a standard database upgrade which takes approximately 2 hours. Post upgrade steps ensure the database is ready for pre-production.

## Configure and Deploy the Apply Group

To apply the transactions generated during the projects, the high-performance parallel, intenerated and coordinated Replicat defined. The Replicat reads the remote trails and apply them to the database.

## Synchronized the Oracle Cloud IaaS database

The synchronization begins at a System Change Number (SCN) that point to the last SCN for the archived log files backup sets. Alter the Replicat to begin at this SCN, and then start it to being applying the transaction to the database.

## Stop the Application

This ensures a precise cut-off. Users are stop for the shortest possible duration to ensure data consistency.

## Enable Database Restricted Session

This step ensures only users with restricted session (DBAs) are allowed to connect to the database. This ensures no transactions are entered until the new system is released for users.

## Verify Database Integrity

Use Oracle GoldenGate 12c Veridata to validate volatile tables are consistent. This step may be performed using SQL scripting. However; for multi-lingual database, use of Veridata may discover data related issues.

## Divert Applications

After the applications testing are complete, it's time to divert the application to the new system running on Oracle Cloud IaaS.

# The Near Zero-Downtime Manager

The migration to Oracle Cloud IaaS is executed using the Near Zero-Downtime (NZDT) manager. Prior to the migration execution, basic metadata regarding the source, downstream and target collected. The migration scripts are included and explained for manual migration to Oracle Cloud IaaS in the absence of NZDT Manager, as it's under restricted distribution.

The migration and upgrade is automated using the wizard-based NZDT manager by traversing through 13 steps as listed next:

1. Introduction - Runtime option
2. Database Type and Storage Options
3. On-premise properties
4. Downstream properties
5. Target properties
6. Workload level
7. Install Oracle GoldenGate 12c
8. Configure Redo-Transport
9. Deploy Integrated Extract
10. Database Clone
11. In-Cloud Database Upgrade
12. Create Apply Group and synchronize Database
13. Summary

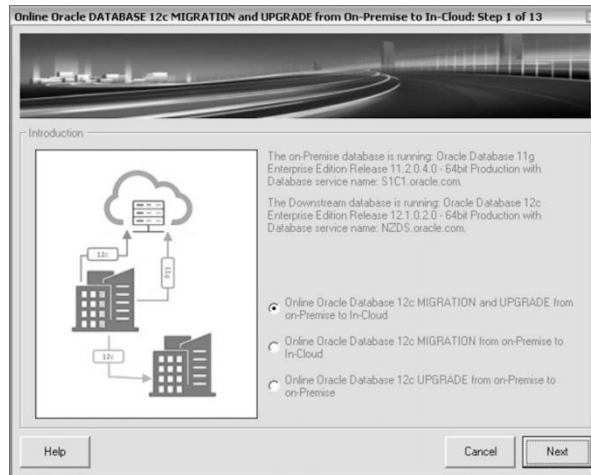
## Login to NZDT Manager

Enter the username and password for the on-premise database and the downstream database to gain access to NZDT Manager. The wizard mode (checked) is a step-by-step to implement the migration and upgrade phases. The Record Session option generates each step executable commands to a file for running on another system.

The screenshot shows the 'Online Oracle DATABASE 12c MIGRATION and UPGRADE for On-Premise and In-Cloud' window. It features two sections for database credentials: 'On-Premise Database' and 'Downstream Database'. Both sections have fields for 'Username' (set to 'system'), 'Password' (masked with asterisks), and 'Connect String' (set to 'S1C1' for on-premise and 'NZDS' for downstream). A 'Use Password Store' checkbox is checked. The right side of the window displays the 'ORACLE GoldenGate 12c' logo and a cloud icon with a server rack. At the bottom, there are checkboxes for 'Use Wizard (for RAC, ASM and in-Cloud support)' (checked) and 'Record Session' (unchecked), along with 'Cancel' and 'OK' buttons. Copyright information for 2017 is visible at the bottom left.

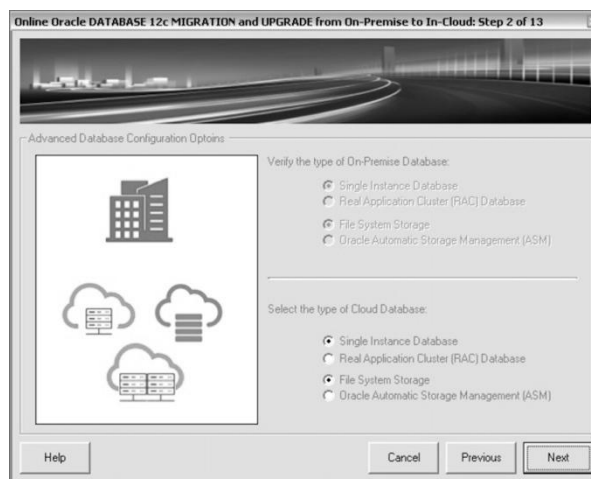
## Step 1: Introduction - Runtime Options

NZDT Manager has three runtime options: Migration and upgrade from on-premise to in-cloud, Migration from on-premise to in-cloud and upgrade from on-premise to on-premise. *Select Migration and upgrade from on-premise to in-cloud.* Click next.



## Step 2: Advanced Database Configuration Options

NZDT manager detects the source database architecture, and default options of identifying the target architecture: Single-instance or multi-instance database, file system or Automatic Storage Management. *Select Single Instance Database, File System Storage.* Click Next.



### Step 3: On-Premise Properties

The properties includes: Host name, IP address, database global name, Oracle database credentials, OS credentials, SYS/SYSTEM password, application credentials, Oracle Home and script location. Check Select from File to seed the data from file.

Online Oracle DATABASE 12c MIGRATION and UPGRADE from On-Premise to In-Cloud: Step 3 of 13

Source System Configuration

On-Premise

Hostname: nz001.oracle.com  
IP Address: 192.168.1.40 ☐ Use DNS

DB Global Name: S1C1.oracle.com  
Oracle Credentials: oracle/oracle  
OS Credentials: root/oracle  
SYS/SYSTEM Password: oracle\_4U/oracle\_4U  
Application Credentials: osm\_repapi/oracle  
Oracle Home: /u01/app/oracle/product/11.2.0/  
Upgrade Scripts: /home/oracle/admin

☒ Seed From File

Help Cancel Previous Next

### Step 4: Downstream Properties

The properties includes: Host name, IP address, database global name, Oracle database credentials, OS credentials, SYS/SYSTEM password, application credentials, Oracle Home, Oracle GoldenGate Home and script location. Check Select from File to seed the data from file.

Online Oracle DATABASE 12c MIGRATION and UPGRADE from On-Premise to In-Cloud: Step 4 of 13

Downstream System Configuration

ORACLE  
GoldenGate 12c

GoldenGate

Hostname: cnzds.oracle.com  
IP Address: 192.168.1.30 ☐ Use DNS

DB Global Name: NZDS.oracle.com  
Oracle Credentials: oracle/oracle  
OS Credentials: root/oracle  
SYS/SYSTEM Password: oracle\_4U/oracle\_4U  
Oracle Home: /u01/app/oracle/product/12.1.0  
GoldenGate Home: /u01/app/oracle/product/12.2.0  
Upgrade Scripts: /home/oracle/admin

☒ Seed From File

Help Cancel Previous Next

## Step 5: In-Cloud Properties

The properties includes: Host name, IP address, database global name, Oracle database credentials, OS credentials, SYS/SYSTEM password, application credentials, Oracle Home, GoldenGate Home and script location. Check Select from File to seed the data from file.

Online Oracle DATABASE 12c MIGRATION and UPGRADE from On-Premise to In-Cloud: Step 5 of 13

Target System Configuration

ORACLE®  
GoldenGate 12c

In-Cloud

Hostname: nzd02.oracle.com  
IP Address: 192.168.1.50 ☐ Use DNS

DB Global Name: T1C1.oracle.com  
Oracle Credentials: oracle/oracle  
OS Credentials: root/oracle  
SYS/SYSTEM Password: oracle\_4U  
Oracle Home: /u01/app/oracle/product/12.1.0  
GoldenGate Home: /u01/app/oracle/product/12.2.0  
Upgrade Scripts: /home/oracle/admin

☒ Seed From File

Help Cancel Previous Next

## Step 6: Starting Workload Simulator

There are four workload levels: *low*, *medium*, *high*, *extreme*. Check skip workload for skipping the workload. Terminate running workload, if any. *Click on START* to begin low workload (default).

Online Oracle DATABASE 12c MIGRATION and UPGRADE from On-Premise to In-Cloud: Step 6 of 13

Application Startup

Select the Workload Level and click START. The workload must remain running throughout the migration and upgrade process. The workload Linux Terminal Window runs on minimized show mode. Before starting a workload, click Terminate Running Workload to kill previously running workloads.

☒ Low  
☐ Medium  
☐ High  
☐ Extreme

☐ Skip workload

[Terminate Running Workload](#)  
[START](#)

Help Cancel Previous Next

The workload simulator is started on the source database by invoking the procedure `CALL_TRANS_LOAD` as illustrated next. To generate enough transactions workload, select a low, medium or high workload, this will generate high-volume transactions for the duration of the migration and upgrade phases.

Refer to Appendix F for more details regarding the Workload Simulator.

Connect to the source database as shown, and then execute on the desired workload.

```
SI1>conn osm_repapi/oracle
Connected.
SI1>
```

## Starting Low Workload

The low workload is suitable for 2-core systems. The elapsed time is 1 second.

```
SQL*Plus: Release 11.2.0.4.0 Production on Fri Mar 23 02:32:37 2018

Copyright (c) 1982, 2013, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing
options

SI1>exec call_trans_load(100000, 1);
```

## Starting Medium Workload

The low workload is suitable for 4-core systems. The elapsed time is 0.1 second.

```
SQL*Plus: Release 11.2.0.4.0 Production on Fri Mar 23 02:34:02 2018

Copyright (c) 1982, 2013, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing
options

SI1>exec call_trans_load(1000000, 0.1);
```

## Starting High Workload

The low workload is suitable for 8-core systems. The elapsed time is 0.01 second.

```
SQL*Plus: Release 11.2.0.4.0 Production on Fri Mar 23 02:35:26 2018
```



```
Copyright (c) 1982, 2013, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing
options

SQL>exec call trans load(10000000, 0.01);
```

## Starting Extreme Workload

The low workload is suitable for 16-core systems. The elapsed time is 0.001 second.

```
SQL*Plus: Release 11.2.0.4.0 Production on Fri Mar 23 02:35:26 2018

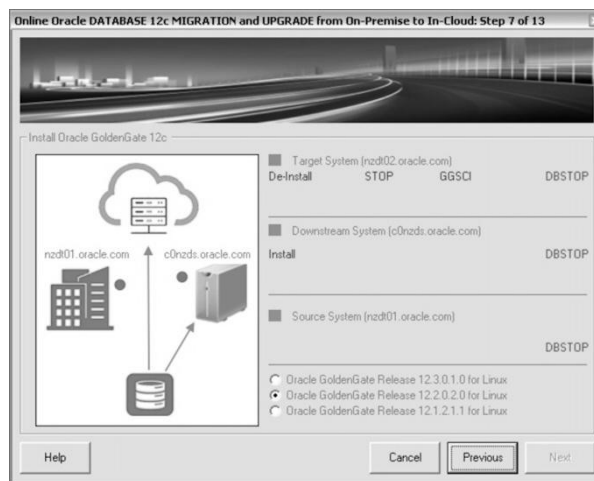
Copyright (c) 1982, 2013, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing
options

SQL>exec call trans load(1000000000, 0.001);
```

## Step 7: Installations

The migration empowers Oracle GoldenGate 12c downstream technology; hence Oracle GoldenGate 12c will be installed on the downstream system and target system only. No installation on the source database system. *Click on the two Install* to begin the installation.



## Install on the Target System

This installs Oracle GoldenGate 12c (12.2) on the target system running Oracle Database 12c (12.1). The silent installation uses a response file – oggcore1.rsp.

```
$/stage/ogg12c/fbo_ggs_Linux_x64_shiphome/Disk1/runInstaller -silent -  
responsefile /home/oracle/oggcore1.rsp -nowelcome -showProgress -force -  
waitforcompletion
```

The Silent install option...

```
oracle.install.responseFileVersion=rsfmt ogginstall response schema v12 1 2  
INSTALL_OPTION=ORA12c  
SOFTWARE_LOCATION=/u01/app/oracle/product/12.2.0/gghome_1  
START_MANAGER=true  
MANAGER_PORT=7840  
DATABASE_LOCATION=/u01/app/oracle/product/12.1.0/dbhome_1  
INVENTORY_LOCATION=/u01/app/oraInventory  
UNIX_GROUP_NAME=oinstall
```

```
/stage/ogg12c/fbo_ggs_Linux_x64_shiphome/Disk1/runInstaller -silent -  
responsefil  
e /home/oracle/oggcore1.rsp -nowelcome -showProgress -force -  
waitforcompletion
```

```
Press Enter to Begin Oracle GoldenGate Silent Install.....  
Do not interrupt the installation while in-progress.....  
It will take approximately 5-10 minutes to complete the install...  
Starting Oracle Universal Installer...
```

```
Checking Temp space: must be greater than 120 MB.    Actual 66236 MB  
Passed  
Checking swap space: must be greater than 150 MB.    Actual 12479 MB  
Passed  
Preparing to launch Oracle Universal Installer from /tmp/OraInstall2018-03-  
23_04  
-56-52AM. Please wait ...You can find the log of this install session at:  
/u01/app/oraInventory/logs/installActions2018-03-23_04-56-52AM.log
```

```
Prepare in progress.  
..... 9% Done.
```

Prepare successful.

```
Copy files in progress.  
..... 55% Done.  
..... 62% Done.  
..... 69% Done.  
..... 74% Done.  
..... 79% Done.  
..... 85% Done.  
.....
```

Copy files successful.

Link binaries in progress.

```

.....
Link binaries successful.

Setup files in progress.
..... 90% Done.

Setup files successful.
The installation of Oracle GoldenGate Core was successful.
Please check '/u01/app/oraInventory/logs/silentInstall2018-03-23_04-56-52AM.log'
for more details.

Configuring and starting manager... in progress.
..... 100% Done.

Configuring and starting manager... successful.
Successfully Setup Software.

```

## Display the Target Instance Properties

Verify the installation on the target by starting GGSCI and display the manager status and the instance directory structure.

```

Oracle GoldenGate Command Interpreter for Oracle
Version 12.2.0.1.1 OGGCORE_12.2.0.1.0_PLATFORMS_151211.1401_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Dec 12 2015 02:56:48
Operating system character set identified as US-ASCII.

Copyright (C) 1995, 2015, Oracle and/or its affiliates. All rights reserved.

GGSCI (c0nzds.oracle.com) 1> info mgr

Manager is running (IP port c0nzds.oracle.com.7840, Process ID 12239).

GGSCI (c0nzds.oracle.com) 2> show

Parameter settings:

SET SUBDIRS      ON
SET DEBUG        OFF

Current directory: /u01/app/oracle/product/12.2.0/gghome_1

Using subdirectories for all process files

Editor: vi

Reports (.rpt)           /u01/app/oracle/product/12.2.0/gghome_1/dirrpt
Parameters (.prm)        /u01/app/oracle/product/12.2.0/gghome_1/dirprm
Replicat Checkpoints (.cpr) /u01/app/oracle/product/12.2.0/gghome_1/dirchk
Extract Checkpoints (.cpe) /u01/app/oracle/product/12.2.0/gghome_1/dirchk
Process Status (.pcs)     /u01/app/oracle/product/12.2.0/gghome_1/dirpcs
SQL Scripts (.sql)        /u01/app/oracle/product/12.2.0/gghome_1/dirsq

```

```
Database Definitions (.def)      /u01/app/oracle/product/12.2.0/gghome_1/dirdef
Dump files (.dmp)              /u01/app/oracle/product/12.2.0/gghome_1/dirdmp
Masterkey wallet files (.wlt)   /u01/app/oracle/product/12.2.0/gghome_1/dirwlt
Credential store files (.crd)    /u01/app/oracle/product/12.2.0/gghome_1/dircrd
```

```
GGSCI (c0nzds.oracle.com) 3>
```

## Install on the Downstream System

This installs Oracle GoldenGate 12c (12.2) on the downstream system running Oracle Database 12c (12.1). The silent installation uses a response file – oggcore1.rsp.

```
$/stage/ogg12c/fbo_ggs_Linux_x64_shiphome/Disk1/runInstaller -silent -
responsefile /home/oracle/oggcore1.rsp -nowelcome -showProgress -force -
waitforcompletion
```

The Silent install option...

```
oracle.install.responseFileVersion=rsfmt ogginstall response schema v12 1 2
INSTALL_OPTION=ORA12c
SOFTWARE_LOCATION=/u01/app/oracle/product/12.2.0/gghome_1
START_MANAGER=true
MANAGER_PORT=7840
DATABASE_LOCATION=/u01/app/oracle/product/12.1.0/dbhome_1
INVENTORY_LOCATION=/u01/app/oraInventory
UNIX_GROUP_NAME=oinstall
```

```
/stage/ogg12c/fbo_ggs_Linux_x64_shiphome/Disk1/runInstaller -silent -
responsefile /home/oracle/oggcore1.rsp -nowelcome -showProgress -force -
waitforcompletion
```

```
Press Enter to Begin Oracle GoldenGate Silent Install.....
Do not interrupt the installation while in-progress.....
It will take approximately 5-10 minutes to complete the install...
Starting Oracle Universal Installer...
```

```
Checking Temp space: must be greater than 120 MB.      Actual 40170 MB
Passed
Checking swap space: must be greater than 150 MB.      Actual 12475 MB
Passed
Preparing to launch Oracle Universal Installer from /tmp/OraInstall2018-03-
23_04
-57-02AM. Please wait ...You can find the log of this install session at:
/u01/app/oraInventory/logs/installActions2018-03-23_04-57-02AM.log
```

```
Prepare in progress.
..... 9% Done.
```

Prepare successful.

```
Copy files in progress.
..... 55% Done.
..... 62% Done.
```

```

..... 69% Done.
..... 74% Done.
..... 79% Done.
..... 85% Done.
.....
Copy files successful.

Link binaries in progress.
.....
Link binaries successful.

Setup files in progress.
..... 90% Done.

Setup files successful.
The installation of Oracle GoldenGate Core was successful.
Please check '/u01/app/oraInventory/logs/silentInstall2018-03-23 04-57-
02AM.log'
for more details.

Configuring and starting manager... in progress.
..... 100% Done.

Configuring and starting manager... successful.
Successfully Setup Software.

```

## Display the Downstream Instance Properties

Verify the installation on the downstream by starting GGSCI and display the manager status and the instance directory structure.

```

Oracle GoldenGate Command Interpreter for Oracle
Version 12.2.0.1.1 OGGCORE_12.2.0.1.0_PLATFORMS_151211.1401_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Dec 12 2015 02:56:48
Operating system character set identified as US-ASCII.

Copyright (C) 1995, 2015, Oracle and/or its affiliates. All rights reserved.

GGSCI (c0nzds.oracle.com) 1> info mgr

Manager is running (IP port c0nzds.oracle.com.7840, Process ID 12239).

GGSCI (c0nzds.oracle.com) 2> show

Parameter settings:

SET SUBDIRS      ON
SET DEBUG        OFF

Current directory: /u01/app/oracle/product/12.2.0/gghome 1

Using subdirectories for all process files

```

```

Editor: vi

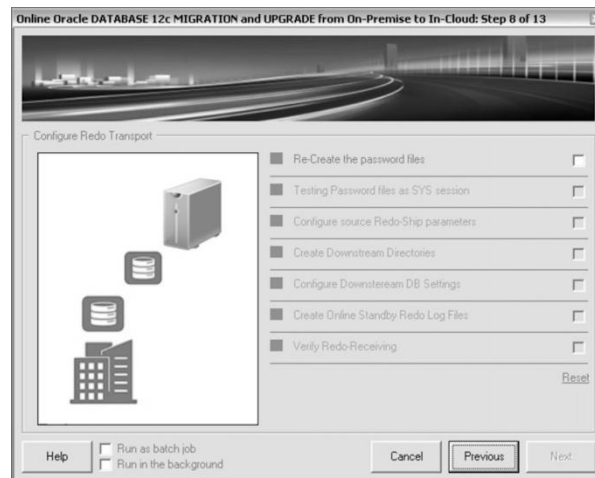
Reports (.rpt)                /u01/app/oracle/product/12.2.0/gghome_1/dirprt
Parameters (.prm)             /u01/app/oracle/product/12.2.0/gghome_1/dirprm
Replicat Checkpoints (.cpr)   /u01/app/oracle/product/12.2.0/gghome_1/dirchk
Extract Checkpoints (.cpe)    /u01/app/oracle/product/12.2.0/gghome_1/dirchk
Process Status (.pcs)         /u01/app/oracle/product/12.2.0/gghome_1/dirpcs
SQL Scripts (.sql)            /u01/app/oracle/product/12.2.0/gghome_1/dirsq1
Database Definitions (.def)    /u01/app/oracle/product/12.2.0/gghome_1/dirdef
Dump files (.dmp)             /u01/app/oracle/product/12.2.0/gghome_1/dirdmp
Masterkey wallet files (.wlt)  /u01/app/oracle/product/12.2.0/gghome_1/dirwlt
Credential store files (.crd)  /u01/app/oracle/product/12.2.0/gghome_1/dircrd

GGSCI (c0nzds.oracle.com) 3>

```

## Step 8: Configure Redo-Transport Service

The redo-transport service ships the source transactions logs to the downstream system. In order to successfully configure redo-transport, the following configurations steps are executed: Re-create the password file, configure redo-transport parameters, create the downstream directories, configure the downstream database parameters, and create the standby redo log files.



### Re-Create the Password File

The password for AS SYSDBA must be identical, re-create the password on the source system, and then copy across to the downstream system with correct file name format and permission. The file format for Linux: orapw<SID>.

```
$export ORACLE_SID=S1C1
```

```

$export ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1

$rm -f $ORACLE_HOME/dbs/orapwS1C1
$
$ORACLE_HOME/bin/orapwd file=$ORACLE_HOME/dbs/orapwS1C1 password=oracle
ignorecase=Y entries=30 force=Y
$
$scp $ORACLE_HOME/dbs/orapwS1C1
oracle@192.168.1.30:/u01/app/oracle/product/12.1.0/dbhome_1/dbs/orapwNZDS
$

```

## Test the Password File

Before proceeding further, it's essential to test the password files by successfully connecting to the source and downstream systems using AS SYSDBA. Use the TNS alias S1C1 for source and T1C1 for target.

```

SQL*Plus: Release 11.2.0.4.0 Production on Fri Mar 23 05:40:40 2018

Copyright (c) 1982, 2013, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing
options

S1C1>exit
Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.4.0
- 64
bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing
options

SQL*Plus: Release 11.2.0.4.0 Production on Fri Mar 23 05:41:44 2018

Copyright (c) 1982, 2013, 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

S1C1>

```

## Configure Redo-Transport Parameters on Source

Set the parameters for sending the transactions logs to the downstream online standby online redo log file, enable the destination and enrolls the instances as part of the data guard configuration.

```

SQL*Plus: Release 11.2.0.4.0 Production on Fri Mar 23 05:44:53 2018

Copyright (c) 1982, 2013, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing
options

S1C1>Prompt "Setting LOG_ARCHIVE_DEST_2 parameter..."
"Setting LOG_ARCHIVE_DEST_2 parameter..."
S1C1>ALTER system SET log_archive_dest_2='SERVICE=nzds SYNC NOREGISTER
VALID_FOR
=(ONLINE_LOGFILES,PRIMARY_ROLE) REOPEN=10 DB_UNIQUE_NAME=nzds' scope=both
2 /

System altered.

S1C1>Prompt "Setting LOG_ARCHIVE_DEST_STATE_2 parameter..."
"Setting LOG_ARCHIVE_DEST_STATE_2 parameter..."
S1C1>ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2=ENABLE scope=both
2 /

System altered.

S1C1>Prompt "Setting LOG_ARCHIVE_CONFIG parameter..."
"Setting LOG_ARCHIVE_CONFIG parameter..."
S1C1>ALTER SYSTEM SET LOG_ARCHIVE_CONFIG='DG_CONFIG=(S1C1, NZDS)' scope=both
2 /

System altered.

S1C1>Prompt "Setting DB parameters complete..."
"Setting DB parameters complete..."
S1C1>

```

## Create the Redo-Log Receiving Directories on Downstream System

Login as oracle user; create the three directories for storing the archived log files, redo log files and standby redo log files on the downstream system.

```

Creating /u01/app/oracle/oradata/NZDS/arch...
$mkdir -p /u01/app/oracle/oradata/NZDS/arch

Creating /u01/app/oracle/oradata/NZDS/redo...
$mkdir -p /u01/app/oracle/oradata/NZDS/redo

Creating /u01/app/oracle/oradata/NZDS/standby...
$mkdir -p /u01/app/oracle/oradata/NZDS/standby

```



## Configure Downstream Database Settings

Set the destinations for the archive log files, the standby archived log files, enable the destinations, and enroll the instances as part of the data guard configuration.

```
SQL*Plus: Release 12.1.0.2.0 Production on Fri Mar 23 06:49:00 2018

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Last Successful login time: Fri Mar 23 2018 01:21:56 +03:00

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
opt
ions

NZDS>Prompt "Setting LOG_ARCHIVE_DEST_1 parameter..."
"Setting LOG_ARCHIVE_DEST_1 parameter..."
NZDS>alter system set
log_archive_dest_1='LOCATION=/u01/app/oracle/oradata/NZDS/
arch VALID_FOR=(ONLINE_LOGFILE,PRIMARY_ROLE)' scope=both
2 /

System altered.

NZDS>Prompt "Setting LOG_ARCHIVE_DEST_2 parameter..."
"Setting LOG_ARCHIVE_DEST_2 parameter..."
NZDS>alter system set
log_archive_dest_2='LOCATION=/u01/app/oracle/oradata/NZDS/
standby VALID_FOR=(STANDBY_LOGFILE,ALL_ROLES)' scope=both
2 /

System altered.

NZDS>Prompt "Setting LOG_ARCHIVE_CONFIG parameter..."
"Setting LOG_ARCHIVE_CONFIG parameter..."
NZDS>ALTER SYSTEM SET LOG_ARCHIVE_CONFIG='DG_CONFIG=(S1C1, NZDS)' scope=both
2 /

System altered.

NZDS>Prompt "Setting LOG_ARCHIVE_DEST_STATE_1 parameter..."
"Setting LOG_ARCHIVE_DEST_STATE_1 parameter..."
NZDS>alter system set log_archive_dest_state_1='ENABLE' scope=both
2 /

System altered.

NZDS>Prompt "Setting LOG_ARCHIVE_DEST_STATE_2 parameter..."
"Setting LOG_ARCHIVE_DEST_STATE_2 parameter..."
NZDS>alter system set log_archive_dest_state_2='ENABLE' scope=both
2 /

System altered.
```

```
NZDS>Prompt "Setting Downstream mining DB parameters complete..."
"Setting Downstream mining DB parameters complete..."
NZDS>
```

## Create Online Standby Redo Log Files

On the downstream, create the online standby redo log files. The number of the online standby redo log files is calculated according to the number of the group on the source database, and starting from the maximum group count plus one. For example, if there are three online groups on the source, then create four online standby redo groups starting at group 4. The size of the of the online standby redo log group should be larger than source online redo log group by approximately 10M.

Table 1 show the configuration, where the source database has three groups with size 50M. The standby database created with four online standby redo log groups with size 60M.

| Source Member Name | Size | Downstream Member Name     | Size |
|--------------------|------|----------------------------|------|
| redo01.log         | 50M  | redo01.log                 | 50M  |
| redo02.log         | 50M  | redo02.log                 | 50M  |
| redo03.log         | 50M  | redo03.log                 | 50M  |
|                    |      | sredo04a.log, sredo04b.log | 60M  |
|                    |      | sredo05a.log, sredo05b.log | 60M  |
|                    |      | sredo06a.log, sredo06b.log | 60M  |
|                    |      | sredo07a.log, sredo07b.log | 60M  |

*Table 1: Configuration of Online Standby Redo Log Groups*

```
SQL*Plus: Release 12.1.0.2.0 Production on Fri Mar 23 06:59:18 2018

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Last Successful login time: Fri Mar 23 2018 06:56:12 +03:00

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
opt
ions

NZDS>set feedback on
NZDS>Prompt "Create standby logfile group 4..."
"Create standby logfile group 4..."
NZDS>alter database drop standby logfile group 4
2 /

Database altered.

NZDS>alter database add standby logfile group 4
```

```

2
('/u01/app/oracle/oradata/NZDS/onlineolog/slog4a.log','/u01/app/oracle/fast_
recovery_area/NZDS/onlineolog/slog4b.log') reuse
3 /

Database altered.

NZDS>Prompt "Create standby logfile group 5..."
"Create standby logfile group 5..."
NZDS>alter database drop standby logfile group 5
2 /

Database altered.

NZDS>alter database add standby logfile group 5
2
('/u01/app/oracle/oradata/NZDS/onlineolog/slog5a.log','/u01/app/oracle/fast_
recovery_area/NZDS/onlineolog/slog5b.log') reuse
3 /

Database altered.

NZDS>Prompt "Create standby logfile group 6..."
"Create standby logfile group 6..."
NZDS>alter database drop standby logfile group 6
2 /

Database altered.

NZDS>alter database add standby logfile group 6
2
('/u01/app/oracle/oradata/NZDS/onlineolog/slog6a.log','/u01/app/oracle/fast_
recovery_area/NZDS/onlineolog/slog6b.log') reuse
3 /

Database altered.

NZDS>Prompt "Create standby logfile group 7..."
"Create standby logfile group 7..."
NZDS>alter database drop standby logfile group 7
2 /

Database altered.

NZDS>alter database add standby logfile group 7
2
('/u01/app/oracle/oradata/NZDS/onlineolog/slog7a.log','/u01/app/oracle/fast_
recovery_area/NZDS/onlineolog/slog7b.log') reuse
3 /

Database altered.

NZDS>Prompt "Standby logfile group complete..."
"Standby logfile group complete..."
NZDS>select group#,thread#,sequence#,archived,status
2 from V$STANDBY_LOG
3 /

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

4 rows selected.

NZDS>

```

## Verify Redo-Transport

To verify the redo-transport is successfully shipping the transactions logs to the downstream system, open two sessions on the source database. On the first session connect to the database AS SYSDBA and issue the SQL ALTER command:

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

On the second session, tail the instance alert file – alertS1C1.log. It must display the line below; which indicates the source is successfully shipping the transactions logs to the downstream system using the specified destination.

```

*****
LGWR: Setting 'active' archival for destination LOG_ARCHIVE_DEST_2
*****

```

```

SQL*Plus: Release 11.2.0.4.0 Production on Fri Mar 23 07:01:56 2018

Copyright (c) 1982, 2013, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing
options

System altered.

System altered.

System altered.

S1C1>

```

```

017 04 18/01 mf 1 924 dhdrkn2h .arc
*****
LGWR: Setting 'active' archival for destination LOG_ARCHIVE_DEST_2
*****
LGWR: Standby redo logfile selected for thread 1 sequence 1533 for
destination L

```

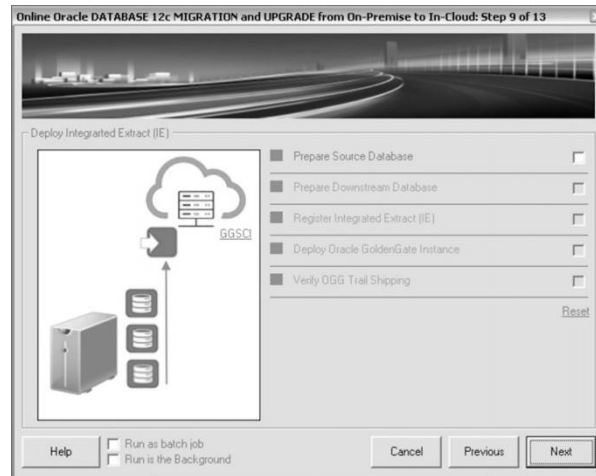
```

OG ARCHIVE DEST 2
Thread 1 advanced to log sequence 1533 (LGWR switch)
Current log# 3 seq# 1533 mem# 0: /u01/app/oracle/oradata/S1C1/redo03.log
Archived Log entry 3085 added for thread 1 sequence 1532 ID 0xbb8ed11 dest
1:

```

## Step 9: Deploy Integrated Extract

The deployment of the integrated extract (IE) begins by performing pre-requisites on the source database, registering the extract with the database, performing pre-requisites on the downstream database, deploy the integrated extract and verify extract trails and remote trails are created.



### Prepare the Source Database

In the source database perform the following: Enable GoldenGate replication, add supplemental log, create dedicated tablespace (OGG\_DAT) and database user (OGG\_ADMIN) for GoldenGate.

```

SQL*Plus: Release 11.2.0.4.0 Production on Fri Mar 23 08:18:20 2018
Copyright (c) 1982, 2013, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing
options

S1C1>set feedback on
S1C1>alter system set enable_goldengate_replication=true scope=both

```

```

2 /

System altered.

S1C1>alter database add supplemental log data
2 /

Database altered.

S1C1>drop tablespace ogg_dat including contents and datafiles
2 /

Tablespace dropped.

S1C1>create tablespace ogg_dat
2 datafile '/u01/app/oracle/oradata/S1C1/ogg_dat01.dbf'
3 size 100M
4 autoextend on
5 /

Tablespace created.

S1C1>drop user ogg_admin cascade
2 /

User dropped.

S1C1>grant dba to ogg_admin identified by oracle
2 /

Grant succeeded.

S1C1>alter user ogg_admin
2 default tablespace ogg_dat
3 quota unlimited on ogg_dat
4 profile default
5 /

User altered.

S1C1>

```

## Prepare the Downstream Database

In the downstream database perform the following: Enable GoldenGate replication, add supplemental log, create dedicated tablespace (OGG\_DAT) and database user (OGG\_ADMIN) for GoldenGate. Grant additional privileges using DBMS\_GOLDENGATE\_AUTH, this grant the user OGG\_ADMIN select privilege on V\$DATABASE data dictionary view.

```

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

```

```

NZDS>set feedback on
NZDS>alter system set enable_goldengate_replication=true scope=both
2 /

System altered.

NZDS>alter system set streams_pool_size=1G scope=both
2 /

System altered.

NZDS>drop tablespace ogg_dat including contents and datafiles
2 /

Tablespace dropped.

NZDS>create tablespace ogg_dat
2 datafile '/u01/app/oracle/oradata/NZDS/datafile/ogg_dat01.dbf'
3 size 100M
4 autoextend on
5 /

Tablespace created.

NZDS>drop user ogg_admin cascade
2 /

User dropped.

NZDS>grant dba to ogg_admin identified by oracle
2 /

Grant succeeded.

NZDS>alter user ogg_admin
2 default tablespace ogg_dat
3 quota unlimited on ogg_dat
4 profile default
5 /

User altered.

NZDS>begin
2 DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE
3 (grantee=>'OGG_ADMIN',
4 privilege_type=>'capture',
5 grant_select_privileges=>true,
6 do_grants=>TRUE);
7 end;
8 /

PL/SQL procedure successfully completed.

NZDS>prompt "Downstream Database Prepared..."
"Downstream Database Prepared..."
NZDS>

```

## Register the Integrated Extract

The integrated extract must be registered with the database; this requires logging to the source database (S1C1) and downstream mining database (NZDS). Verify the integrated extract is registered by querying DBA\_CAPTURE views on the downstream database.

```
Oracle GoldenGate Command Interpreter for Oracle
Version 12.2.0.1.1 OGGCORE 12.2.0.1.0 PLATFORMS 151211.1401_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Dec 12 2015 02:56:48
Operating system character set identified as US-ASCII.

Copyright (C) 1995, 2015, Oracle and/or its affiliates. All rights reserved.

GGSCI (c0nzds.oracle.com) 1>
GGSCI (c0nzds.oracle.com) 2> dblogin USERID ogg_admin@S1C1, PASSWORD oracle

Successfully logged into database.

GGSCI (c0nzds.oracle.com as ogg_admin@S1C1) 3> miningdblogin userid
ogg_admin@NZDS, password oracle

Successfully logged into mining database.

GGSCI (c0nzds.oracle.com as ogg_admin@S1C1) 4> unregister extract ES1C101,
database

2018-03-23 08:24:35 ERROR OGG-08222 EXTRACT ES1C101 must be registered
with the database to perform this operation.

GGSCI (c0nzds.oracle.com as ogg_admin@S1C1) 5> register extract ES1C101,
database

2018-03-23 08:24:48 INFO OGG-02003 Extract ES1C101 successfully
registered
with database at SCN 61246810.

GGSCI (c0nzds.oracle.com as ogg_admin@S1C1) 6> shell
/home/oracle/admin/dba_capture.sh

SQL*Plus: Release 12.1.0.2.0 Production on Fri Mar 23 08:24:49 2018

Copyright (c) 1982, 2014, Oracle. All rights reserved.

NZDS>Connected.
NZDS> 2
CAPTURE_NAME
-----
```



```

START_SCN STATUS
-----
OGG$CAP_ES1C101
61246810 ENABLED

NZDS>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

```

Add table-level supplemental logging while still connected to the source database (S1C1).

```

GGSCI (c0nzds.oracle.com as ogg_admin@S1C1) 7> add trandata osm_repapi.*

Logging of supplemental redo log data is already enabled for table
OSM_REPAPI.CUSTOMERS.

TRANDATA for instantiation CSN has been added on table
'OSM_REPAPI.CUSTOMERS'.
Logging of supplemental redo log data is already enabled for table
OSM_REPAPI.DISCOUNT_TYPES.

TRANDATA for instantiation CSN has been added on table
'OSM_REPAPI.DISCOUNT TYPE
S'.
2018-03-23 08:24:53 WARNING OGG-06439 No unique key is defined for table
OSM_REPAPI.DOCUMENTS. All viable columns will be used to represent the key, but may not
guarantee uniqueness. KEYCOLS may be used to define the key.

Logging of supplemental redo log data is already enabled for table
OSM_REPAPI.OSM_DOCUMENTS.

TRANDATA for instantiation CSN has been added on table
'OSM_REPAPI.OSM_DOCUMENTS
'.
Logging of supplemental redo log data is already enabled for table
OSM_REPAPI.POLICIES.

TRANDATA for instantiation CSN has been added on table
'OSM_REPAPI.POLICIES'.
Logging of supplemental redo log data is already enabled for table
OSM_REPAPI.POLICY_TYPES.

TRANDATA for instantiation CSN has been added on table
'OSM_REPAPI.POLICY TYPES'

```

```

.
Logging of supplemental redo log data is already enabled for table
OSM_REPAPI.RA
NDOM_DOMAIN.

TRANSDATA for instantiation CSN has been added on table
'OSM_REPAPI.RANDOM DOMAIN'
.
Logging of supplemental redo log data is already enabled for table
OSM_REPAPI.RA
NDOM_FIRST.

TRANSDATA for instantiation CSN has been added on table
'OSM_REPAPI.RANDOM_FIRST'
.
Logging of supplemental redo log data is already enabled for table
OSM_REPAPI.RA
NDOM_LAST.

TRANSDATA for instantiation CSN has been added on table
'OSM_REPAPI.RANDOM_LAST'.

Logging of supplemental redo log data is already enabled for table
OSM_REPAPI.RA
NDOM_MIDDLE.

TRANSDATA for instantiation CSN has been added on table
'OSM_REPAPI.RANDOM MIDDLE'
.
GGSCI (c0nzds.oracle.com as ogg_admin@S1C1) 8>

```

## Deploy Oracle GoldenGate Instance

The instance manager is running, to deploy the Oracle GoldenGate 12c instance on the downstream system by performing the following: configure and define the primary extract, configure and define the secondary extract (data pump).

### Configure the Primary Extract

```

EXTRACT ES1C101
SETENV (ORACLE_SID = NZDS)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
SETENV (TNS_ADMIN = "/u01/app/oracle/product/12.1.0/dbhome_1/network/admin")
USERID ogg_admin@S1C1, PASSWORD oracle
tranlogoptions mininguser ogg_admin@NZDS, miningpassword oracle
tranlogoptions integratedparams (max_sga_size 128, downstream_real_time_mine
Y)

EXTTRAIL ./dirdat/sa
TABLE osm_repapi.*;

```

## Configure the Secondary Extract

```
EXTRACT PS1C101
SETENV (ORACLE_SID = NZDS)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@S1C1, PASSWORD oracle
RMTHOST 192.168.1.62, MGRPORT 7840
RMTTRAIL ./dirdat/ta
TABLE osm repapi.*;
```

## Define the Primary Extract and Extract's Trail

```
Oracle GoldenGate Command Interpreter for Oracle
Version 12.2.0.1.1 OGGCORE_12.2.0.1.0_PLATFORMS_151211.1401_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Dec 12 2015 02:56:48
Operating system character set identified as US-ASCII.

Copyright (C) 1995, 2015, Oracle and/or its affiliates. All rights reserved.

GGSCI (c0nzds.oracle.com) 1>
GGSCI (c0nzds.oracle.com) 2> add extract ES1C101, integrated tranlog, begin now

EXTRACT (Integrated) added.

GGSCI (c0nzds.oracle.com) 3> add exttrail ./dirdat/sa, extract ES1C101, megabytes 100

EXTTRAIL added.
```

## Define the Secondary Extract and Remote Trail

```
GGSCI (c0nzds.oracle.com) 4> add extract PS1C101, extrailsources ./dirdat/sa

EXTRACT added.

GGSCI (c0nzds.oracle.com) 5> add rmttrail ./dirdat/ta, extract PS1C101, megabytes 100

RMTTRAIL added.
```

## Check Status and Start Extracts

```
GGSCI (c0nzds.oracle.com) 6> info all
```

```

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
EXTRACT      STOPPED      ES1C101    00:00:00      00:00:04
EXTRACT      STOPPED      PS1C101    00:00:00      00:00:02

GGSCI (c0nzds.oracle.com) 7> start *

Sending START request to MANAGER ...
EXTRACT ES1C101 starting

Sending START request to MANAGER ...
EXTRACT PS1C101 starting

GGSCI (c0nzds.oracle.com) 8> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
EXTRACT      STARTING     ES1C101    00:00:00      00:00:06
EXTRACT      RUNNING      PS1C101    00:00:00      00:00:04

GGSCI (c0nzds.oracle.com) 9> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
EXTRACT      STARTING     ES1C101    00:00:00      00:00:07
EXTRACT      RUNNING      PS1C101    00:00:00      00:00:05

GGSCI (c0nzds.oracle.com) 10> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
EXTRACT      STARTING     ES1C101    00:00:00      00:00:08
EXTRACT      RUNNING      PS1C101    00:00:00      00:00:06

GGSCI (c0nzds.oracle.com) 11>

```

## Verify Remote Trail Shipping

It's essential to verify the remote trails are sent successfully to the target instance. Navigate to `./dirdat/` directory and list trail files; they should be accumulating because the workload simulator is running on the source database.

```
total 8
drwxr-x--- 2 oracle oinstall 4096 Mar 23 04:57 .
drwxr-xr-x 26 oracle oinstall 4096 Mar 23 04:57 ..
Wait...
total 8
drwxr-x--- 2 oracle oinstall 4096 Mar 23 04:57 .
drwxr-xr-x 26 oracle oinstall 4096 Mar 23 04:57 ..
Wait...
total 8
drwxr-x--- 2 oracle oinstall 4096 Mar 23 04:57 .
drwxr-xr-x 26 oracle oinstall 4096 Mar 23 04:57 ..
Wait...
```

## Step 10: Database Clone

As part of database clone, the tasks executed are: Perform online database backup from on-premise (target), transport the backup to Oracle cloud IaaS (auxiliary instance), clone the database using RMAN DUPLICATE command and identify and send the clone SCN to Oracle cloud IaaS instance.



There are two techniques to clone an Oracle database:

- Clone the database from backup
- Clone the database from an active database

The source database is referred to as the *target*, and the cloned database referred to as the *auxiliary*. Next, cloning the database from the most recent backup of the database adapted. The backup composed of the database file, the current control file and the list of associated archived log files. The backup is transported to the auxiliary instance in preparation of running RMAN DUPLICATE command.

## Perform Online Database Backup from On-Premise

An online full database backup composed of: the database files, archived redo log files, controlfile and server parameter file (SPFILE). It's conducted using Oracle Recovery Manager (RMAN). However; it's very essential to explicitly specify the maximum size of the back piece to suite the network bandwidth, the reason when transporting the backup to Oracle cloud IaaS, in the event of transport failure only the last failed file is re-transported. The script below indicates the maximum backup set piece size is 512M.

```
Recovery Manager: Release 11.2.0.4.0 - Production on Fri Mar 23 10:12:46
2018
```

```
Copyright (c) 1982, 2011, Oracle and/or its affiliates. All rights
reserved.
```

```
connected to target database: S1C1 (DBID=196667409)
```

```
RMAN> CROSSCHECK BACKUP;
2> CONFIGURE RETENTION POLICY TO REDUNDANCY 1;
3> RUN {
4> ALLOCATE CHANNEL c1 TYPE DISK maxpiecesize 512M;
5> ALLOCATE CHANNEL c2 TYPE DISK maxpiecesize 512M;
6> ALLOCATE CHANNEL c3 TYPE DISK maxpiecesize 512M;
7> delete noprompt backup;
8> BACKUP DATABASE INCLUDE CURRENT CONTROLFILE
9> FORMAT '/home/oracle/BACKUP/IaaSDBF_%U.dbf';
10> SQL 'ALTER SYSTEM SWITCH LOGFILE';
11> BACKUP ARCHIVELOG ALL
12> FORMAT '/home/oracle/BACKUP/IaaSARC_%U.arc';
13> #delete force obsolete;
14> RELEASE CHANNEL c1;
15> RELEASE CHANNEL c2;
16> RELEASE CHANNEL c3;
17> }
```

```
piece handle=/home/oracle/BACKUP/IaaSARC_gisugcei_2_1.arc
tag=TAG20180323T101305 comment=NONE
channel c3: backup set complete, elapsed time: 00:00:09
Finished backup at 23-MAR-18
```

```
Starting Control File and SPFILE Autobackup at 23-MAR-18
piece
handle=/u01/app/oracle/fast_recovery_area/S1C1/autobackup/2018_03_23/o1_mf_s
_971518427_fc9b9cx8_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 23-MAR-18
```

```
released channel: c1
```

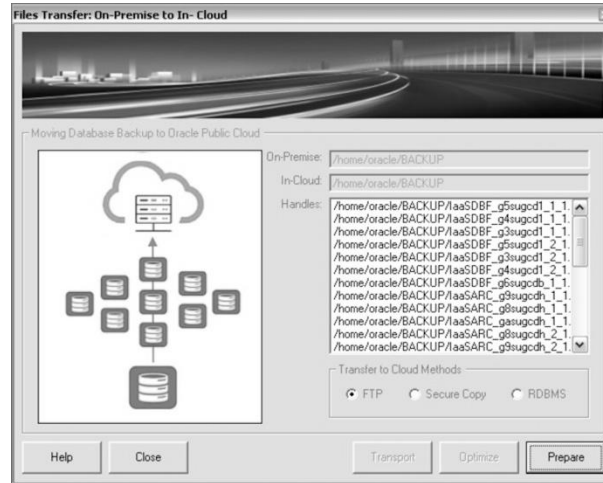
```
released channel: c2
```

```
released channel: c3
```

```
Recovery Manager complete.
```

# Transport Database Backup to Oracle Cloud IaaS

Transport the backup sets to Oracle cloud IaaS. Select FTP (default), and then click on Prepare, Optimize and Transport. The backup set files (handlers) are sent to the directory specified by the In-Cloud directory.



Ensure the secure FTP daemon is running the source and target systems.

```
[root@nzd01 admin]# /sbin/service vsftpd restart
Shutting down vsftpd: [ OK ]
Starting vsftpd for vsftpd: [ OK ]
[root@nzd01 admin]#
```

```
[root@nzd02 admin]# /sbin/service vsftpd restart
Shutting down vsftpd: [ OK ]
Starting vsftpd for vsftpd: [ OK ]
[root@nzd02 admin]#
```

Click on the handler list to view the list of files to be transferred from the source system to the target system on Oracle cloud IaaS.

## Clone Database

The clone empowers RMAN DUPLICATE command to clone the database in preparation for database upgrade.

```
RMAN> CONFIGURE DEVICE TYPE DISK PARALLELISM 3;
2> RUN {
3> DUPLICATE TARGET DATABASE TO T1C1
4> BACKUP LOCATION '/home/oracle/BACKUP'
```

```

5> NOFILENAMECHECK;
6> }
7>
8>
using target database control file instead of recovery catalog
old RMAN configuration parameters:
CONFIGURE DEVICE TYPE DISK PARALLELISM 3 BACKUP TYPE TO BACKUPSET;
new RMAN configuration parameters:
CONFIGURE DEVICE TYPE DISK PARALLELISM 3 BACKUP TYPE TO BACKUPSET;
new RMAN configuration parameters are successfully stored

Starting Duplicate Db at 23-MAR-18
allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: SID=171 device type=DISK
allocated channel: ORA_AUX_DISK_2
channel ORA_AUX_DISK_2: SID=10 device type=DISK
allocated channel: ORA_AUX_DISK_3
channel ORA_AUX_DISK_3: SID=172 device type=DISK

contents of Memory Script:
{
  sql clone "create spfile from memory";
}
executing Memory Script

sql statement: create spfile from memory

contents of Memory Script:
{
  shutdown clone immediate;
  startup clone nomount;
}
executing Memory Script

Oracle instance shut down

connected to auxiliary database (not started)
Oracle instance started

Total System Global Area      229683200 bytes

Fixed Size                     2251936 bytes
Variable Size                  171967328 bytes
Database Buffers               50331648 bytes
Redo Buffers                   5132288 bytes

contents of Memory Script:
{
  sql clone "alter system set  db_name =
'S1C1' comment=
'Modified by RMAN duplicate' scope=spfile";
  sql clone "alter system set  db_unique_name =
'T1C1' comment=
'Modified by RMAN duplicate' scope=spfile";
  shutdown clone immediate;
  startup clone force nomount
  restore clone primary controlfile;
  alter clone database mount;
}

```



```

executing Memory Script

sql statement: alter system set db_name = 'S1C1' comment= 'Modified by
RMAN
duplicate'' scope=spfile

sql statement: alter system set db unique name = 'T1C1' comment=
'Modified
by RMAN duplicate'' scope=spfile

Oracle instance shut down

Oracle instance started

Total System Global Area      229683200 bytes

Fixed Size                    2251936 bytes
Variable Size                 171967328 bytes
Database Buffers              50331648 bytes
Redo Buffers                   5132288 bytes

Starting restore at 23-MAR-18
allocated channel: ORA_AUX_DISK_1
channel ORA_AUX_DISK_1: SID=170 device type=DISK
allocated channel: ORA_AUX_DISK_2
channel ORA_AUX_DISK_2: SID=10 device type=DISK
allocated channel: ORA_AUX_DISK_3
channel ORA_AUX_DISK_3: SID=171 device type=DISK

channel ORA_AUX_DISK_1: starting datafile backup set restore
channel ORA_AUX_DISK_1: restoring control file
channel ORA_AUX_DISK_1: reading from backup piece
/u01/app/oracle/fast_recovery_
area/S1C1/autobackup/2018_03_23/ol_mf_s_971518427_fc9b9cx8 .bkp
channel ORA_AUX_DISK_1: ORA-19870: error while restoring backup piece
/u01/app/o
racle/fast_recovery_area/S1C1/autobackup/2018_03_23/ol_mf_s_971518427_fc9b9c
x8_
bkp
ORA-19505: failed to identify file
"/u01/app/oracle/fast_recovery_area/S1C1/auto
backup/2018_03_23/ol_mf_s_971518427_fc9b9cx8 .bkp"
ORA-27037: unable to obtain file status
Linux-x86_64 Error: 2: No such file or directory
Additional information: 3

failover to previous backup

channel ORA_AUX_DISK_1: starting datafile backup set restore
channel ORA_AUX_DISK_1: restoring control file
channel ORA_AUX_DISK_1: reading from backup piece
/u01/app/oracle/fast_recovery_
area/S1C1/autobackup/2018_03_23/ol_mf_s_971518381_fc9b7xn6 .bkp
channel ORA_AUX_DISK_1: ORA-19870: error while restoring backup piece
/u01/app/o
racle/fast_recovery_area/S1C1/autobackup/2018_03_23/ol_mf_s_971518381_fc9b7x
n6
bkp

```

```

ORA-19505: failed to identify file
"/u01/app/oracle/fast_recovery_area/S1C1/auto
backup/2018_03_23/o1_mf_s_971518381_fc9b7xn6_.bkp"
ORA-27037: unable to obtain file status
Linux-x86 64 Error: 2: No such file or directory
Additional information: 3

failover to previous backup

channel ORA_AUX_DISK_1: starting datafile backup set restore
channel ORA_AUX_DISK_1: restoring control file
channel ORA_AUX_DISK_1: reading from backup piece
/home/oracle/BACKUP/IaaSDBF g6
sugcdb_1_1.dbf
channel ORA_AUX_DISK_1: piece
handle=/home/oracle/BACKUP/IaaSDBF_g6sugcdb_1_1.db
f tag=TAG20180323T101249
channel ORA_AUX_DISK_1: restored backup piece 1
channel ORA_AUX_DISK_1: restore complete, elapsed time: 00:00:01
output file name=/u01/app/oracle/product/11.2.0/dbhome_1/dbs/cntrlT1C1.dbf
Finished restore at 23-MAR-18

database mounted

contents of Memory Script:
{
  set until scn 61255535;
  set newname for datafile 1 to
"/u01/app/oracle/oradata/T1C1/system01.dbf";
  set newname for datafile 2 to
"/u01/app/oracle/oradata/T1C1/sysaux01.dbf";
  set newname for datafile 3 to
"/u01/app/oracle/oradata/T1C1/undotbs01.dbf";
  set newname for datafile 4 to
"/u01/app/oracle/oradata/T1C1/users01.dbf";
  set newname for datafile 5 to
"/u01/app/oracle/oradata/T1C1/ogg_dat01.dbf";
  restore
  clone database
  ;
}
executing Memory Script

```

## Send the Clone System Change Number to Target

This step is to allow the synchronization begins at the SCN of the clone operation, which is the SCN of the last archive log backup piece. From the DUPLICA log file, search for the SCN with the format: `set until scn 61255535;`

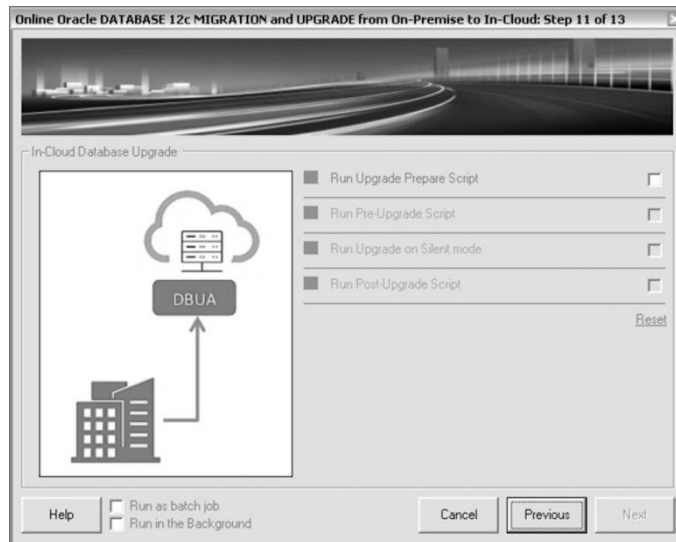
## Step 11: Database Upgrade

From the cloned database running on Oracle cloud IaaS, the database upgrade performed. There are two methods to upgrade the database: Database Upgrade

Assistant (DBUA) graphical user interface and DBUA silent mode. The upgrade goes four stages:

- Prepare the database for upgrade
- Database pre-upgrade
- Database upgrade
- Database post-upgrade

This section performs the database upgrade empowering DBUA silent mode option. The silent mode uses a response file as an input to execute the database upgrade. The database upgrade updates the database dictionary; hence it's an offline operation conducted on the clone database rather than the source database where users are connected and running business transactions.



The database upgrade path is selected based on business requirements. For business operating 24/7, the database must remain open during the database upgrade. There are options to deliver online database upgrade, however; empower Oracle GoldenGate is Oracle best practice for achieving near zero-downtime for database upgrade.

Figure 2 show the migration path for an Oracle database. For near zero-time, it clearly shows Oracle GoldenGate 12c is the option.

There are two modes to run the Oracle Database Upgrade Assistant (DBUA), Graphical User Interface (GUI) or silent mode. Though Oracle recommends using the GUI option of the DBUA for ease of use, the following database upgrade is

conducted using the silent mode option. The primary reason of using the silent mode is the absence of graphical console, instead using a remote X-Windows terminal.

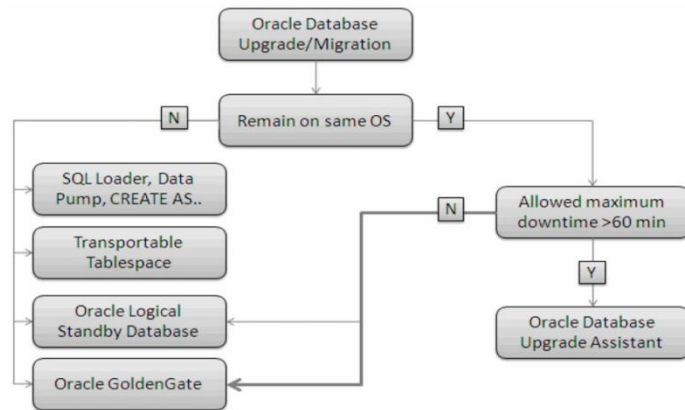


Figure 2: Oracle Database Upgrade Path

Because the silent mode is empowered to upgrade the database, the following task must be performed:

- Prepare the database
- Database pre-upgrade
- Database upgrade
- Database post-upgrade

## Prepare the Database for Upgrade

The prepare database upgrade script is executed from Oracle Database 12c (new home) while connected to Oracle Database 11g (old home) database. Review the output generated and follows the recommended actions. Recommended tasks must be executed on the source database prior to running the database pre-upgrade.

```
SQL*Plus: Release 12.1.0.2.0 Production on Fri Mar 23 10:45:51 2018
```

```
Copyright (c) 1982, 2014, Oracle. All rights reserved.
```

```
SQL> Connected.
```

```
SQL>
```

```
Loading Pre-Upgrade Package...
```

```
Executing Pre-Upgrade Checks in T1C1...
```

```
*****
```

```
==> ERRORS FOUND for T1C1 <==
```

```
The following are *** ERROR LEVEL CONDITIONS *** that must be addressed  
prior to attempting your upgrade.  
Failure to do so will result in a failed upgrade.
```

```
You MUST resolve the above errors prior to upgrade
```

```
*****
```

```
*****
```

```
==> PRE-UPGRADE RESULTS for T1C1 <==
```

```
ACTIONS REQUIRED:
```

```
3. Review results of the pre-upgrade checks:
```

```
/u01/app/oracle/product/11.2.0/dbhome_1/cfgtoollogs/T1C1/preupgrade/preupgra  
de.  
Log
```

```
2. Execute in the SOURCE environment BEFORE upgrade:
```

```
/u01/app/oracle/product/11.2.0/dbhome_1/cfgtoollogs/T1C1/preupgrade/preupgra  
de_  
fixups.sql
```

```
3. Execute in the NEW environment AFTER upgrade:
```

```
/u01/app/oracle/product/11.2.0/dbhome_1/cfgtoollogs/T1C1/postupgr  
ade_  
_fixups.sql
```

```
*****
```

```
Pre-Upgrade Checks in T1C1 Completed.
```

```
SQL> Disconnected from Oracle Database 11g Enterprise Edition Release  
11.2.0.4.0
```

```
- 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing  
options
```

## Database Pre-Upgrade

The clone database is now ready for upgrade. However; before executing the upgrade program, the following is highly recommended to run:

- Gather data dictionary
- Enlarge the fast recovery area (FRA)
- Adjust the open cursors parameter OPEN\_CURSORS

```
SQL*Plus: Release 11.2.0.4.0 Production on Fri Mar 23 10:47:52 2018

Copyright (c) 1982, 2013, Oracle. All rights reserved.

SQL> Connected.
SQL> SQL> set feedback on
SQL> begin
  2      dbms_stats.gather_dictionary_stats;
  3  end;
  4  /

PL/SQL procedure successfully completed.

SQL> alter system set db_recovery_file_dest_size=10000M scope=both
  2  /

System altered.

SQL> alter system set open_cursors=1000 scope=both
  2  /

System altered.

SQL>
SQL> Disconnected from Oracle Database 11g Enterprise Edition Release
11.2.0.4.0
- 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing
options
```

## Upgrade Database

The database upgrade executed on silent mode. Invoke the dbua program from the new oracle home as shown below. Regardless of the database size, the timeframe of the database upgrade is approximately 90 minutes. To speed up the database upgrade processing, raise the value for the attribute DEGREE\_OF\_PARALLELISM, which specifies the degree of parallelism when executing database related tasks.

Because the database time-zone is release specific, it's essential to include the time zone upgrade attribute.

```
/u01/app/oracle/product/12.1.0/dbhome_1/bin/dbua -silent -sid T1C1 \  
-oracleHome /u01/app/oracle/product/11.2.0/dbhome_1 \  
-autoextendFiles \  
-recompile_invalid_objects true \  
-degree of parallelism 2 \  
-emConfiguration NONE \  
-sysDBAUserName sys \  
-sysDBAPassword oracle_4U \  
-upgradeTimezone
```

```
Log files for the upgrade operation are located at:  
/u01/app/oracle/cfgtoollogs/dbua/T1C1/upgrade48  
Performing Pre Upgrade  
4% complete  
20% complete  
Performing RDBMS Upgrade  
20% complete  
...  
...  
60% complete  
Upgrading Timezone  
80% complete  
Generating Summary  
Database upgrade has been completed successfully, and the database is ready  
to use.  
100% complete  
Check the log file  
"/u01/app/oracle/cfgtoollogs/dbua/logs/silent.log_1521791455863" for upgrade  
details.
```

## Database Post-Upgrade

Though the Enterprise Data Warehouse (EDW) has less post-upgrade procedure, the general database post-upgrade involves the following:

- Verify database options are all valid
- Verify database object are all valid
- Technical testing completed and satisfactory
- Application testing completed and user's acceptance sign
- Validate the backup and restore procedures
- Gradually apply Oracle Database 12c new features associated with EDW such as in-memory, materialized view, advanced compression and partitioning.

It's recommended to empower Oracle Real Application Testing (RAT) for technical testing; RAT provides automation and scalability testing. Enable database flashback before the application testing initiated; a database rewind is performed after completion of the applications testing and preparing for database synchronization.

## Step 12: Create Apply Group and Synchronize Target Database

The final step is to synchronize the target database with the transactions logs that have been occurring since the timestamp of completing the database clone. This step composed of the following tasks: Create the apply group (Replicat), Synchronize the in-cloud database, stop the application, enable restricted session, verify data synchronization and divert the application to the Enterprise Data Warehouse (EDW).



### Create Apply Group

The apply group (Replicat) created using the option: coordinated, parallel and integrated.

```
Oracle GoldenGate Command Interpreter for Oracle
Version 12.2.0.1.1 OGGCORE 12.2.0.1.0 PLATFORMS 151211.1401 FBO
Linux, x64, 64bit (optimized), Oracle 12c on Dec 12 2015 02:56:48
Operating system character set identified as US-ASCII.

Copyright (C) 1995, 2015, Oracle and/or its affiliates. All rights reserved.

GGSCI (nzdt02.oracle.com) 1>
GGSCI (nzdt02.oracle.com) 2> dblogin userid ogg_admin@T1C1, password oracle
```



```

Successfully logged into database.

GGSCI (nzdt02.oracle.com as ogg_admin@T1C1) 3> delete checkpointtable!

No checkpoint table specified. Using GLOBALS specification
(ogg admin.checkpttab
)...
ERROR: OCI Error ORA-00942: table or view does not exist (status = 942).
Droppin
g checkpoint table ogg admin.checkpttab, SQL <DROP TABLE
ogg_admin.checkpttab>.

GGSCI (nzdt02.oracle.com as ogg_admin@T1C1) 4> add checkpointtable

No checkpoint table specified. Using GLOBALS specification
(ogg_admin.checkpttab
)...

Successfully created checkpoint table ogg_admin.checkpttab.

GGSCI (nzdt02.oracle.com as ogg admin@T1C1) 5> stop *

No ER groups found, but some coordinated threads may have been excluded.

GGSCI (nzdt02.oracle.com as ogg_admin@T1C1) 6> delete replicat rs1c1

ERROR: REPLICAT RS1C1 does not exist.

GGSCI (nzdt02.oracle.com as ogg_admin@T1C1) 7> add replicat rs1c1, exttrail
./dirdat/ta

REPLICAT added.

GGSCI (nzdt02.oracle.com as ogg admin@T1C1) 8>

```

## Synchronize In-Cloud Database

The in-cloud database synchronization begins after at the SCN of the clone database.

```

Oracle GoldenGate Command Interpreter for Oracle
Version 12.2.0.1.1 OGGCORE 12.2.0.1.0 PLATFORMS 151211.1401 FBO
Linux, x64, 64bit (optimized), Oracle 12c on Dec 12 2015 02:56:48
Operating system character set identified as US-ASCII.

Copyright (C) 1995, 2015, Oracle and/or its affiliates. All rights reserved.

GGSCI (nzdt02.oracle.com) 1>
GGSCI (nzdt02.oracle.com) 2> STOP REPLICAT RS1C1

```

```

REPLICAT RS1C1 is already stopped.

GGSCI (nzdt02.oracle.com) 3> START REPLICAT RS1C1, AFTERCSN 62502634

Sending START request to MANAGER ...
REPLICAT RS1C1 starting

GGSCI (nzdt02.oracle.com) 4>
Oracle GoldenGate Command Interpreter for Oracle
Version 12.2.0.1.1 OGGCORE 12.2.0.1.0 PLATFORMS 151211.1401 FBO
Linux, x64, 64bit (optimized), Oracle 12c on Dec 12 2015 02:56:48
Operating system character set identified as US-ASCII.

Copyright (C) 1995, 2015, Oracle and/or its affiliates. All rights reserved.

GGSCI (nzdt02.oracle.com) 1>
GGSCI (nzdt02.oracle.com) 2> info all

```

| Program  | Status  | Group | Lag at Chkpt | Time Since Chkpt |
|----------|---------|-------|--------------|------------------|
| MANAGER  | RUNNING |       |              |                  |
| REPLICAT | RUNNING | RS1C1 | 00:00:00     | 00:01:54         |

```

GGSCI (nzdt02.oracle.com) 3>

```

## Stop Applications

To ensure data consistency, the applications stopped for momentarily to allow complete data synchronization.

```

declare
  cursor c1 is select s.username, s.sid, s.serial#
                from v$session s
                join v$sqlarea a on a.hash value = s.sql hash value and
a.sql_text like '%call_trans_load%' and s.status='ACTIVE' and
username='OSM_REPAPI';
  vsid number;
  vserial# number;
  vusername varchar2(32);
begin
  open c1;
  loop
    fetch c1 into vusername, vsid, vserial#;
    exit when c1%notfound;
    execute immediate 'alter system disconnect session
'||''''||vsid||','||vserial#||''''||' immediate';
  end loop;
  close c1;

```

```

exception
    when others then
        null;
end;
/

```

## Enable Restricted Sessions

To further ensure data consistency, the source database to remain on restricted mode so that no business users connect to it incidentally.

```

$ORACLE_HOME/bin/sqlplus /nolog <<EOF
connect / as sysdba
@/home/oracle/admin/kill_call_trans_load.sql
alter system enable restricted session
/
EOF

```

## Verify On-Premise Database and In-Cloud Database

This step uses the row count technique to ensure tables on the source and the tables on the target have the same number of rows. A more sophisticated technique is to deploy Oracle GoldenGate 12c Veridata to conduct data verifications.

```

create public database link T1C1.oracle.com
connect to osm repapi identified by oracle
using 'T1C1'
/

```

```

select (select count(*) from osm_repapi.customers) - (select count(*) from
osm_repapi.customers@T1C1.oracle.com) RowsGap from dual
/

```

## Divert Applications

The analytical applications diverted to the new EDW target. This is application running on the middle tier – application Server.

```

[oracle@nzdt02 admin]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sat Mar 24 13:39:30 2018

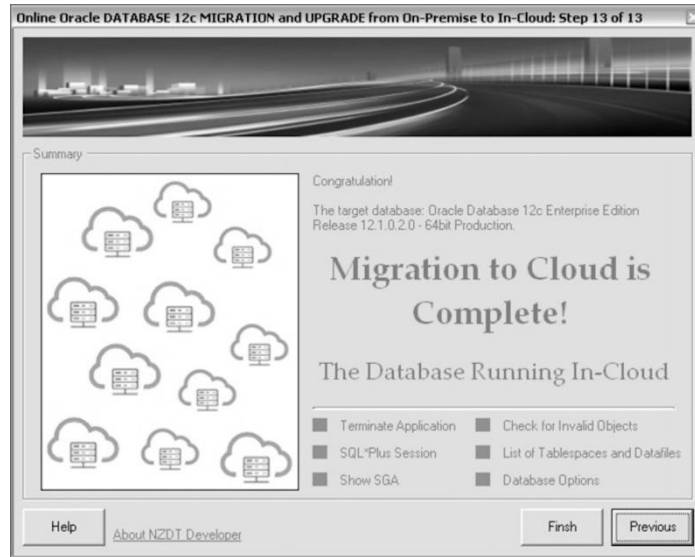
Copyright (c) 1982, 2014, Oracle. All rights reserved.

T1C1>conn osm_repapi/oracle
Connected.
T1C1>exec call_trans_load(10000, 1);

```

## Step 13: Confirmation

The online migration and upgrade of an Enterprise Data Warehouse (EDW) to Oracle cloud IaaS is hard to achieve without Oracle GoldenGate 12c working in the background, online preserving the transaction though sophisticated change data capture technology.



## Database Version

Login AS SYSDBA to the Enterprise Data Warehouse (EDW) database (Target), verify the database is running on Oracle Database 12c.

```
SQL*Plus: Release 12.1.0.2.0 Production on Fri Mar 23 15:09:07 2018

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Last Successful login time: Fri Mar 23 2018 15:02:43 +03:00

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

T1C1>
```

Congratulations, the last step shows the migration is completed successfully, the upgrade is completed successfully. The brilliant Oracle Recovery Manager (RMAN)

for slicing the database into fine-grain backup set, making the move of the database from on-premise to Oracle cloud IaaS guaranteed.

*This is the new way for the shift & lift to Oracle cloud IaaS; this is my way to the journey to Oracle Cloud IaaS!*

## Summary

Shifting the workload from on-premise to Oracle cloud IaaS is a migration of on-premise database to Oracle Cloud Infrastructure as a Service (IaaS). This appendix has demonstrated not only the migration but also the upgrade on-premise Oracle Database 11g to in-cloud Oracle Database 12c.

The workload simulator generates enough transactions workload for the duration of the migration to Oracle cloud IaaS. Select a workload reasonable for the available system resources.

Installation of Oracle GoldenGate 12c executed on the downstream and target database. No installation required on the source database. Installations are done using silent mode, which install and start the manager process on the port specified in the response file.

Configure the redo-transport to ship the transactions logs from the source database to the downstream running Oracle GoldenGate 12c. Upon arrival to the downstream system, Oracle GoldenGate 12c database mining engine generates the Logical Change Records (LCRs), for which change data capture is performed.

The integrated extract (IE) deployed by registering the extract with the database, and then creating the primary extract, extract trail, secondary extract and remote trail. Before proceeding further, verifying the extract is healthy and local and remote trails are accumulating.

Backup the database and transport the backup sets to Oracle Cloud IaaS using sFTP, the backup is verified ensure database files and relevant archive log files are available.

The database clone is executed using Oracle Recovery Manager (RMAN). RMAN uses the latest copy of online backup to clone the database using RMAN DUPLICATE command. The duplicate system change number is identified for use with the apply process – Replicat.

The database upgrade is performed against the cloned database. Using the DBUA on silent mode to upgrade the database requires running pre-upgrade and post-upgrade scripts.

Create the apply process – Replicat, to synchronize the target database with trail files accumulated for the duration of the project. Use the system change number of the database clone for the Replicat as the starting SCN. Verify the database upgrade by running data dictionary queries to display Oracle Database 12c version, etc.

# Automatic Conflict Detection and Resolution for Active-Active Database Availability Architecture



## Overview

With the announcement of Oracle GoldenGate 12c version 12.3, the deployment of active-active database for high-availability has been greatly automated. Design, build and deploy n-way Active-Active database architecture, where data streaming happens on all directions, empowers the automated build-in automated conflict detection and resolution and optimized data (n-1)-way re-streaming to achieved symmetrically distributed database applications.

Data conflict occurs when a replicated row update nearly at the same time, or following network restore after an extended time of failure. The resolution is implemented in the database and not in Oracle GoldenGate. The conflict enabled using Oracle database API calls, which create an invisible column to implement and enforce the resolution method.

The automated conflict detection and resolution (CDR) is certified on:

- Oracle GoldenGate 12c (12.3)
- Oracle Database 12c (12.2)

The predominant advantage of n-way Active-Active database architecture it delivers equal high availability for the database instance and database files. In the event of a site failure within the configuration, the site's applications users are diverted to a running database instance.

A major advantage of 2-way or higher Active-Active database architecture is rolling upgrade and maintenance. During such destructive operations such as database upgrade, the apply process where the maintenance is undergoing is suspended until the operation completes. After operation completes, the accumulated trail files are applied to the database and the n-way replication re-enabled. Moving to the next system, diverting the system's applications users and the same sequence of steps is applied to execute the database upgrade on this system.

# Architecture

Though, Oracle GoldenGate 12c is capable for supporting n-way bi-directional replication, this Appendix covers 2-way data replication implementing active-active database availability architecture. Figure 1 illustrates 2-way replication implemented using Oracle GoldenGate 12c integrated extract (IE) technology deployed locally.

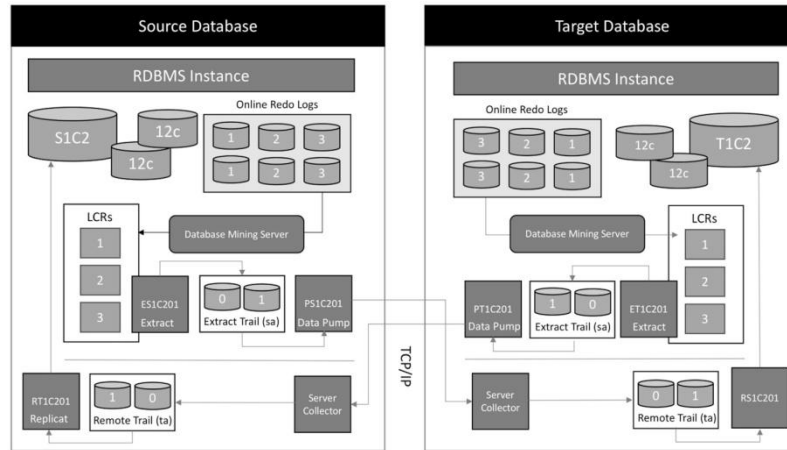


Figure 1: 2-Way Bi-Directional Data Replication

When Oracle GoldenGate 12c integrated extract (IE) locally deployed, where each database mining server is run by the database instance where Oracle GoldenGate 12c instance is running. The data pump process send data to the target database, and then send the same trail to

Expanding the

## Conflict Detection and Resolutions

T

Refer to Appendix E for installation and Oracle GoldenGate 12c pre-requisite for database preparation.



# Configure and Execute Initial Data Load

Using the direct initial-load method uses an initial load extract for reading the source database tables, across TCP/IP sending data directly to the target system initial-load Replicat. The special run Replicat process applies the data to the target database using the database native RDBMS SQL calls.

## Create the Target Checkpoint Table

The checkpoint database table is the default for the Replicat. Before attempting to define the Replicat, the checkpoint table should be created. The steps are show below:

1. Create or modify the GLOBALS parameter file by adding the parameter CHECKPOINT\*TABLE followed by the database schema and table name.
2. Login to the Database using GGSCI
3. Create the checkpoint table by run the GGSCI command ADD CHECKPINTTABLE

```
[oracle@linux5acdr gghome_1]$ cd $OGG_HOME
[oracle@linux5acdr gghome_1]$ vi GLOBALS
```

```
checkpointtable ogg_admin.chkpttab
```

Save and verify the changes.

```
[oracle@linux5acdr gghome_1]$ cat GLOBALS
checkpointtable ogg_admin.chkpttab

[oracle@linux5acdr gghome_1]$
```

Create the checkpoint table.

```
[oracle@linux5acdr gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE 12.3.0.1.0 PLATFORMS 170721.0154 FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.
```

```

GGSCI (linux5acdr.oracle.com) 1> dblogin userid ogg_admin@T1C2, password
oracle
Successfully logged into database.

GGSCI (linux5acdr.oracle.com as ogg_admin@T1C2) 2> add checkpointtable

No checkpoint table specified. Using GLOBALS specification
ogg_admin.chkpttab...

Successfully created checkpoint table ogg_admin.chkpttab.

GGSCI (linux5acdr.oracle.com as ogg_admin@T1C2) 5> shell clear

```

## Configure and Define the Initial-Load Extract

The initial load extract read the database table, instructed by the parameter *SOURCEISTABLE*. In a Direct load method, the initial load Replicat is started by the initial load Extract. After the initial load is completed, the Replicat goes into stop mode automatically.

The first step, prepare the parameter file for (eload), the second step defines the initial load extract.

```

GGSCI (linux4.oracle.com) 1> edit params eload

```

```

EXTRACT eload
USERID ogg_admin@S1C2, PASSWORD oracle
RMTHOST 192.168.1.50, MGRPORT 7850
RMTTASK REPLICAT, GROUP rload
TABLE osm_repapi.customers;
TABLE osm_repapi.policies;
TABLE osm_repapi.claims;

```

```

GGSCI (linux4.oracle.com) 3> add extract eload SOURCEISTABLE
EXTRACT added.

```

```

GGSCI (linux4.oracle.com) 4>

```

## Configure and Define the Initial-Load Replicat

Pare the parameter file for the initial Replicat (rload). Notice, the parameter *HANDLECOLLISIONS* is used as precaution to avoid aborting the initial load in the event of insert error message (ORA-00001). The Replicat will go into stop mode after completion; this is designated by the parameter *SPECIALRUN* when defining the Replicat.

```

[oracle@linux5acdr ~]$ cd $OGG_HOME

```

```
[oracle@linux5acdr gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE_12.3.0.1.0_PLATFORMS_170721.0154_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (linux5acdr.oracle.com) 1>
```

```
GGSCI (linux5acdr.oracle.com) 1> edit params rload
```

```
REPLICAT rload
USERID ogg_admin@T1C2, password oracle
ASSUMETARGETDEFS
HANDLECOLLISIONS
DISCARDFILE ./dirrpt/initload01.dsc, PURGE
MAP osm_repapi.*, TARGET osm_repapi.*;
```

```
GGSCI (linux5acdr.oracle.com) 2> ADD REPLICAT rload, SPECIALRUN
REPLICAT added.
```

```
GGSCI (linux5acdr.oracle.com) 3>
```

## Starting the Initial-Load

It's now time to start the initial load process. From the Source system GGSCI, start the initial load extract (rload). To check for error, used the GGSCI command VIEW REPORT RLOAD. Before starting the initial-load Replicat, reconfigure the manager process to allow access rule for all program coming from the source Oracle GoldenGate instance.

```
GGSCI (linux4.oracle.com) 9> edit params mgr
```

```
PORT 7840
ACCESSRULE, PROG *, IPADDR 192.168.1.40, ALLOW
```

```
GGSCI (linux4.oracle.com) 13> stop mgr!
```

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

```
GGSCI (linux4.oracle.com) 14> start mgr
Manager started.
```

```
GGSCI (linux4.oracle.com) 15>
```

```
GGSCI (linux5acdr.oracle.com) 3> edit params mgr
```

```
PORT 7850  
ACCESSRULE, PROG *, IPADDR 192.168.1.40, ALLOW
```

```
GGSCI (linux5acdr.oracle.com) 5> stop mgr!  
  
Sending STOP request to MANAGER ...  
Request processed.  
Manager stopped.
```

```
GGSCI (linux5acdr.oracle.com) 6> start mgr  
Manager started.
```

```
GGSCI (linux5acdr.oracle.com) 7>
```

```
GGSCI (linux4.oracle.com as ogg_admin@S1C2) 15> start extract eload
```

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

```
GGSCI (linux4.oracle.com as ogg_admin@S1C2) 16>
```

## Verify the Initial Load from the Source

Verify the initial load done either using SQL\*Plus rows count script, or using the GGSCI command – VIEW REPORT <initial load extract name>. Run the command on the source and target for comparing the INSERT is matching.

```
GGSCI (linux4.oracle.com) 20> view report eload
```

```
2018-01-06 15:37:38 INFO OGG-01017 Wildcard resolution set to IMMEDIATE  
because SOURCEISTABLE is used.
```

```
*****
```

```
Oracle GoldenGate Capture for Oracle  
Version 12.3.0.1.0 OGGCORE_12.3.0.1.0_PLATFORMS_170721.0154_FBO  
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:33:48
```

```
Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.
```

```
Starting at 2018-01-06 15:37:38
```

```
*****
```

```
Operating System Version:  
Linux
```

Version #2 SMP Thu Feb 23 20:03:53 PST 2017, Release 4.1.12-  
61.1.28.el6uek.x86\_64  
Node: linux4.oracle.com  
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: 16257

Description:

```
*****
**                Running with the following parameters                **
*****
```

2018-01-06 15:37:38 INFO OGG-03059 Operating system character set identified as UTF-8.

2018-01-06 15:37:38 INFO OGG-02695 ANSI SQL parameter syntax is used for parameter parsing.

2018-01-06 15:37:38 INFO OGG-01360 EXTRACT is running in Initial Load mode.

2018-01-06 15:37:38 INFO OGG-01889 Flush size (max message size) is set to 27,985.  
EXTRACT eload  
USERID ogg\_admin@S1C2, PASSWORD \*\*\*

2018-01-06 15:37:38 INFO OGG-03522 Setting session time zone to source database time zone 'GMT'.  
RMTHOST 192.168.1.50, MGRPORT 7850  
RMTTASK REPLICAT, GROUP rload  
TABLE osm\_repapi.customers;  
TABLE osm\_repapi.policies;  
TABLE osm\_repapi.claims;

2018-01-06 15:37:38 INFO OGG-01851 filecaching started: thread ID: 140406234986240.

2018-01-06 15:37:38 INFO OGG-01815 Virtual Memory Facilities for: COM  
anon alloc: mmap(MAP\_ANON) anon free: munmap  
file alloc: mmap(MAP\_SHARED) file free: munmap  
target directories:  
/u01/app/oracle/product/12.3.0/gghome\_1/dirtmp.

Database Version:  
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production  
PL/SQL Release 12.2.0.1.0 - Production  
CORE 12.2.0.1.0 Production  
TNS for Linux: Version 12.2.0.1.0 - Production  
NLSRTL Version 12.2.0.1.0 - Production

Database Language and Character Set:  
NLS\_LANGUAGE = "AMERICAN"  
NLS\_TERRITORY = "AMERICA"

```

NLS_CHARACTERSET = "AL32UTF8"

2018-01-06 15:37:38 INFO      OGG-06509 Using the following key columns for
source table OSM_REPAPI.CUSTOMERS: CUST_NO.

2018-01-06 15:37:38 INFO      OGG-06509 Using the following key columns for
source table OSM_REPAPI.POLICIES: POL_NO.

2018-01-06 15:37:38 INFO      OGG-06509 Using the following key columns for
source table OSM_REPAPI.CLAIMS: CUST_NO, POL_NO
.

2018-01-06 15:37:44 INFO      OGG-02911 Processing table
OSM_REPAPI.CUSTOMERS.

GGSCI (linux4.oracle.com) 22>

```

## Verify the Initial Load from the Target System

Run the GGSCI VIEW REPORT on the target to obtain the statistics result for comparison.

```

GGSCI (linux5acdr.oracle.com) 13> view report rload

```

```

*****
                Oracle GoldenGate Delivery for Oracle
                Version 12.3.0.1.0 OGGCORE_12.3.0.1.0_PLATFORMS_170721.0154_FBO
                Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:36:12

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

```

```

                Starting at 2018-01-06 15:42:47
*****

```

```

Operating System Version:
Linux
Version #2 SMP Thu Feb 23 20:03:53 PST 2017, Release 4.1.12-
61.1.28.el6uek.x86_64
Node: linux5acdr.oracle.com
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: 16192

Description:

```

*****
**                                Running with the following parameters                                **

```

```

*****

2018-01-06 15:42:52 INFO      OGG-03059  Operating system character set
identified as UTF-8.

2018-01-06 15:42:52 INFO      OGG-02695  ANSI SQL parameter syntax is used
for parameter parsing.

2018-01-06 15:42:52 INFO      OGG-01360  REPLICAT is running in Remote Task
mode.
REPLICAT rload
USERID ogg_admin@T1C2, password ***
ASSUMETARGETDEFS
HANDLECOLLISIONS
DISCARDFILE ./dirrpt/initload01.dsc, PURGE
MAP osm_repapi.*, TARGET osm_repapi.*;

2018-01-06 15:42:52 INFO      OGG-06451  Triggers will be suppressed by
default.

2018-01-06 15:42:52 INFO      OGG-01815  Virtual Memory Facilities for: COM
anon alloc: mmap(MAP_ANON) anon free: munmap
file alloc: mmap(MAP_SHARED) file free: munmap
target directories:
/u01/app/oracle/product/12.3.0/gghome_1/dirtmp.

Database Version:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
PL/SQL Release 12.2.0.1.0 - Production
CORE 12.2.0.1.0 Production
TNS for Linux: Version 12.2.0.1.0 - Production
NLSRTL Version 12.2.0.1.0 - Production

Database Language and Character Set:
NLS_LANGUAGE = "AMERICAN"
NLS_TERRITORY = "AMERICA"
NLS_CHARACTERSET = "AL32UTF8"

2018-01-06 15:42:52 INFO      OGG-06604  Database T1C2.ORACLE.COM CPU info:
CPU Count 1, CPU Core Count 1, CPU Socket Count 1.

*****
**                               Run Time Messages                               **
*****

2018-01-06 15:42:52 INFO      OGG-03522  Setting session time zone to source
database time zone 'GMT'.

2018-01-06 15:42:52 WARNING OGG-02760  ASSUMETARGETDEFS is ignored because
trail file contains table definitions.

2018-01-06 15:42:52 INFO      OGG-06506  Wildcard MAP resolved (entry
osm_repapi.): MAP "OSM_REPAPI"."CUSTOMERS", TARGET osm_repapi."CUSTOMERS".

2018-01-06 15:42:53 INFO      OGG-02756  The definition for table
OSM_REPAPI.CUSTOMERS is obtained from the trail file.

```

```
2018-01-06 15:42:53 INFO OGG-06511 Using following columns in default
map by name: CUST_NO, CUST_NAME, CUST_EMAIL, CUST_MOBILE, CUST_ADDRESS,
CUST_
TS, CUST_AGE, REG_ID, EDUL_ID, PROF_ID.
```

```
2018-01-06 15:42:53 INFO OGG-06510 Using the following key columns for
target table OSM_REPAPI.CUSTOMERS: CUST_NO.
```

```
2018-01-06 15:43:17 INFO OGG-06506 Wildcard MAP resolved (entry
osm_repapi.*): MAP "OSM_REPAPI"."POLICIES", TARGET osm_repapi."POLICIES".
```

```
2018-01-06 15:43:17 INFO OGG-02756 The definition for table
OSM_REPAPI.POLICIES is obtained from the trail file.
```

```
2018-01-06 15:43:17 INFO OGG-06511 Using following columns in default
map by name: POL_NO, POL_FROM, POL_TO, POL_VALUE, POL_SUB_TOTAL, PT_CODE,
DT
CODE, POL_TOTAL, CUST_NO.
```

```
2018-01-06 15:43:17 INFO OGG-06510 Using the following key columns for
target table OSM_REPAPI.POLICIES: POL_NO.
```

```
GGSCI (linux5acdr.oracle.com) 14>
```

Alternatively, use SQL\*Plus to compare the rows counts between the two databases.

## Source Database

```
SQL> conn osm_repapi/oracle
Connected.
SQL> select count(*) from customers;

COUNT (*)
-----
240972

SQL> select count(*) from policies;

COUNT (*)
-----
241034

SQL> select count(*) from claims;

COUNT (*)
-----
4052

SQL>
```



## Target Database

```
SQL> conn osm_repapi/oracle
Connected.
SQL> select count(*) from customers;

COUNT (*)
-----
240972

SQL> select count(*) from policies;

COUNT (*)
-----
241034

SQL> select count(*) from claims;

COUNT (*)
-----
4052

SQL>
```

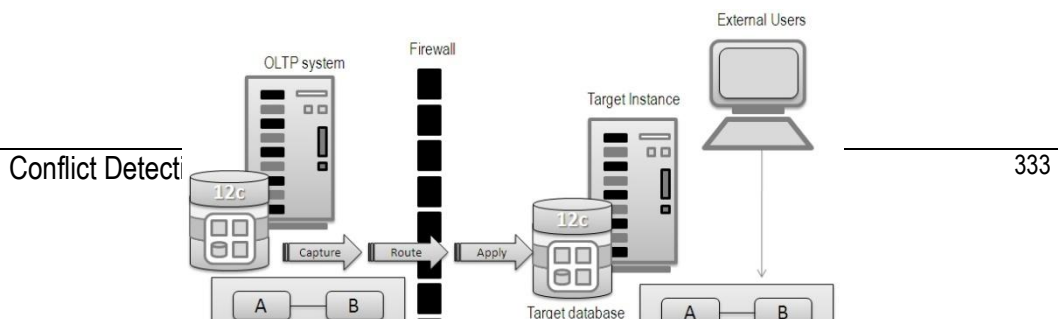
## Configure and Deploy Unidirectional Data Replication

This workshop illustrates two Oracle GoldenGate 12c focus areas (Implementations).

- Database optimization through query offloading
- Secure Web-Services Platform

The purpose of Database optimization through query offloading is to divert query-intensive and BI dashboard to the query offloading instance. And, when taking advantage of the database query re-write and materialized views, reports that takes hours to complete, may finish on few minutes or seconds.

Secure Web-Services needs to query and manipulate the database, instead of exposing the complete database to external users, only the related tables/views needed by the Web Services are exposed. Oracle GoldenGate 12c move data on real-time from the production database to a target Oracle GoldenGate 12c instance. Applying the different types of Oracle Database 12c security options and features, leads to highly secure environment for a database accessed by external users. The following figure illustrate the components of Secure Web Service Platform.



## Enable Table-Level Supplemental Logging

Enable table-level supplemental logging using GGSCI command ADD TRANDATA. By default, it only capture the before image of the key. If you need the before image of all columns, use the COLS attribute followed by the column names.

```
[oracle@linux4acdr gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE_12.3.0.1.0_PLATFORMS_170721.0154_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (linux4.oracle.com) 1> dblogin userid ogg_admin@S1C2, password oracle
Successfully logged into database.

GGSCI (linux4.oracle.com as ogg_admin@S1C2) 2> add trandata osm_repapi.*
```

```
2018-01-06 16:45:03 INFO      OGG-15132  Logging of supplemental redo data
enabled for table OSM_REPAPI.CLAIMS.

2018-01-06 16:45:03 INFO      OGG-15133  TRANDATA for scheduling columns has
been added on table OSM_REPAPI.CLAIMS.

2018-01-06 16:45:03 INFO      OGG-15135  TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.CLAIMS.

2018-01-06 16:45:03 INFO      OGG-15132  Logging of supplemental redo data
enabled for table OSM_REPAPI.CLAIM_STATUS.

2018-01-06 16:45:03 INFO      OGG-15133  TRANDATA for scheduling columns has
been added on table OSM_REPAPI.CLAIM_STATUS.

2018-01-06 16:45:03 INFO      OGG-15135  TRANDATA for instantiation CSN has
been added on table OSM_REPAPI.CLAIM_STATUS.
```

2018-01-06 16:45:03 INFO OGG-15132 Logging of supplemental redo data enabled for table OSM\_REPAPI.CUSTOMERS.

2018-01-06 16:45:03 INFO OGG-15133 TRANDATA for scheduling columns has been added on table OSM\_REPAPI.CUSTOMERS.

2018-01-06 16:45:03 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.CUSTOMERS.

2018-01-06 16:45:03 INFO OGG-15132 Logging of supplemental redo data enabled for table OSM\_REPAPI.DISCOUNT\_TYPES.

2018-01-06 16:45:03 INFO OGG-15133 TRANDATA for scheduling columns has been added on table OSM\_REPAPI.DISCOUNT\_TYPES.

2018-01-06 16:45:03 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.DISCOUNT\_TYPES.

2018-01-06 16:45:03 INFO OGG-15132 Logging of supplemental redo data enabled for table OSM\_REPAPI.EDU\_LEVELS.

2018-01-06 16:45:03 INFO OGG-15133 TRANDATA for scheduling columns has been added on table OSM\_REPAPI.EDU\_LEVELS.

2018-01-06 16:45:03 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.EDU\_LEVELS.

2018-01-06 16:45:04 INFO OGG-15132 Logging of supplemental redo data enabled for table OSM\_REPAPI.OSM\_DOCUMENTS.

2018-01-06 16:45:04 INFO OGG-15133 TRANDATA for scheduling columns has been added on table OSM\_REPAPI.OSM\_DOCUMENTS.

2018-01-06 16:45:04 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.OSM\_DOCUMENTS.

2018-01-06 16:45:04 INFO OGG-15132 Logging of supplemental redo data enabled for table OSM\_REPAPI.POLICIES.

2018-01-06 16:45:04 INFO OGG-15133 TRANDATA for scheduling columns has been added on table OSM\_REPAPI.POLICIES.

2018-01-06 16:45:04 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.POLICIES.

2018-01-06 16:45:04 INFO OGG-15132 Logging of supplemental redo data enabled for table OSM\_REPAPI.POLICY\_TYPES.

2018-01-06 16:45:04 INFO OGG-15133 TRANDATA for scheduling columns has been added on table OSM\_REPAPI.POLICY\_TYPES.

2018-01-06 16:45:04 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.POLICY\_TYPES.

2018-01-06 16:45:04 INFO OGG-15132 Logging of supplemental redo data enabled for table OSM\_REPAPI.PROFESSIONS.

2018-01-06 16:45:04 INFO OGG-15133 TRANDATA for scheduling columns has been added on table OSM\_REPAPI.PROFESSIONS.

2018-01-06 16:45:04 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.PROFESSIONS.

2018-01-06 16:45:04 INFO OGG-15132 Logging of supplemental redo data enabled for table OSM\_REPAPI.RANDOM\_DOMAIN.

2018-01-06 16:45:04 INFO OGG-15133 TRANDATA for scheduling columns has been added on table OSM\_REPAPI.RANDOM\_DOMAIN.

2018-01-06 16:45:04 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.RANDOM\_DOMAIN.

2018-01-06 16:45:04 INFO OGG-15132 Logging of supplemental redo data enabled for table OSM\_REPAPI.RANDOM\_FIRST.

2018-01-06 16:45:04 INFO OGG-15133 TRANDATA for scheduling columns has been added on table OSM\_REPAPI.RANDOM\_FIRST.

2018-01-06 16:45:04 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.RANDOM\_FIRST.

2018-01-06 16:45:04 INFO OGG-15132 Logging of supplemental redo data enabled for table OSM\_REPAPI.RANDOM\_LAST.

2018-01-06 16:45:04 INFO OGG-15133 TRANDATA for scheduling columns has been added on table OSM\_REPAPI.RANDOM\_LAST.

2018-01-06 16:45:04 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.RANDOM\_LAST.

2018-01-06 16:45:04 INFO OGG-15132 Logging of supplemental redo data enabled for table OSM\_REPAPI.RANDOM\_MIDDLE.

2018-01-06 16:45:04 INFO OGG-15133 TRANDATA for scheduling columns has been added on table OSM\_REPAPI.RANDOM\_MIDDLE.

2018-01-06 16:45:04 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.RANDOM\_MIDDLE.

2018-01-06 16:45:04 INFO OGG-15132 Logging of supplemental redo data enabled for table OSM\_REPAPI.REGIONS.

2018-01-06 16:45:04 INFO OGG-15133 TRANDATA for scheduling columns has been added on table OSM\_REPAPI.REGIONS.

2018-01-06 16:45:04 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.REGIONS.

2018-01-06 16:45:04 INFO OGG-10471 \*\*\*\*\* Oracle Goldengate support information on table OSM\_REPAPI.CLAIMS \*\*\*\*\*  
 Oracle Goldengate support native capture on table OSM\_REPAPI.CLAIMS.  
 Oracle Goldengate marked following column as key columns on table OSM\_REPAPI.CLAIMS: CUST\_NO, POL\_NO.

2018-01-06 16:45:04 INFO OGG-10471 \*\*\*\*\* Oracle Goldengate support information on table OSM\_REPAPI.CLAIM\_STATUS \*\*\*\*\*  
 Oracle Goldengate support native capture on table OSM\_REPAPI.CLAIM\_STATUS.

Oracle Goldengate marked following column as key columns on table  
OSM\_REPAPI.CLAIM\_STATUS: CLM\_STATUS.

2018-01-06 16:45:04 INFO OGG-10471 \*\*\*\*\* Oracle Goldengate support  
information on table OSM\_REPAPI.CUSTOMERS \*\*\*\*\*  
Oracle Goldengate support native capture on table OSM\_REPAPI.CUSTOMERS.  
Oracle Goldengate marked following column as key columns on table  
OSM\_REPAPI.CUSTOMERS: CUST\_NO.

2018-01-06 16:45:04 INFO OGG-10471 \*\*\*\*\* Oracle Goldengate support  
information on table OSM\_REPAPI.DISCOUNT\_TYPES \*\*\*\*\*  
Oracle Goldengate support native capture on table OSM\_REPAPI.DISCOUNT\_TYPES.  
Oracle Goldengate marked following column as key columns on table  
OSM\_REPAPI.DISCOUNT\_TYPES: DT\_CODE.

2018-01-06 16:45:04 INFO OGG-10471 \*\*\*\*\* Oracle Goldengate support  
information on table OSM\_REPAPI.EDU\_LEVELS \*\*\*\*\*  
Oracle Goldengate support native capture on table OSM\_REPAPI.EDU\_LEVELS.  
Oracle Goldengate marked following column as key columns on table  
OSM\_REPAPI.EDU\_LEVELS: EDUL\_ID.

2018-01-06 16:45:04 INFO OGG-10471 \*\*\*\*\* Oracle Goldengate support  
information on table OSM\_REPAPI.OSM\_DOCUMENTS \*\*\*\*\*  
Oracle Goldengate support native capture on table OSM\_REPAPI.OSM\_DOCUMENTS.  
Oracle Goldengate marked following column as key columns on table  
OSM\_REPAPI.OSM\_DOCUMENTS: NAME.

2018-01-06 16:45:04 INFO OGG-10471 \*\*\*\*\* Oracle Goldengate support  
information on table OSM\_REPAPI.POLICIES \*\*\*\*\*  
Oracle Goldengate support native capture on table OSM\_REPAPI.POLICIES.  
Oracle Goldengate marked following column as key columns on table  
OSM\_REPAPI.POLICIES: POL\_NO.

2018-01-06 16:45:04 INFO OGG-10471 \*\*\*\*\* Oracle Goldengate support  
information on table OSM\_REPAPI.POLICY\_TYPES \*\*\*\*\*  
Oracle Goldengate support native capture on table OSM\_REPAPI.POLICY\_TYPES.  
Oracle Goldengate marked following column as key columns on table  
OSM\_REPAPI.POLICY\_TYPES: PT\_CODE.

2018-01-06 16:45:04 INFO OGG-10471 \*\*\*\*\* Oracle Goldengate support  
information on table OSM\_REPAPI.PROFESSIONS \*\*\*\*\*  
Oracle Goldengate support native capture on table OSM\_REPAPI.PROFESSIONS.  
Oracle Goldengate marked following column as key columns on table  
OSM\_REPAPI.PROFESSIONS: PROF\_ID.

2018-01-06 16:45:04 INFO OGG-10471 \*\*\*\*\* Oracle Goldengate support  
information on table OSM\_REPAPI.RANDOM\_DOMAIN \*\*\*\*\*  
Oracle Goldengate support native capture on table OSM\_REPAPI.RANDOM\_DOMAIN.  
Oracle Goldengate marked following column as key columns on table  
OSM\_REPAPI.RANDOM\_DOMAIN: RD\_NAME.

2018-01-06 16:45:04 INFO OGG-10471 \*\*\*\*\* Oracle Goldengate support  
information on table OSM\_REPAPI.RANDOM\_FIRST \*\*\*\*\*  
Oracle Goldengate support native capture on table OSM\_REPAPI.RANDOM\_FIRST.  
Oracle Goldengate marked following column as key columns on table  
OSM\_REPAPI.RANDOM\_FIRST: RF\_NAME.

2018-01-06 16:45:04 INFO OGG-10471 \*\*\*\*\* Oracle Goldengate support  
information on table OSM\_REPAPI.RANDOM\_LAST \*\*\*\*\*

```
Oracle Goldengate support native capture on table OSM_REPAPI.RANDOM_LAST.  
Oracle Goldengate marked following column as key columns on table  
OSM_REPAPI.RANDOM_LAST: RL_NAME.
```

```
2018-01-06 16:45:04 INFO OGG-10471 ***** Oracle Goldengate support  
information on table OSM_REPAPI.RANDOM_MIDDLE *****  
Oracle Goldengate support native capture on table OSM_REPAPI.RANDOM_MIDDLE.  
Oracle Goldengate marked following column as key columns on table  
OSM_REPAPI.RANDOM_MIDDLE: RM_NAME.
```

```
2018-01-06 16:45:04 INFO OGG-10471 ***** Oracle Goldengate support  
information on table OSM_REPAPI.REGIONS *****  
Oracle Goldengate support native capture on table OSM_REPAPI.REGIONS.  
Oracle Goldengate marked following column as key columns on table  
OSM_REPAPI.REGIONS: REG_ID.
```

```
GGSCI (linux4.oracle.com as ogg_admin@S1C2) 3>
```

## Configure and Define the Primary Group

Defining the primary group composed of three steps: Prepare the parameter file, create the extract and link the primary extract with the local trail - sa.

### Prepare the Parameter File

The basic parameter files are listed. The first line is the Extract name used by the ADD EXTRACT statement.

```
GGSCI (linux4.oracle.com as ogg_admin@S1C2) 3> edit params es1c201
```

```
EXTRACT ES1C201  
SETENV (ORACLE_SID = S1C2)  
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.2.0/dbhome_1")  
SETENV (NLS_LANG = AMERICAN AMERICA.AL32UTF8)  
USERID ogg_admin@S1C2, PASSWORD oracle  
EXTRAIL ./dirdat/sa  
TABLE osm_repapi.*;
```

### Define the Primary Integrated Extract

The primary extract read the database transaction logs. The name of the Extract is ES1C201.

Register the primary integrated extract

```
GGSCI (linux4.oracle.com as ogg_admin@S1C2) 6> register extract ES1C201  
database
```

```
2018-01-06 16:52:14 INFO OGG-02003 Extract ES1C201 successfully
registered with database at SCN 2739158.
```

```
GGSCI (linux4.oracle.com as ogg_admin@S1C2) 7>
```

Define the integrated extract

```
GGSCI (linux4.oracle.com as ogg_admin@S1C2) 7> ADD EXTRACT ES1C201,
INTEGRATED TRANLOG, BEGIN now
EXTRACT (Integrated) added.
```

```
GGSCI (linux4.oracle.com as ogg_admin@S1C2) 8>
```

## Define the Local Trail

Link the local trail with primary extract - ES1C201.

```
GGSCI (linux4.oracle.com as ogg_admin@S1C2) 8> ADD EXTTRAIL ./dirdat/sa,
EXTRACT ES1C201, MEGABYTES 32
EXTTRAIL added.
```

```
GGSCI (linux4.oracle.com as ogg_admin@S1C2) 9>
```

## Configure and Define the Secondary Group

The secondary group (Data pump) PS1C201 routes data to the target system.

```
GGSCI (linux4.oracle.com as ogg_admin@S1C2) 9> edit params PS1C201
```

```
EXTRACT PS1C201
USERID ogg_admin@S1C2, PASSWORD oracle
NOPASSTHRU
RMTHOST 192.168.1.50, MGRPORT 7850
RMTTRAIL ./dirdat/ta
TABLE osm_repapi.*;
```

## Define the Secondary Extract

Oracle GoldenGate best practices uses a data pump to offload the primary extract from sending data to the target system. The following GGSCI command define the secondary capture (data pump) – PS1C201.

```
GGSCI (linux4.oracle.com as ogg_admin@S1C2) 10> ADD EXTRACT PS1C201,
EXTTRAILSOURCE ./dirdat/sa
EXTRACT added.
```

```
GGSCI (linux4.oracle.com as ogg_admin@S1C2) 11>
```

## Define the Remote Trail

Link the remote trail with primary extract - PS1C201.

```
GGSCI (linux4.oracle.com as ogg_admin@S1C2) 11> ADD RMTTRAIL ./dirdat/ta,  
EXTRACT ps1c201, MEGABYTES 32  
RMTTRAIL added.
```

```
GGSCI (linux4.oracle.com as ogg_admin@S1C2) 12>
```

## Prepare and Define the Delivery Group

The Replicat is part of the delivery group. The Replicat parameter file and definition follows. Note, the Replicat needs a checkpoint table which has been already created during the initial-load setup.

### Prepare the Parameter File

The parameter file for the Replicat – RS1C201. Notice, the Replicat invoke an SQL statement to fetch the target system current LOCALTIMESTAMP. The purpose of this column is the implementation of the built-in Conflict Detection and Resolution (CDR), or some other purpose such as calculating the Replicat lagging.

```
GGSCI (linux5acdr.oracle.com) 1> edit params RS1C2
```

```
REPLICAT RS1C2  
SETENV (ORACLE_SID = T1C2)  
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)  
USERID ogg_admin@T1C2, PASSWORD oracle  
ASSUMETARGETDEFS  
REPERERROR (DEFAULT, DISCARD)  
DISCARDFILE /u01/app/oracle/product/12.3.0/gghome_1/dirrpt/RT1C2.dsc, PURGE  
--  
MAP osm_repapi.customers, TARGET osm_repapi.customers;  
MAP osm_repapi.policies, TARGET osm_repapi.policies;  
MAP osm_repapi.claims, TARGET osm_repapi.claims;
```

## Define the Replicat

Use the add REPLICAT GGSCI command to define the Replicat.

```
GGSCI (linux5acdr.oracle.com) 2> ADD REPLICAT rs1c2, integrated, parallel,  
EXTTRAIL ./dirdat/ta
```



```

REPLICAT (Parallel) added.

GGSCI (linux5acdr.oracle.com) 3>

```

## Start the Extracts and Replicat Processes

Start the Primary and Secondary extracts (ES1C201 and PS1C201).

```

GGSCI (linux4.oracle.com as ogg_admin@S1C2) 29> start *

Sending START request to MANAGER ...
EXTRACT ES1C201 starting

Sending START request to MANAGER ...
EXTRACT PS1C201 starting

GGSCI (linux4.oracle.com as ogg_admin@S1C2) 30> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
EXTRACT      RUNNING     ES1C201    00:00:00      00:00:01
EXTRACT      RUNNING     PS1C201    00:00:00      00:00:08

GGSCI (linux4.oracle.com as ogg_admin@S1C2) 31>

```

Start the Replicat process (RS1C201).

```

GGSCI (linux5acdr.oracle.com) 5> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
REPLICAT     STOPPED     RS1C201    00:00:00      00:03:15

```

```

GGSCI (linux5acdr.oracle.com) 3> start *

Sending START request to MANAGER ...
REPLICAT RS1C2 starting

GGSCI (linux5acdr.oracle.com) 4> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
REPLICAT     RUNNING     RS1C2      00:00:00      00:05:00

```

## Run and Verify Data Movement from Source to Target

Run the application from the source system and generate statistics on source and target. The provided PL/SQL workload simulator uses different actual parameters to create several transaction levels. Using the default value generate a reasonable workload which insert 5000 rows with a transaction interval of 5 seconds. Invoke the workload by entering the name of the procedure – `call_trans_load` without parameters.

```
SQL> exec call_trans_load;
```

```
[oracle@linux4acdr gghome_1]$ . oraenv
ORACLE_SID = [S1C2] ?
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@linux4acdr gghome_1]$ sqlplus osm_repapi/oracle

SQL*Plus: Release 12.2.0.1.0 Production on Sat Jan 6 17:10:18 2018

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Last Successful login time: Sat Jan 06 2018 15:44:53 +03:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> exec call_trans_load;
```

Open a new terminal on the target system, verify rows count using SQL\*Plus.

```
[oracle@linux5acdr dirdat]$ sqlplus /nolog

SQL*Plus: Release 12.2.0.1.0 Production on Sat Jan 6 17:11:48 2018

Copyright (c) 1982, 2016, Oracle. All rights reserved.

SQL> conn osm_repapi/oracle
Connected.
SQL> select count(*) from customers;

  COUNT(*)
-----
    241142

SQL> /

  COUNT(*)
-----
    241152

SQL>
```

Display statistics from Oracle GoldenGate.

```
[oracle@linux5acdr dirdat]$ cd $OGG_HOME
[oracle@linux5acdr gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE 12.3.0.1.0 PLATFORMS 170721.0154 FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (linux5acdr.oracle.com) 1> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
REPLICAT     RUNNING     RS1C2      00:00:00      00:00:10

GGSCI (linux5acdr.oracle.com) 2> stats replicat RS1C2
```

Sending STATS request to REPLICAT RS1C2 ...

Parallel Replicat Statistics:

|                        |        |
|------------------------|--------|
| Total transactions     | 699.00 |
| Large transactions     | 0.00   |
| Total metadata changes | 0.00   |
| DDL operations         | 0.00   |
| Replicated procedures  | 0.00   |
| Event Actions          | 0.00   |
| Datatype functionality | 0.00   |
| Store Procedures       | 0.00   |
| Checkpoint errors      | 0.00   |
| Transactions retried   | 0.00   |

Start of Statistics at 2018-01-06 17:13:19.

Replicating from OSM\_REPAPI.CUSTOMERS to OSM\_REPAPI.CUSTOMERS:

```
*** Total statistics since 2018-01-06 17:10:33 ***
Total inserts          313.00
Total updates           0.00
Total deletes           0.00
Total discards          0.00
Total operations       313.00

*** Daily statistics since 2018-01-06 17:10:33 ***
Total inserts          313.00
Total updates           0.00
Total deletes           0.00
```

|                                                              |        |
|--------------------------------------------------------------|--------|
| Total discards                                               | 0.00   |
| Total operations                                             | 313.00 |
| *** Hourly statistics since 2018-01-06 17:10:33 ***          |        |
| Total inserts                                                | 313.00 |
| Total updates                                                | 0.00   |
| Total deletes                                                | 0.00   |
| Total discards                                               | 0.00   |
| Total operations                                             | 313.00 |
| *** Latest statistics since 2018-01-06 17:10:33 ***          |        |
| Total inserts                                                | 313.00 |
| Total updates                                                | 0.00   |
| Total deletes                                                | 0.00   |
| Total discards                                               | 0.00   |
| Total operations                                             | 313.00 |
| Replicating from OSM_REPAPI.POLICIES to OSM_REPAPI.POLICIES: |        |
| *** Total statistics since 2018-01-06 17:10:33 ***           |        |
| Total inserts                                                | 310.00 |
| Total updates                                                | 0.00   |
| Total deletes                                                | 0.00   |
| Total discards                                               | 0.00   |
| Total operations                                             | 310.00 |
| *** Daily statistics since 2018-01-06 17:10:33 ***           |        |
| Total inserts                                                | 310.00 |
| Total updates                                                | 0.00   |
| Total deletes                                                | 0.00   |
| Total discards                                               | 0.00   |
| Total operations                                             | 310.00 |
| *** Hourly statistics since 2018-01-06 17:10:33 ***          |        |
| Total inserts                                                | 310.00 |
| Total updates                                                | 0.00   |
| Total deletes                                                | 0.00   |
| Total discards                                               | 0.00   |
| Total operations                                             | 310.00 |
| *** Latest statistics since 2018-01-06 17:10:33 ***          |        |
| Total inserts                                                | 310.00 |
| Total updates                                                | 0.00   |
| Total deletes                                                | 0.00   |
| Total discards                                               | 0.00   |
| Total operations                                             | 310.00 |
| Replicating from OSM_REPAPI.CLAIMS to OSM_REPAPI.CLAIMS:     |        |
| *** Total statistics since 2018-01-06 17:10:33 ***           |        |
| Total inserts                                                | 37.00  |
| Total updates                                                | 0.00   |
| Total deletes                                                | 0.00   |
| Total discards                                               | 0.00   |
| Total operations                                             | 37.00  |
| *** Daily statistics since 2018-01-06 17:10:33 ***           |        |
| Total inserts                                                | 37.00  |
| Total updates                                                | 0.00   |

```

Total deletes                0.00
Total discards               0.00
Total operations             37.00

*** Hourly statistics since 2018-01-06 17:10:33 ***
Total inserts                37.00
Total updates                0.00
Total deletes                0.00
Total discards               0.00
Total operations             37.00

*** Latest statistics since 2018-01-06 17:10:33 ***
Total inserts                37.00
Total updates                0.00
Total deletes                0.00
Total discards               0.00
Total operations             37.00

End of Statistics.

```

```
GGSCI (linux5acdr.oracle.com) 3>
```

## Configure and Deploy Bi-Directional Data Replication

The

### Enable Table-Level Supplemental Logging

Enable table-level supplemental logging using GGSCI command ADD TRANDATA. By default, it only capture the before image of the key. If you need the before image of all columns, use the COLS attribute followed by the column names.

```

[oracle@linux5acdr gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE_12.3.0.1.0_PLATFORMS_170721.0154_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (linux5acdr.oracle.com) 1> dblogin userid ogg_admin@T1C2, password
oracle
Successfully logged into database.

GGSCI (linux5acdr.oracle.com as ogg_admin@T1C2) 2> add trandata osm_repapi.*

```

```

2018-01-08 21:25:42 INFO    OGG-15131  Logging of supplemental redo log
data is already enabled for table OSM_REPAPI.CLAIMS.

```

2018-01-08 21:25:42 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.CLAIMS.

2018-01-08 21:25:42 INFO OGG-15131 Logging of supplemental redo log data is already enabled for table OSM\_REPAPI.CLAIM\_STATUS.

2018-01-08 21:25:42 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.CLAIM\_STATUS.

2018-01-08 21:25:42 INFO OGG-15131 Logging of supplemental redo log data is already enabled for table OSM\_REPAPI.CUSTOMERS.

2018-01-08 21:25:42 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.CUSTOMERS.

2018-01-08 21:25:42 INFO OGG-15131 Logging of supplemental redo log data is already enabled for table OSM\_REPAPI.DISCOUNT\_TYPES.

2018-01-08 21:25:42 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.DISCOUNT\_TYPES.

2018-01-08 21:25:42 INFO OGG-15131 Logging of supplemental redo log data is already enabled for table OSM\_REPAPI.EDU\_LEVELS.

2018-01-08 21:25:42 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.EDU\_LEVELS.

2018-01-08 21:25:42 INFO OGG-15131 Logging of supplemental redo log data is already enabled for table OSM\_REPAPI.OSM\_DOCUMENTS.

2018-01-08 21:25:42 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.OSM\_DOCUMENTS.

2018-01-08 21:25:42 INFO OGG-15131 Logging of supplemental redo log data is already enabled for table OSM\_REPAPI.POLICIES.

2018-01-08 21:25:42 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.POLICIES.

2018-01-08 21:25:42 INFO OGG-15131 Logging of supplemental redo log data is already enabled for table OSM\_REPAPI.POLICY\_TYPES.

2018-01-08 21:25:42 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.POLICY\_TYPES.

2018-01-08 21:25:42 INFO OGG-15131 Logging of supplemental redo log data is already enabled for table OSM\_REPAPI.PROFESSIONS.

2018-01-08 21:25:42 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.PROFESSIONS.

2018-01-08 21:25:42 INFO OGG-15131 Logging of supplemental redo log data is already enabled for table OSM\_REPAPI.RANDOM\_DOMAIN.

2018-01-08 21:25:42 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.RANDOM\_DOMAIN.

2018-01-08 21:25:42 INFO OGG-15131 Logging of supplemental redo log data is already enabled for table OSM\_REPAPI.RANDOM\_FIRST.

2018-01-08 21:25:42 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.RANDOM\_FIRST.

2018-01-08 21:25:42 INFO OGG-15131 Logging of supplemental redo log data is already enabled for table OSM\_REPAPI.RANDOM\_LAST.

2018-01-08 21:25:42 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.RANDOM\_LAST.

2018-01-08 21:25:42 INFO OGG-15131 Logging of supplemental redo log data is already enabled for table OSM\_REPAPI.RANDOM\_MIDDLE.

2018-01-08 21:25:42 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.RANDOM\_MIDDLE.

2018-01-08 21:25:42 INFO OGG-15131 Logging of supplemental redo log data is already enabled for table OSM\_REPAPI.REGIONS.

2018-01-08 21:25:42 INFO OGG-15135 TRANDATA for instantiation CSN has been added on table OSM\_REPAPI.REGIONS.

2018-01-08 21:25:42 INFO OGG-10471 \*\*\*\*\* Oracle Goldengate support information on table OSM\_REPAPI.CLAIMS \*\*\*\*\*  
Oracle Goldengate support native capture on table OSM\_REPAPI.CLAIMS.  
Oracle Goldengate marked following column as key columns on table OSM\_REPAPI.CLAIMS: CUST\_NO, POL\_NO.

2018-01-08 21:25:42 INFO OGG-10471 \*\*\*\*\* Oracle Goldengate support information on table OSM\_REPAPI.CLAIM\_STATUS \*\*\*\*\*  
Oracle Goldengate support native capture on table OSM\_REPAPI.CLAIM\_STATUS.  
Oracle Goldengate marked following column as key columns on table OSM\_REPAPI.CLAIM\_STATUS: CLM\_STATUS.

2018-01-08 21:25:42 INFO OGG-10471 \*\*\*\*\* Oracle Goldengate support information on table OSM\_REPAPI.CUSTOMERS \*\*\*\*\*  
Oracle Goldengate support native capture on table OSM\_REPAPI.CUSTOMERS.  
Oracle Goldengate marked following column as key columns on table OSM\_REPAPI.CUSTOMERS: CUST\_NO.

2018-01-08 21:25:42 INFO OGG-10471 \*\*\*\*\* Oracle Goldengate support information on table OSM\_REPAPI.DISCOUNT\_TYPES \*\*\*\*\*  
Oracle Goldengate support native capture on table OSM\_REPAPI.DISCOUNT\_TYPES.  
Oracle Goldengate marked following column as key columns on table OSM\_REPAPI.DISCOUNT\_TYPES: DT\_CODE.

2018-01-08 21:25:42 INFO OGG-10471 \*\*\*\*\* Oracle Goldengate support information on table OSM\_REPAPI.EDU\_LEVELS \*\*\*\*\*  
Oracle Goldengate support native capture on table OSM\_REPAPI.EDU\_LEVELS.  
Oracle Goldengate marked following column as key columns on table OSM\_REPAPI.EDU\_LEVELS: EDUL\_ID.

2018-01-08 21:25:42 INFO OGG-10471 \*\*\*\*\* Oracle Goldengate support information on table OSM\_REPAPI.OSM\_DOCUMENTS \*\*\*\*\*  
Oracle Goldengate support native capture on table OSM\_REPAPI.OSM\_DOCUMENTS.  
Oracle Goldengate marked following column as key columns on table OSM\_REPAPI.OSM\_DOCUMENTS: NAME.

```

2018-01-08 21:25:43 INFO OGG-10471 ***** Oracle Goldengate support
information on table OSM_REPAPI.POLICIES *****
Oracle Goldengate support native capture on table OSM_REPAPI.POLICIES.
Oracle Goldengate marked following column as key columns on table
OSM_REPAPI.POLICIES: POL_NO.

2018-01-08 21:25:43 INFO OGG-10471 ***** Oracle Goldengate support
information on table OSM_REPAPI.POLICY_TYPES *****
Oracle Goldengate support native capture on table OSM_REPAPI.POLICY_TYPES.
Oracle Goldengate marked following column as key columns on table
OSM_REPAPI.POLICY_TYPES: PT_CODE.

2018-01-08 21:25:43 INFO OGG-10471 ***** Oracle Goldengate support
information on table OSM_REPAPI.PROFESSIONS *****
Oracle Goldengate support native capture on table OSM_REPAPI.PROFESSIONS.
Oracle Goldengate marked following column as key columns on table
OSM_REPAPI.PROFESSIONS: PROF_ID.

2018-01-08 21:25:43 INFO OGG-10471 ***** Oracle Goldengate support
information on table OSM_REPAPI.RANDOM_DOMAIN *****
Oracle Goldengate support native capture on table OSM_REPAPI.RANDOM_DOMAIN.
Oracle Goldengate marked following column as key columns on table
OSM_REPAPI.RANDOM_DOMAIN: RD_NAME.

2018-01-08 21:25:43 INFO OGG-10471 ***** Oracle Goldengate support
information on table OSM_REPAPI.RANDOM_FIRST *****
Oracle Goldengate support native capture on table OSM_REPAPI.RANDOM_FIRST.
Oracle Goldengate marked following column as key columns on table
OSM_REPAPI.RANDOM_FIRST: RF_NAME.

2018-01-08 21:25:43 INFO OGG-10471 ***** Oracle Goldengate support
information on table OSM_REPAPI.RANDOM_LAST *****
Oracle Goldengate support native capture on table OSM_REPAPI.RANDOM_LAST.
Oracle Goldengate marked following column as key columns on table
OSM_REPAPI.RANDOM_LAST: RL_NAME.

2018-01-08 21:25:43 INFO OGG-10471 ***** Oracle Goldengate support
information on table OSM_REPAPI.RANDOM_MIDDLE *****
Oracle Goldengate support native capture on table OSM_REPAPI.RANDOM_MIDDLE.
Oracle Goldengate marked following column as key columns on table
OSM_REPAPI.RANDOM_MIDDLE: RM_NAME.

2018-01-08 21:25:43 INFO OGG-10471 ***** Oracle Goldengate support
information on table OSM_REPAPI.REGIONS *****
Oracle Goldengate support native capture on table OSM_REPAPI.REGIONS.
Oracle Goldengate marked following column as key columns on table
OSM_REPAPI.REGIONS: REG_ID.

GGSCI (linux5acdr.oracle.com as ogg_admin@T1C2) 3>

```

## Configure and Define the Primary Group

Defining the primary group composed of three steps: Prepare the parameter file, create the extract and link the primary extract with the local trail - sa.



## Prepare the Parameter File

The basic parameter files are listed. The first line is the Extract name used by the ADD EXTRACT statement.

```
GGSCI (linux5acdr.oracle.com as ogg_admin@T1C2) 3> edit params et1c201
```

```
EXTRACT ET1C201
SETENV (ORACLE_SID = T1C2)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.2.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@T1C2, PASSWORD oracle
EXTTRAIL ./dirdat/sa
TABLE osm repapi.*;
```

## Define the Primary Integrated Extract

The primary extract read the database transaction logs. The name of the Extract is ET1C201.

Register the primary extract

```
GGSCI (linux5acdr.oracle.com as ogg_admin@T1C2) 8> register extract ET1C201
database
```

```
2018-01-08 21:31:13 INFO    OGG-02003 Extract ET1C201 successfully
registered with database at SCN 3140791.
```

```
GGSCI (linux5acdr.oracle.com as ogg_admin@T1C2) 9>
```

Define the extract

```
GGSCI (linux5acdr.oracle.com as ogg_admin@T1C2) 10> ADD EXTRACT ET1C201,
INTEGRATED TRANLOG, BEGIN now
EXTRACT (Integrated) added.
```

```
GGSCI (linux5acdr.oracle.com as ogg_admin@T1C2) 11>
```

## Define the Local Trail

Link the local trail with primary extract - ES1C201.

```
GGSCI (linux5acdr.oracle.com as ogg_admin@T1C2) 12> ADD EXTTRAIL
./dirdat/sa, EXTRACT ET1C201 , MEGABYTES 32
EXTTRAIL added.
```

```
GGSCI (linux5acdr.oracle.com as ogg_admin@T1C2) 13>
```

## Configure and Define the Secondary Group

The secondary group (Data pump) PT1C201 routes data to the target system.

```
GGSCI (linux5acdr.oracle.com as ogg_admin@T1C2) 13> edit params PT1C201
```

```
EXTRACT PT1C201  
USERID ogg_admin@T1C2, PASSWORD oracle  
NOPASSTHRU  
RMTHOST 192.168.1.40, MGRPORT 7840  
RMTRAIL ./dirdat/ta  
TABLE osm repapi.*;
```

## Define the Secondary Extract

Oracle GoldenGate best practices uses a data pump to offload the primary extract from sending data to the target system. The following GGSCI command define the secondary capture (data pump) – PS1C201.

```
GGSCI (linux5acdr.oracle.com as ogg_admin@T1C2) 14> ADD EXTRACT PT1C201,  
EXTRTRAILSOURCE ./dirdat/sa  
EXTRACT added.
```

```
GGSCI (linux5acdr.oracle.com as ogg_admin@T1C2) 15>
```

## Define the Remote Trail

Link the remote trail with primary extract - PS1C201.

```
GGSCI (linux5acdr.oracle.com as ogg_admin@T1C2) 15> ADD RMTTRAIL  
./dirdat/ta, EXTRACT pt1c201, MEGABYTES 32  
RMTTRAIL added.
```

```
GGSCI (linux5acdr.oracle.com as ogg_admin@T1C2) 16>
```

## Prepare and Define the Delivery Group

The Replicat is part of the delivery group. The Replicat parameter file, and definition follows. Note, the Replicat needs a checkpoint table which has been already created during the initial-load setup.

## Prepare the Parameter File

The parameter file for the Replicat – RS1C201. Notice, the Replicat invoke an SQL statement to fetch the target system current LOCALTIMESTAMP. The purpose of this column is the implementation of the built-in Conflict Detection and Resolution (CDR), or some other purpose such as calculating the Replicat lagging.

```
GGSCI (linux4.oracle.com) 10> edit params RT1C2
```

```
REPLICAT RT1C2
SETENV (ORACLE_SID = S1C2)
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@S1C2, PASSWORD oracle
-- BATCHSQL
ASSUMETARGETDEFS
REPERERROR (DEFAULT, DISCARD)
DISCARDFILE /u01/app/oracle/product/12.3.0/gghome_1/dirrpt/RT1C2.dsc, PURGE
--
MAP osm_repapi.customers, TARGET osm_repapi.customers;
MAP osm_repapi.policies, TARGET osm_repapi.policies;
MAP osm_repapi.claims, TARGET osm_repapi.claims;
```

## Create the Checkpoint Table

The checkpoint database table is the default for the Replicat. Before attempting to define the Replicat, the checkpoint table should be created. The steps are show below:

4. Create or modify the GLOBALS parameter file by adding the parameter CHECKPOINTTABLE followed by the database schema and table name.
5. Login to the Database using GGSCI
6. Create the checkpoint table by run the GGSCI command ADD CHECKPINTTABLE

```
[oracle@linux4acdr gghome_1]$ cd $OGG_HOME
[oracle@linux4acdr gghome_1]$ vi GLOBALS
```

```
checkpointtable ogg_admin.chkpttab
```

Save and verify the changes.

```
[oracle@linux4acdr gghome_1]$ cat GLOBALS
checkpointtable ogg_admin.chkpttab

[oracle@linux4acdr gghome_1]$
```

Create the checkpoint table.

```
[oracle@linux4acdr gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE_12.3.0.1.0_PLATFORMS_170721.0154_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (linux4.oracle.com) 1> dblogin userid ogg_admin@S1C2, password oracle
Successfully logged into database.

GGSCI (linux4.oracle.com as ogg_admin@S1C2) 2> add checkpointtable

No checkpoint table specified. Using GLOBALS specification
ogg admin.chkpttab...

Successfully created checkpoint table ogg_admin.chkpttab.

GGSCI (linux4.oracle.com as ogg_admin@S1C2) 3>
```

## Define the Replicat

Use the add REPLICAT GGSCI command to define the Replicat.

```
GGSCI (linux4.oracle.com as ogg_admin@S1C2) 3> ADD REPLICAT rtlc2,  
integrated, parallel, EXTTRAIL ./dirdat/ta  
REPLICAT (Parallel) added.

GGSCI (linux4.oracle.com as ogg_admin@S1C2) 4>
```

## Start the Extracts and Replicat Processes

Start the Primary and Secondary extracts (ET1C201 and PT1C201).

```
GGSCI (linux5acdr.oracle.com as ogg_admin@T1C2) 17> start *

Sending START request to MANAGER ...
EXTRACT ET1C201 starting

Sending START request to MANAGER ...
EXTRACT PT1C201 starting
REPLICAT RS1C2 is already running.

GGSCI (linux5acdr.oracle.com as ogg_admin@T1C2) 18> info all
```

```

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
EXTRACT      RUNNING      ET1C201    00:00:00      00:18:27
EXTRACT      RUNNING      PT1C201    00:00:00      00:15:23
REPLICAT     RUNNING      RS1C2      00:00:00      00:00:00

GGSCI (linux5acdr.oracle.com as ogg_admin@T1C2) 19>

```

Start the Replicat process (RT1C101).

```

GGSCI (linux4.oracle.com as ogg_admin@S1C2) 4> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
EXTRACT      RUNNING      ES1C201    00:00:00      00:00:06
EXTRACT      RUNNING      PS1C201    00:00:00      00:00:07
REPLICAT     STOPPED      RT1C2      00:00:00      00:03:26

GGSCI (linux4.oracle.com as ogg_admin@S1C2) 5> start *
EXTRACT ES1C201 is already running.
EXTRACT PS1C201 is already running.

Sending START request to MANAGER ...
REPLICAT RT1C2 starting

GGSCI (linux4.oracle.com as ogg_admin@S1C2) 6> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
MANAGER      RUNNING
EXTRACT      RUNNING      ES1C201    00:00:00      00:00:00
EXTRACT      RUNNING      PS1C201    00:00:00      00:00:02
REPLICAT     RUNNING      RT1C2      00:00:00      00:03:31

GGSCI (linux4.oracle.com as ogg_admin@S1C2) 7>

```

## Run and Verify Data Movement from Source to Target

Run the application from the source system and generate statistics on source and target. The provided PL/SQL workload simulator uses different actual parameters to create several transaction levels. Using the default value generate a reasonable workload which insert 5000 rows with a transaction interval of 5 seconds. Invoke the workload by entering the name of the procedure – call\_trans\_load without parameters.

```
SQL>exec call_trans_load;
```

```
[oracle@linux5acdr gghome_1]$ . oraenv
ORACLE_SID = [T1C2] ?
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@linux5acdr gghome_1]$ sqlplus osm_repapi/oracle

SQL*Plus: Release 12.2.0.1.0 Production on Mon Jan 8 21:56:54 2018

Copyright (c) 1982, 2016, Oracle. All rights reserved.

Last Successful login time: Sat Jan 06 2018 17:11:54 +03:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

SQL> sho user
USER is "OSM_REPAPI"
SQL> exec call_trans_load;
```

Open a new terminal on the target system, verify rows count using SQL\*Plus.

```
[oracle@linux4acdr ~]$ . oraenv
ORACLE_SID = [S1C2] ?
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@linux4acdr ~]$ sqlplus /nolog

SQL*Plus: Release 12.2.0.1.0 Production on Mon Jan 8 21:58:44 2018

Copyright (c) 1982, 2016, Oracle. All rights reserved.

SQL> conn osm_repapi/oracle
Connected.
SQL> select count(*) from customers;

  COUNT(*)
-----
    242052

SQL> /

  COUNT(*)
-----
    242062

SQL>
```

Display statistics from Oracle GoldenGate.

```
[oracle@linux4acdr gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE_12.3.0.1.0_PLATFORMS_170721.0154_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.
```

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (linux4.oracle.com) 1> **info all**

| Program  | Status  | Group   | Lag at Chkpt | Time Since Chkpt |
|----------|---------|---------|--------------|------------------|
| MANAGER  | RUNNING |         |              |                  |
| EXTRACT  | RUNNING | ES1C201 | 00:00:02     | 00:00:05         |
| EXTRACT  | RUNNING | PS1C201 | 00:00:00     | 00:00:00         |
| REPLICAT | RUNNING | RT1C2   | 00:00:00     | 00:00:08         |

GGSCI (linux4.oracle.com) 2> **stats replicat RT1C2**

Sending STATS request to REPLICAT RT1C2 ...

Parallel Replicat Statistics:

|                        |        |
|------------------------|--------|
| Total transactions     | 825.00 |
| Large transactions     | 0.00   |
| Total metadata changes | 0.00   |
| DDL operations         | 0.00   |
| Replicated procedures  | 0.00   |
| Event Actions          | 0.00   |
| Datatype functionality | 0.00   |
| Store Procedures       | 0.00   |
| Checkpoint errors      | 0.00   |
| Transactions retried   | 0.00   |

Start of Statistics at 2018-01-08 22:00:50.

Replicating from OSM\_REPAPI.CUSTOMERS to OSM\_REPAPI.CUSTOMERS:

\*\*\* Total statistics since 2018-01-08 21:57:29 \*\*\*

|                  |        |
|------------------|--------|
| Total inserts    | 371.00 |
| Total updates    | 0.00   |
| Total deletes    | 0.00   |
| Total discards   | 0.00   |
| Total operations | 371.00 |

\*\*\* Daily statistics since 2018-01-08 21:57:29 \*\*\*

|                  |        |
|------------------|--------|
| Total inserts    | 371.00 |
| Total updates    | 0.00   |
| Total deletes    | 0.00   |
| Total discards   | 0.00   |
| Total operations | 371.00 |

\*\*\* Hourly statistics since 2018-01-08 22:00:00 \*\*\*

|                  |       |
|------------------|-------|
| Total inserts    | 90.00 |
| Total updates    | 0.00  |
| Total deletes    | 0.00  |
| Total discards   | 0.00  |
| Total operations | 90.00 |

```

*** Latest statistics since 2018-01-08 21:57:29 ***
    Total inserts          371.00
    Total updates          0.00
    Total deletes          0.00
    Total discards         0.00
    Total operations       371.00

Replicating from OSM_REPAPI.POLICIES to OSM_REPAPI.POLICIES:

*** Total statistics since 2018-01-08 21:57:29 ***
    Total inserts          372.00
    Total updates          0.00
    Total deletes          0.00
    Total discards         0.00
    Total operations       372.00

*** Daily statistics since 2018-01-08 21:57:29 ***
    Total inserts          372.00
    Total updates          0.00
    Total deletes          0.00
    Total discards         0.00
    Total operations       372.00

*** Hourly statistics since 2018-01-08 22:00:00 ***
    Total inserts          90.00
    Total updates          0.00
    Total deletes          0.00
    Total discards         0.00
    Total operations       90.00

*** Latest statistics since 2018-01-08 21:57:29 ***
    Total inserts          372.00
    Total updates          0.00
    Total deletes          0.00
    Total discards         0.00
    Total operations       372.00

Replicating from OSM_REPAPI.CLAIMS to OSM_REPAPI.CLAIMS:

*** Total statistics since 2018-01-08 21:57:29 ***
    Total inserts          56.00
    Total updates          0.00
    Total deletes          0.00
    Total discards         0.00
    Total operations       56.00

*** Daily statistics since 2018-01-08 21:57:29 ***
    Total inserts          56.00
    Total updates          0.00
    Total deletes          0.00
    Total discards         0.00
    Total operations       56.00

*** Hourly statistics since 2018-01-08 22:00:00 ***
    Total inserts          15.00
    Total updates          0.00
    Total deletes          0.00
    Total discards         0.00

```



```

Total operations                                15.00

*** Latest statistics since 2018-01-08 21:57:29 ***
Total inserts                                56.00
Total updates                                0.00
Total deletes                                0.00
Total discards                               0.00
Total operations                             56.00

End of Statistics.

```

```
GGSCI (linux4.oracle.com) 3>
```

## Disable Replicat Looping on Source (S1C2)

Modify the Extract parameter file by including the parameter EXCLUDEUSER

```
GGSCI (linux4.oracle.com) 4> edit params ES1C201
```

```

EXTRACT ES1C201
SETENV (ORACLE_SID = S1C2)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.2.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN AMERICA.AL32UTF8)
USERID ogg_admin@S1C2, PASSWORD oracle
TRANLOGOPTIONS EXCLUDEUSER ogg_admin
EXTTRAIL ./dirdat/sa
TABLE osm_repapi.*;

```

```
GGSCI (linux4.oracle.com) 5> stop ES1C201
```

```

Sending STOP request to EXTRACT ES1C201 ...
Request processed.

```

```
GGSCI (linux4.oracle.com) 8> start ES1C201
```

```

Sending START request to MANAGER ...
EXTRACT ES1C201 starting

```

```
GGSCI (linux4.oracle.com) 9> info all
```

| Program  | Status  | Group   | Lag at Chkpt | Time Since Chkpt |
|----------|---------|---------|--------------|------------------|
| MANAGER  | RUNNING |         |              |                  |
| EXTRACT  | RUNNING | ES1C201 | 00:00:00     | 00:00:01         |
| EXTRACT  | RUNNING | PS1C201 | 00:00:00     | 00:00:05         |
| REPLICAT | RUNNING | RT1C2   | 00:00:00     | 00:00:06         |

```
GGSCI (linux4.oracle.com) 10>
```

## Disable Replicat Looping on Target (T1C2)

```
GGSCI (linux5acdr.oracle.com) 1> edit params ET1C201
```

```
EXTRACT ET1C201
SETENV (ORACLE_SID = T1C2)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.2.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@T1C2, PASSWORD oracle
TRANLOGOPTIONS EXCLUDEUSER ogg_admin
EXTTRAIL ./dirdat/sa
TABLE osm_repapi.*;
```

```
GGSCI (linux5acdr.oracle.com) 26> stop RS1C2
```

```
Sending STOP request to REPLICAT RS1C2 ...
Request processed.
```

```
GGSCI (linux5acdr.oracle.com) 27> info all
```

| Program  | Status  | Group   | Lag at Chkpt | Time Since Chkpt |
|----------|---------|---------|--------------|------------------|
| MANAGER  | RUNNING |         |              |                  |
| EXTRACT  | RUNNING | ET1C201 | 00:00:00     | 00:00:01         |
| EXTRACT  | RUNNING | PT1C201 | 00:00:00     | 00:00:06         |
| REPLICAT | STOPPED | RS1C2   | 00:00:00     | 00:00:02         |

```
GGSCI (linux5acdr.oracle.com) 35> start RS1C2
```

```
Sending START request to MANAGER ...
REPLICAT RS1C2 starting
```

```
GGSCI (linux5acdr.oracle.com) 36> info all
```

| Program  | Status  | Group   | Lag at Chkpt | Time Since Chkpt |
|----------|---------|---------|--------------|------------------|
| MANAGER  | RUNNING |         |              |                  |
| EXTRACT  | RUNNING | ET1C201 | 00:00:00     | 00:00:00         |
| EXTRACT  | RUNNING | PT1C201 | 00:00:00     | 00:00:05         |
| REPLICAT | RUNNING | RS1C2   | 00:00:00     | 00:00:10         |

```
GGSCI (linux5acdr.oracle.com) 37>
```

## Deploy Automatic Conflict Detection and Resolution on Source (S1C2)

The final workshop implements Active-Active Oracle GoldenGate 12c topology. The unidirectional topology is extended by adding Extract and Replicat parameters to

support Conflict detection and Resolution (CDR). The following table lists the parameter and description.

```
[oracle@linux4acdr ~]$ sqlplus /nolog

SQL*Plus: Release 12.2.0.1.0 Production on Mon Jan 8 22:23:35 2018

Copyright (c) 1982, 2016, Oracle. All rights reserved.

SQL> conn / as sysdba
Connected.
SQL> grant all on DBMS_GOLDENGATE_ADM to ogg_admin;

Grant succeeded.

SQL>
```

```
SQL> conn ogg_admin/oracle
Connected.
SQL> begin
2      DBMS_GOLDENGATE_ADM.ADD_AUTO_CDR(
3          schema_name => 'OSM_REPAPI',
4          table_name => 'customers');
5  end;
6  /

PL/SQL procedure successfully completed.

SQL> begin
2      DBMS_GOLDENGATE_ADM.ADD_AUTO_CDR(
3          schema_name => 'OSM_REPAPI',
4          table_name => 'policies');
5  end;
6  /

PL/SQL procedure successfully completed.

SQL> begin
2      DBMS_GOLDENGATE_ADM.ADD_AUTO_CDR(
3          schema_name => 'OSM_REPAPI',
4          table_name => 'claims');
5  end;
6  /

PL/SQL procedure successfully completed.

SQL>
```

```
GGSCI (linux4.oracle.com) 27> edit params RT1C2
```

```
REPLICAT RT1C2
SETENV (ORACLE_SID = S1C2)
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@S1C2, PASSWORD oracle
-- BATCHSQL
```

```

ASSUMETARGETDEFS
REFERROR (DEFAULT, DISCARD)
DISCARDFILE /u01/app/oracle/product/12.3.0/gghome_1/dirrpt/RT1C2.dsc, PURGE,
megabytes 1000
--
MAPINVISIBLECOLUMNS
MAP osm repapi.customers, TARGET osm repapi.customers;
MAP osm_repapi.policies, TARGET osm_repapi.policies;
MAP osm repapi.claims, TARGET osm repapi.claims;

```

## Deploy Automatic Conflict Detection and Resolution on Source (T1C2)

The final workshop implements Active-Active Oracle GoldenGate 12c topology. The unidirectional topology is extended by adding Extract and Replicat parameters to support Conflict detection and Resolution (CDR). The following table lists the parameter and description.

```

[oracle@linux5acdr ~]$ sqlplus /nolog

SQL*Plus: Release 12.2.0.1.0 Production on Mon Jan 8 22:32:14 2018

Copyright (c) 1982, 2016, Oracle. All rights reserved.

SQL> conn / as sysdba
Connected.
SQL> grant all on DBMS_GOLDENGATE_ADM to ogg_admin;

Grant succeeded.

SQL>

```

```

SQL> conn ogg_admin/oracle
Connected.
SQL> begin
2     DBMS_GOLDENGATE_ADM.ADD_AUTO_CDR(
3         schema_name => 'OSM_REPAPI',
4         table_name => 'customers');
5 end;
6 /

PL/SQL procedure successfully completed.

SQL> begin
2     DBMS_GOLDENGATE_ADM.ADD_AUTO_CDR(
3         schema_name => 'OSM_REPAPI',
4         table_name => 'policies');
5 end;
6 /

PL/SQL procedure successfully completed.

```

```
SQL> begin
2   DBMS_GOLDENGATE_ADM.ADD_AUTO_CDR(
3       schema_name =>'OSM_REPAPI',
4       table_name =>'claims');
5   end;
6   /
```

PL/SQL procedure successfully completed.

```
SQL>
```

```
GGSCI (linux5acdr.oracle.com) 2> edit params RS1C2
```

```
REPLICAT RS1C2
SETENV (ORACLE_SID = T1C2)
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@T1C2, PASSWORD oracle
-- BATCHSQL
ASSUMETARGETDEFS
REPERERROR (DEFAULT, DISCARD)
DISCARDFILE /u01/app/oracle/product/12.3.0/gghome_1/dirrpt/RT1C2.dsc, PURGE,
megabytes 1000
--
MAPINVISIBLECOLUMNS
MAP osm_repapi.customers, TARGET osm_repapi.customers;
MAP osm_repapi.policies, TARGET osm_repapi.policies;
MAP osm_repapi.claims, TARGET osm_repapi.claims;
```

## Testing Automatic Conflict Detection and Resolution

The following perform UPDATE Automatic CDR.

On Source

```
SQL> conn osm_repapi/oracle
Connected.
SQL> select max(cust_no) from customers;

MAX(CUST_NO)
-----
1242931

SQL> select cust_email
2   from customers
3  where cust_no=1242931;

CUST_EMAIL
-----
Philip.Alsuwaidi@gmail.com

SQL>
```

On Target

```

SQL> conn osm_repapi/oracle
Connected.
SQL> select max(cust_no) from customers;

MAX(CUST NO)
-----
          1242931

SQL> select cust_email
2   from customers
3  where cust_no=1242931;

CUST EMAIL
-----
Philip.Alsuwaidi@gmail.com

SQL>

```

## Stop the data pump On Source

```

GGSCI (linux4.oracle.com) 30> stop PS1C201

Sending STOP request to EXTRACT PS1C201 ...
Request processed.

GGSCI (linux4.oracle.com) 31> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
-----
MANAGER      RUNNING
EXTRACT      RUNNING     ES1C201    00:00:00     00:00:02
EXTRACT      STOPPED     PS1C201    00:00:00     00:00:12
REPLICAT     RUNNING     RT1C2      00:00:00     00:00:08

GGSCI (linux4.oracle.com) 32>

```

## Stop the data pump On Target

```

GGSCI (linux5acdr.oracle.com) 4> stop PT1C201

Sending STOP request to EXTRACT PT1C201 ...
Request processed.

GGSCI (linux5acdr.oracle.com) 5> info all

Program      Status      Group      Lag at Chkpt  Time Since Chkpt
-----
MANAGER      RUNNING
EXTRACT      RUNNING     ET1C201    00:00:00     00:00:09
EXTRACT      STOPPED     PT1C201    00:00:00     00:00:03
REPLICAT     RUNNING     RS1C2      00:00:00     00:00:10

```

```
GGSCI (linux5acdr.oracle.com) 6>
```

## On Source

```
SQL> update customers
  2 set cust_email='Philip.Alsuwaidi@yahoo.com'
  3 where cust_no=1242931;

1 row updated.

SQL> commit;

Commit complete.

SQL>
```

## On Target

```
SQL> update customers
  2 set cust_email='Philip.Alsuwaidi@hotmail.com'
  3 where cust_no=1242931;

1 row updated.

SQL> commit;

Commit complete.

SQL>
```

## Start the Data Pump on Source

```
GGSCI (linux4.oracle.com) 33> start *
EXTRACT ES1C201 is already running.

Sending START request to MANAGER ...
EXTRACT PS1C201 starting
REPLICAT RT1C2 is already running.

GGSCI (linux4.oracle.com) 34>

GGSCI (linux4.oracle.com) 34> info all
```

| Program  | Status  | Group   | Lag at Chkpt | Time Since Chkpt |
|----------|---------|---------|--------------|------------------|
| MANAGER  | RUNNING |         |              |                  |
| EXTRACT  | RUNNING | ES1C201 | 00:00:00     | 00:00:09         |
| EXTRACT  | RUNNING | PS1C201 | 00:00:00     | 00:00:05         |
| REPLICAT | RUNNING | RT1C2   | 00:00:00     | 00:00:06         |

```
GGSCI (linux4.oracle.com) 35>
```

## Start the Data Pump on Target

```
GGSCI (linux5acdr.oracle.com) 7> start *
EXTRACT ET1C201 is already running.

Sending START request to MANAGER ...
EXTRACT PT1C201 starting
REPLICAT RS1C2 is already running.

GGSCI (linux5acdr.oracle.com) 8> info all
```

| Program  | Status  | Group   | Lag at Chkpt | Time Since Chkpt |
|----------|---------|---------|--------------|------------------|
| MANAGER  | RUNNING |         |              |                  |
| EXTRACT  | RUNNING | ET1C201 | 00:00:00     | 00:00:01         |
| EXTRACT  | RUNNING | PT1C201 | 00:00:00     | 00:01:50         |
| REPLICAT | RUNNING | RS1C2   | 00:00:00     | 00:00:05         |

```
GGSCI (linux5acdr.oracle.com) 9>
```

## Verify Automatic CDR by selecting from source

```
SQL> select cust_email
2 from customers
3 where cust_no=1242931;
```

| CUST_EMAIL                   |
|------------------------------|
| Philip.Alsuwaidi@hotmail.com |

```
SQL>
```

## Verify Automatic CDR by selecting from target

```
SQL> select cust_email
2 from customers
3 where cust_no=1242931;
```

| CUST EMAIL                   |
|------------------------------|
| Philip.Alsuwaidi@hotmail.com |

```
SQL>
```



## Summary

Enterprise running business critical systems highly benefit from n-way Active-Active database architecture from a business and technical prospective, where n-Way Active-Active database becomes the foundation for delivering 24/7 business continuity. From the business prospective business units are isolated where users connect to their dedicated database, from a technical prospective this site is part of the high-availability footprint. Expanding by adding nodes to the footprint is delivered online without compromising the target database of the instantiation.

With the release of Oracle Database 12c (12.2) and Oracle GoldenGate 12c (12.3), there is not better technique of handling conflict detection and resolution (CDR). The technique has been automated by calling Oracle database API, where the CDR Metadata stored in the database data dictionary. The technique uses an invisible column for deciding on the conflict resolution. The default CDR is the timestamp for DML statements INSERT, UPDATE and DELETE.

*What is the package to enable CDR?*

Members of the package DBMS\_GOLDENGATE\_ADM are invoked to enable and manage the lifecycle of the automated built-in conflict detection and resolution. The package allows the following:

- Add conflict detection and resolutions
- Remove conflict detection and resolutions
- Add conflict detection and resolution column group
- Remove conflict detection and resolution column group

Though n-way Active-Active database architecture is an appealing technology, a major criticism is the applications interoperability. The application *logic* running at the application server tiers must be GoldenGate-aware. For example, some of the packaged applications have their own way of handling sequences, leading to unworkable data replication in n-way Active-Active database architecture. However; when the application is in-house developed that is the best match. In all case, the applications must be certified for interoperability with Oracle GoldenGate.

However; Oracle Real Application Cluster (RAC) is irreplaceable with Oracle GoldenGate 12c, to the contrary, Oracle GoldenGate 12c becomes highly available when running in RAC environment, where in the event of node failure, it transparently failover to the surviving instance, where Oracle GoldenGate 12c is configured for automatic re-start.

# Environment Setup



## Overview

Nowadays, the vast majority of computing production environments is built using virtualization; the predominant reasons are less cost and manageability. This environment is built using Oracle virtualization solution, Oracle VirtualBox.

The Enterprise Data Warehouse (EDW) environment composed of four systems:

- The primary database system
- The standby database system
- The target database system
- The big data Kafka system

This document explains the installation and Oracle VirtualBox and importing the virtual machines. Figure 1 shows and describes the four systems.

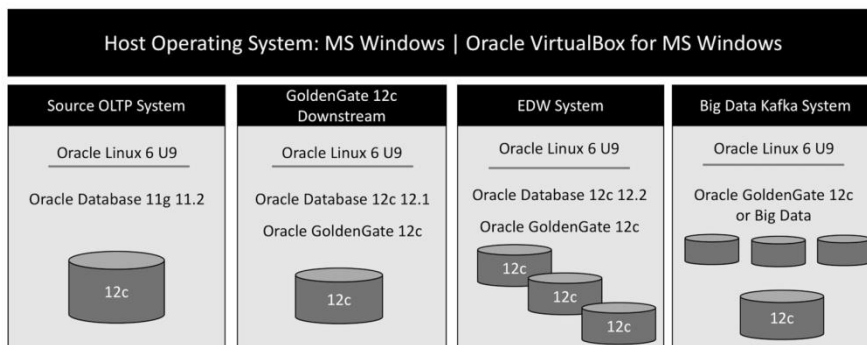


*Figure 1: VirtualBox Environment*

Though the workshop has been developed on Oracle VirtualBox for MS Windows, the import OVA files are usable with supported host operating systems such as MAC OS and Linux.

# Systems Properties

Tables 1, 2, 3 and 4 show the details for the systems depicted on Figure 1. These systems may be tightly connected on the same server or loosely connected.

## Source Database System

|                    |                   |
|--------------------|-------------------|
| Hostname           | oltp01.oracle.com |
| OS root password   | oracle            |
| OS oracle password | oracle            |
| Global Name        | OLTP.oracle.com   |
| SYS                | oracle_4U         |
| SYSTEM             | oracle_4U         |
| Database Version   | 11.2.0.4.0        |
| Archive Log Mode   | Enabled           |

*Table 1: Transactional System details*

## Downstream GoldenGate Database Mining System

|                    |                  |
|--------------------|------------------|
| Hostname           | dss01.oracle.com |
| OS root password   | oracle           |
| OS oracle password | oracle           |
| Global Name        | DSDB.oracle.com  |
| SYS                | oracle_4U        |
| SYSTEM             | oracle_4U        |
| Database Version   | 12.1.0.2.0       |
| Archive Log Mode   | Enabled          |

*Table 2: Oracle GoldenGate 12c Downstream System details*

## EDW System

|                    |                  |
|--------------------|------------------|
| Hostname           | edw01.oracle.com |
| OS root password   | oracle           |
| OS oracle password | oracle           |
| Global Name        | EDW.oracle.com   |
| SYS                | oracle_4U        |
| SYSTEM             | oracle_4U        |
| Database Version   | 12.2.0.1.0       |
| Archive Log Mode   | Disabled         |

*Table 3: EDW System details*

# Big Data Kafka System

|                    |                    |
|--------------------|--------------------|
| Hostname           | kafka01.oracle.com |
| OS root password   | oracle             |
| OS oracle password | oracle             |
| Global Name        | kafka.oracle.com   |
| SYS                | oracle_4U          |
| SYSTEM             | oracle_4U          |
| Database Version   | 12.2.0.1.0         |
| Archive Log Mode   | Enabled            |

*Table 4: Kafka System details*

## Install Oracle VirtualBox

The Oracle VirtualBox is two parts installations. Install Oracle VirtualBox and Oracle VirtualBox Extensions Pack. Figure 1 show installation of Oracle VirtualBox for MS Windows. The Oracle VirtualBox is available on these operating systems:

- MS Windows
- MAC OS
- Linux
- Solaris

*Figure 2: Installation of Oracle VirtualBox*

The Oracle VirtualBox Extension Installation proceeds quickly after installing the VirtualBox as shown next:

Figure 3: Installation of Oracle VirtualBox

## Import Virtual Machines

The import option creates the three virtual machines. Navigate File -> Import Appliance, and select the OVA file: OracleEDW01.OVA. Figure 4 shows the import appliance. Check Reinitialize the MAC address and click on Import.

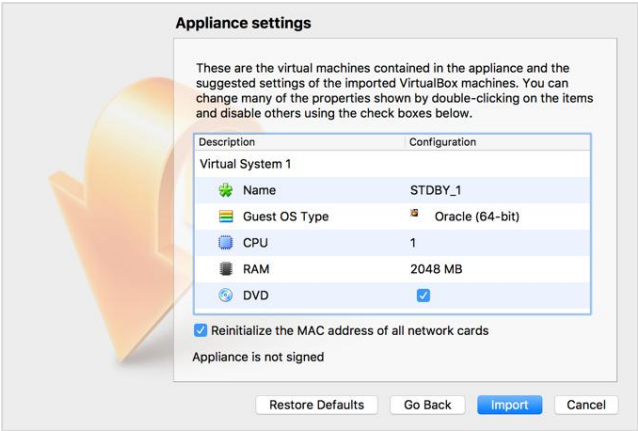


Figure 4: Oracle VirtualBox Import Appliance

# Virtual Machines Memory Settings

The setting must be done before starting the virtual machines. Figure 2 shows the Oracle VirtualBox setting.

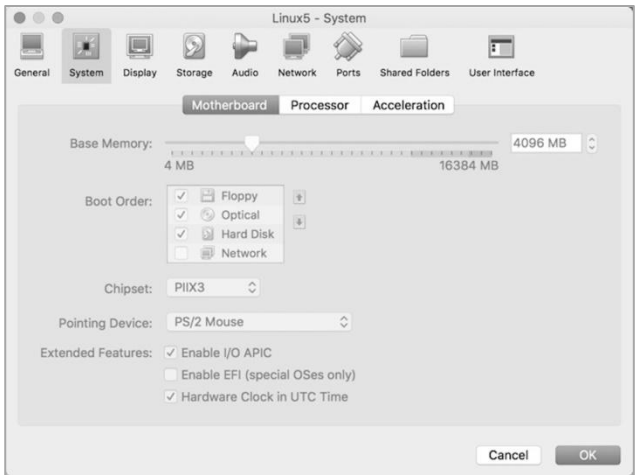


Figure 5: VirtualBox Memory Settings

Table 4 suggests the memory allocation according to the available host system physical memory. The recommended host memory size is 16GB or higher.

| Host Memory Size | Source | Downstream | Target (EDW) |
|------------------|--------|------------|--------------|
| 8GB              | 2GB    | 2GB        | 3GB          |
| 16GB             | 3GB    | 3GB        | 8GB          |
| 32GB             | 6GB    | 8GB        | 16GB         |

Table 4: Virtual Machines Memory Settings

Though the workshop developed on a system installed with 128GB RAM, but it was tested on 128GB and 16GB, so that it's workable for the majority of the audience, running a system with 16GB.

# Virtual Machines Network Settings

The virtual machine setting is shown in Figure 6. Follow the instructions shown while attached to: Bridged Adapter.

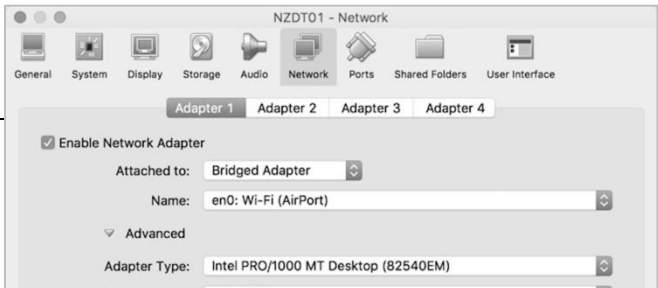


Figure 6: Network Settings

# Power on the Virtual Machines

T

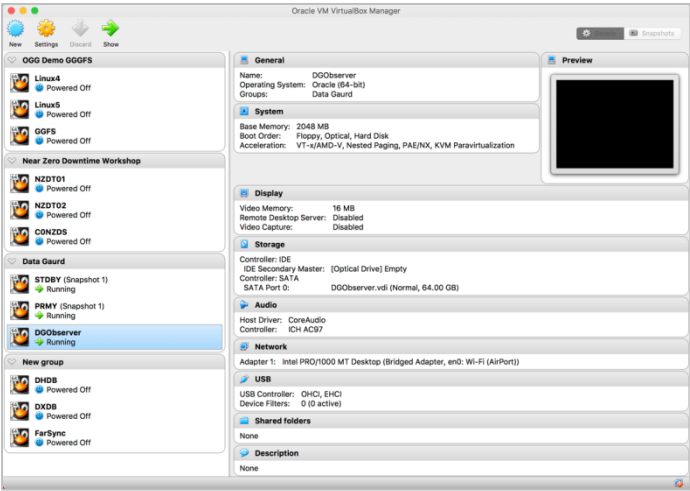


Figure X: Power on the Virtual Machines

# Configure the Local Domain Name System (DNS)

Edit the local DNS file /etc/hosts to add the three virtual machine servers networking information.

```
[oracle@linux4acdr Disk1]$ cat /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4
localhost4.localhostdomain4
:::1        localhost localhost.localdomain localhost6
localhost6.localhostdomain6
192.168.1.40    linux4acdr.oracle.com  linux4acdr
192.168.1.50    linux5acdr.oracle.com  linux5acdr
192.168.1.30    linux5acdr.oracle.com  linux5acdr

[oracle@linux4acdr Disk1]$
```

## Configure Networking

In order to reach the virtual machine from the MS windows, you have two configuration options (may be more).

- Create MS Windows Loopback (Virtual network)
- Connect to dedicated DHCP network

The workshop uses the second option using wireless WiFi as illustrated in Figure x. The wireless router must be configured to send a DHCP TCP/IP address to the host only with the format: 192.168.1.X and 255.255.255.0, the virtual machines Linux system has a static IP addresses within the same subnet configuration as shown in Table x.

| Machine    | TCP/IP Address | Subnet        |
|------------|----------------|---------------|
| Host       | DHCP           | DHCP          |
| Source     | 192.168.1.40   | 255.255.255.0 |
| Downstream | 192.168.1.30   | 255.255.255.0 |
| Target     | 192.168.1.50   | 255.255.255.0 |

Table X: Network TCP/IP Configuration

## Login using PuTTY Terminals for MS Windows

Test login as shown in Figure x, Figure y and Figure z.

|            |                              |
|------------|------------------------------|
| IP Address | 192.168.1.40<br>192.168.1.30 |
|------------|------------------------------|



|                 |              |
|-----------------|--------------|
|                 | 192.168.1.50 |
| <b>Username</b> | root         |
| <b>Password</b> | oracle       |

*Figure x: Login to Systems*

## PuTTY Configuration

Save PuTTY configuration for subsequent logins.

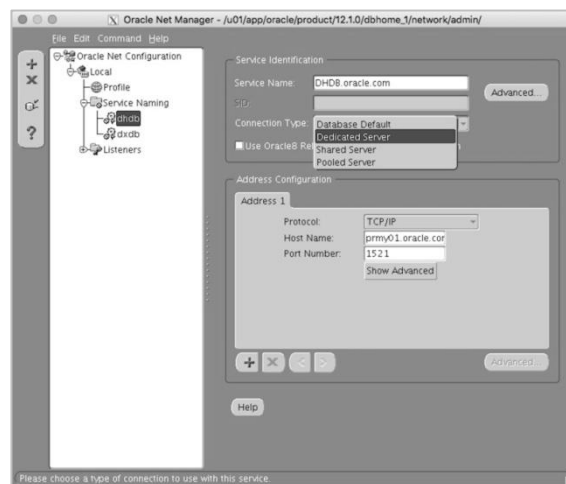
# Configure SQLNET

SQLNET configuration files (listener.ora and tnsnames.ora) already generated and tested. Use NETCA and NETMGR tools to modify the configuration files. SQLNET configuration files are stored under \$ORACLE\_HOME/network/admin, unless explicitly defined by the environment variable TNS\_ADMIN.

Use the NETCA to configure the listener and connect strings.

*Figure X: NETCA utility*

Use the NETMGR to register the listener with the database.



## PING Utility

Use the Oracle ping utility (tnsping) to verify the connect strings are successful. The following

## Automated Management

The automation of tasks avoids unnecessary manual regularly executed commands and scripts. This section automates tasks and size the database component directly related to fragile

### Enable Archive Log Files Purge

Use RMAN to delete archive log files by executing the shell script: \$HOME/ClearArch.sh. *Do not follow this purge technique in production environments; rather use the Recovery Manager (RMAN) ARCHIVE LOG retention policy.*

```
#!/bin/sh
export ORACLE_SID=BRM
export ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
export HOME=/home/oracle
$ORACLE_HOME/bin/rman target / @$HOME/ClearArch.rcv log=$HOME/ClearArch.log
append
```

Which invoke the recovery manager script: \$HOME/ClearArch.rcv

```
delete noprompt archivelog all;
```

The RMAN delete command is scheduled as a database job: CLEARARCH, which runs every 15 minutes.

```
begin
```

```

dbms_scheduler.create_job(
  job_name=>'ClearArch',
  job_type=>'EXECUTABLE',
  job_action=>'/home/oracle/ClearArch.sh',
  repeat_interval=>'FREQ=MINUTELY; interval=15; bysecond=0;',
  enabled=>TRUE
);
end;
/

```

Monitor the log files ClearArch.log to verify archive files are successfully deleted.

## Fast Recovery Area

Configure the Fast Recovery Area (FRA) to support extensive workload as shown below.

```

EDW> alter system set db_recovery_file_dest_size=100G scope=both;

System altered.

EDW>

```

## Automatic Database Files Extension

Ensure the database files are automatically extended to support extensive workload as shown below.

```

EDW> select default_tablespace
2   from dba_users
3  where username='OSM_REPAPI';

DEFAULT TABLESPACE
-----
OSM_DAT

BRM> alter database datafile
2   '/u01/app/oracle/oradata/EDW/osm_dat01.dbf' autoextend on;

Database altered.

EDW>

```

## Oracle GoldenGate Pre-Requisites

There are database pre-requisite for Oracle GoldenGate 12c. These pre-requisites are essential to operate Oracle GoldenGate successfully. The database administrator is highly recommended, but not mandatory.

- Enable Archive log Mode on the source
- Enable database supplemental logging on the source
- Create GoldenGate database administrator on source, downstream and target.

## Database Archive Mode

The source database must operate in archive log mode. Query the database archive mode and enable it, if disabled.

```
[oracle@linux4acdr ~]$ . oraenv
ORACLE_SID = [S1C2] ?
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@linux4acdr ~]$ sqlplus /nolog

SQL*Plus: Release 12.2.0.1.0 Production on Sat Jan 6 13:18:26 2018

Copyright (c) 1982, 2016, Oracle. All rights reserved.

SQL> conn / as sysdba
Connected.
SQL> select log_mode
       2 from v$database;

LOG MODE
-----
NOARCHIVELOG

SQL>
```

If the query returns NOARCHIVELOG, enable the archive log mode as shown next.

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount
ORACLE instance started.

Total System Global Area          3154116608 bytes
Fixed Size                        8625656 bytes
Variable Size                     2499805704 bytes
Database Buffers                  637534208 bytes
Redo Buffers                      8151040 bytes
Database mounted.
SQL> alter database archivelog;

Database altered.

SQL> select log_mode
       2 from v$database;

LOG MODE
```

```
-----  
ARCHIVELOG  
  
SQL>
```

## Database Supplemental Logging

The database supplemental logging must be enabled. This is required by Oracle GoldenGate to perform database transactions logs change data capture. Query the current supplemental state and enable, if disabled. Enabling beyond the minimum supplement logging generates excessive transaction log records.

```
SQL> select supplemental_log_data_min  
2 from v$database;  
  
SUPPLEME  
-----  
NO  
  
SQL> alter database add supplemental log data;  
  
Database altered.  
  
SQL> select supplemental_log_data_min  
2 from v$database;  
  
SUPPLEME  
-----  
YES  
  
SQL>
```

Now, open the database and perform switch log file. This enables the database supplemental logging.

```
SQL> alter database open;  
  
Database altered.  
  
SQL> alter system switch logfile;  
  
System altered.  
  
SQL>
```

Configure the database to support Oracle GoldenGate integrated extract (IE) deployed locally or on a downstream. Begin by allocating System Global Area (SGA) boundary, which allows enabling Automatic Shared Memory Management.

```
SQL> alter system set memory_max_target=3000M scope=spfile;
```

```

System altered.

SQL> alter system set memory_target=3000M scope=both;

System altered.

SQL>

```

The minimum size for the streams pool size is 750M; re-size the streams pool size required by the integrated extract based on the volume of the transactions stream sent from the source database. The following ALTER command allocates 1GB for streams pool size memory area.

```

SQL> alter system set streams_pool_size=1000M scope=spfile;

System altered.

SQL>

```

Activate the memory parameters by bouncing the database; this is because MEMORY\_MAX\_TARGET is static parameter.

```

SQL> startup force
ORACLE instance started.

Total System Global Area          3154116608 bytes
Fixed Size                        8625656 bytes
Variable Size                     2499805704 bytes
Database Buffers                  637534208 bytes
Redo Buffers                      8151040 bytes
Database mounted.
Database opened.

SQL>

```

Verify the parameters settings

```

SQL> show parameters memory_max_target

NAME                                TYPE                                VALUE
-----                                -----                                -----
memory_max_target                   big integer                         3008M
SQL> show parameters memory_target

NAME                                TYPE                                VALUE
-----                                -----                                -----
memory target                       big integer                         3008M
SQL> show parameters streams_pool_size

NAME                                TYPE                                VALUE
-----                                -----                                -----
streams_pool_size                   big integer                         1008M
SQL>

```

## Create GoldenGate Oracle Database User

Create Oracle GoldenGate, grant the privileges and test connection to the source database instance using the connect string. Because this a test instance, the DBA privilege is granted. In a production this must be avoided and explicitly grant the privileges required for the capture.

```
SQL> create tablespace ogg_dat;

Tablespace created.

SQL> grant dba to ogg_admin identified by oracle;

Grant succeeded.

SQL> alter user ogg_admin default tablespace ogg_dat;

User altered.

SQL>
```

Verify create database session

```
SQL> conn ogg_admin/oracle@<connect string>

Connected.

SQL>
```

In the event of unable to connect, the connection (TNS error) must be resolved before continuing further with Oracle GoldenGate deployment.

## Install Oracle GoldenGate 12c

There are two methods for installing Oracle GoldenGate 12c.

- Oracle GoldenGate 12c GUI Installer
- Oracle GoldenGate 12c Silent Installer

### Oracle GoldenGate 12c GUI Installer

Starting the installer program (runInstaller) without parameters lunches the Oracle GoldenGate 12c GUI Installer as shown in Figure X. Select the database version, click next and follow the step to complete the installation.

```
[oracle@linux4acdr Disk1]$ pwd
/stage/ogg12c/fbo_ggs_Linux_x64_shiphome/Disk1
[oracle@linux4acdr Disk1]$ ./runInstaller
```



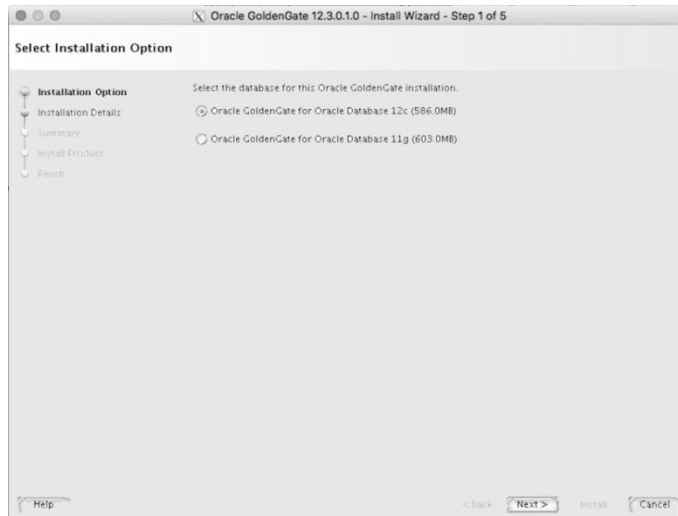


Figure x: Oracle GoldenGate 12c GUI Installer

## Oracle GoldenGate 12c Silent Installer

Silent method requires the preparation of the response file for the specific Oracle GoldenGate 12c version. This shows a response for Oracle GoldenGate 12c 12.3 (line 1), start the manager process (line 4) at port 7840 (line 5).

```
[oracle@linux4acdr ~]$ cd
[oracle@linux4acdr ~]$ cat oggcore1.rsp

oracle.install.responseFileVersion=rsfmt_ogginstall_response_schema_v12_3_0
INSTALL_OPTION=ORA12c
SOFTWARE_LOCATION=/u01/app/oracle/product/12.3.0/gghome_1
START_MANAGER=true
MANAGER_PORT=7840
DATABASE_LOCATION=/u01/app/oracle/product/12.2.0/dbhome_1
INVENTORY_LOCATION=/u01/app/oraInventory
UNIX_GROUP_NAME=oinstall
[oracle@linux4acdr ~]$
```

Make sure the DATABASE\_LOCATION is matching the ORACLE\_HOME.

```
[oracle@linux4acdr ~]$ echo $ORACLE_HOME
/u01/app/oracle/product/12.2.0/dbhome_1
[oracle@linux4acdr ~]$
```

Run the installer on silent mode

```
[oracle@linux4acdr ~]$ runInstaller -silent -responsefile
/home/oracle/oggcore1.rsp -nowelcome -showProgress -force -waitforcompletion
```

DISPLAY turned off Successfully

The Silent install option...

```
oracle.install.responseFileVersion=rsfmt_ogginstall_response_schema_v12_3_0
INSTALL_OPTION=ORA12c
SOFTWARE_LOCATION=/u01/app/oracle/product/12.3.0/gghome_1
START_MANAGER=true
MANAGER_PORT=7840
DATABASE_LOCATION=/u01/app/oracle/product/12.2.0/dbhome_1
INVENTORY_LOCATION=/u01/app/oraInventory
UNIX_GROUP_NAME=oinstall
```

```
/stage/ogg1230/fbo_ggs_Linux_x64_shiphome/Disk1/runInstaller -silent -
responsefile /home/oracle/oggcore1.rsp -nowelcome -showProgress -force -
waitforcompletion
```

Press Enter to Begin Oracle GoldenGate Silent Install.....  
Do not interrupt the installation while in-progress.....  
It will take approximately 5-10 minutes. Press Enter to begin.....  
Starting Oracle Universal Installer...

```
Checking Temp space: must be greater than 120 MB.   Actual 50219 MB
Passed
Checking swap space: must be greater than 150 MB.   Actual 9991 MB   Passed
Preparing to launch Oracle Universal Installer from /tmp/OraInstall2018-01-
06_02-54-51PM. Please wait ...You can find the log of this install session
at:
/u01/app/oraInventory/logs/installActions2018-01-06_02-54-51PM.log
```

```
Prepare in progress.
..... 9% Done.
```

Prepare successful.

```
Copy files in progress.
..... 31% Done.
..... 44% Done.
..... 65% Done.
..... 70% Done.
..... 89% Done.
.....
Copy files successful.
```

Link binaries in progress.

Link binaries successful.

Setup files in progress.

Setup files successful.

Setup Inventory in progress.

Setup Inventory successful.

Finish Setup successful.  
The installation of Oracle GoldenGate Core was successful.  
Please check '/u01/app/oraInventory/logs/silentInstall2018-01-06\_02-54-51PM.log' for more details.

Running Configuration Assistants in progress.

Configuring and starting manager... in progress.  
..... 100% Done.

Configuring and starting manager... successful.  
Successfully Setup Software.  
[oracle@linux4acdr ~]\$

Start GGSCI and verify the manager process is running on port 7840.

```
[oracle@linux4acdr gghome_1]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.3.0.1.0 OGGCORE 12.3.0.1.0 PLATFORMS 170721.0154 FBO
Linux, x64, 64bit (optimized), Oracle 12c on Jul 21 2017 23:31:13
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2017, Oracle and/or its affiliates. All rights reserved.

GGSCI (linux4.oracle.com) 1> info mgr

Manager is running (IP port linux4.oracle.com.7840, Process ID 14030).

GGSCI (linux4.oracle.com) 2>
```

Verify the install directories are created. Note the Oracle GoldenGate 12c root directory.

```
GGSCI (linux4.oracle.com) 2> show

Parameter settings:

SET DEBUG      OFF

Current directory: /u01/app/oracle/product/12.3.0/gghome_1

Editor: vi

Reports (.rpt)           /u01/app/oracle/product/12.3.0/gghome_1/dirprt
Parameters (.prm)        /u01/app/oracle/product/12.3.0/gghome_1/dirprm
Replicat Checkpoints (.cpr) /u01/app/oracle/product/12.3.0/gghome_1/dirchk
Extract Checkpoints (.cpe) /u01/app/oracle/product/12.3.0/gghome_1/dirchk
Process Status (.pcs)     /u01/app/oracle/product/12.3.0/gghome_1/dirpcs
SQL Scripts (.sql)        /u01/app/oracle/product/12.3.0/gghome_1/dirsq
Database Definitions (.def) /u01/app/oracle/product/12.3.0/gghome_1/dirdef
Dump files (.dmp)         /u01/app/oracle/product/12.3.0/gghome_1/dirdmp
```

```
Masterkey wallet files (.wlt) /u01/app/oracle/product/12.3.0/gghome_1/dirwlt
Credential store files (.crd) /u01/app/oracle/product/12.3.0/gghome_1/dircrd
```

```
GGSCI (linux4.oracle.com) 3> exit
[oracle@linux4acdr gghome_1]$
```

## Creating Workload Simulator Database API

The database schema for the workload simulator owns the database API to run the workload simulator; it's configured as shown next.

|                             |            |
|-----------------------------|------------|
| <b>Database Schema</b>      | OSM_REPAPI |
| <b>Database Password</b>    | oracle     |
| <b>Default Tablespace</b>   | OSM_DAT    |
| <b>Temporary Tablespace</b> | TEMP02     |

*The following steps are executed on the source database.*

### Create the Tablespace

Create the tablespace (OSM\_DAT) for the workload simulator, the tablespace's database file is auto extended to sustain long running transactions.

```
SQL> create tablespace OSM_DAT
2 datafile '/u01/app/oracle/oradata/OLTP/OSM_DAT01.dbf'
3 size 500M
4 autoextend on;
```

Tablespace created.

```
SQL>
```

### Create the Database Schema

Create the workload simulator schema, set the default permanent tablespace and the default temporary tablespace.

```
SQL> grant dba to osm_repapi identified by oracle;
```

Grant succeeded.

```
SQL> alter user osm_repapi default tablespace OSM_DAT
2 quota unlimited on OSM_DAT;
```

User altered.

```
SQL>
```

## Grant the System Privileges

Grant the system privilege as shown next. This is a test environment; these must be avoided on production system.

```
SQL> grant execute on dbms_lock to osm_repapi;
```

```
Grant succeeded.
```

```
SQL>
```

## Create the Import Directory

Create the database import directory and grant the privilege to OSM\_REPAPI.

```
SQL> create directory impdir as '/stage/dmp';
```

```
Directory created.
```

```
SQL> grant all on directory impdir to osm_repapi;
```

```
Grant succeeded.
```

```
SQL>
```

*Repeat the same steps on the target (EDW) database.*

## Importing the Workload Simulator Database Schema on OLTP

Import the table's structure and data.

```
[oracle@oltp01 dmp]$ pwd
/stage/dmp
[oracle@oltp01 dmp]$ ls -al *.dmp
-rw-r----- 1 root root 1743392768 Apr 13 18:06 OLTP02.dmp

[oracle@oltp01 dmp]$ impdp osm_repapi/oracle directory=impdir
dumpfile=OLTP02.dmp
```

```
Import: Release 11.2.0.4.0 - Production on Fri Apr 13 18:07:42 2018
```

```
Copyright (c) 1982, 2011, Oracle and/or its affiliates. All rights
reserved.
```

```
Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 -
64bit Production
```

```

With the Partitioning, OLAP, Data Mining and Real Application Testing
options
Master table "OSM_REPAPI"."SYS_IMPORT_FULL_01" successfully loaded/unloaded
Starting "OSM_REPAPI"."SYS_IMPORT_FULL_01":  osm_repapi/*****
directory=impdir dumpfile=OLTP02.dmp
Processing object type SCHEMA_EXPORT/USER
ORA-31684: Object type USER:"OSM_REPAPI" 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/TABLESPACE_QUOTA
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 "OSM_REPAPI"."CONTACTS"                883.8 MB 17213477 rows
. . imported "OSM_REPAPI"."CALLS"                    477.6 MB 6229053 rows
. . imported "OSM_REPAPI"."CALL_DETAILS"              281.6 MB 6229053 rows
. . imported "OSM_REPAPI"."POLICIES"                  14.72 MB 266497 rows
. . imported "OSM_REPAPI"."CUSTOMERS"                 688.7 KB 6249 rows
. . imported "OSM_REPAPI"."MOBILE_LINES"              1.315 MB 34337 rows
. . imported "OSM_REPAPI"."SIMS"                      1.643 MB 34337 rows
. . imported "OSM_REPAPI"."CLAIMS"                   228.8 KB 7138 rows
. . imported "OSM_REPAPI"."CITIES"                    6.328 KB 5 rows
. . imported "OSM_REPAPI"."CLAIM_STATUS"              5.453 KB 2 rows
. . imported "OSM_REPAPI"."COUNTRIES"                 5.898 KB 3 rows
. . imported "OSM_REPAPI"."DISCOUNT_TYPES"           5.898 KB 3 rows
. . imported "OSM_REPAPI"."EDU_LEVELS"                5.523 KB 5 rows
. . imported "OSM_REPAPI"."PACKAGE_TYPES"             7.406 KB 4 rows
. . imported "OSM_REPAPI"."POLICY_TYPES"              5.523 KB 4 rows
. . imported "OSM_REPAPI"."PROFESSIONS"               5.546 KB 7 rows
. . imported "OSM_REPAPI"."RANDOM_DOMAIN"              5.070 KB 5 rows
. . imported "OSM_REPAPI"."RANDOM_FIRST"               5.132 KB 11 rows
. . imported "OSM_REPAPI"."RANDOM_LAST"                5.164 KB 11 rows
. . imported "OSM_REPAPI"."RANDOM_MIDDLE"              5.054 KB 6 rows
. . imported "OSM_REPAPI"."REGIONS"                   5.507 KB 5 rows
. . imported "OSM_REPAPI"."OSM_DOCUMENTS"              0 KB 0 rows
Processing object type SCHEMA_EXPORT/PROCEDURE/PROCEDURE
Processing object type SCHEMA_EXPORT/PROCEDURE/ALTER_PROCEDURE
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
Job "OSM_REPAPI"."SYS_IMPORT_FULL_01" completed with 1 error(s) at Fri Apr
13 18:11:26 2018 elapsed 0 00:03:43

[oracle@oltp01 dmp]$

```

## Create OGG\_ADMIN User on EDW

The Oracle GoldenGate 12c administrator should be created before starting the import. Otherwise, some errors returned during the import session.

```

SQL> grant dba to ogg_admin identified by oracle;

Grant succeeded.

SQL>

```

# Importing the Workload Simulator Database Schema on EDW

Import the table's structure and data. Note there is a tablespace remap option, from OSM\_DAT\_16K to OSM\_DAT.

```
[oracle@edw01 dmp]$ impdp osm_repapi/oracle directory=impdir
dumpfile=EDW01.dmp remap_tablespace=OSM_DAT_16K:OSM_DAT
```

Import: Release 12.2.0.1.0 - Production on Fri Apr 13 21:50:09 2018

Copyright (c) 1982, 2017, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production

```
Master table "OSM_REPAPI"."SYS_IMPORT_FULL_01" successfully loaded/unloaded
Starting "OSM_REPAPI"."SYS_IMPORT_FULL_01": osm_repapi/*****
directory=impdir dumpfile=EDW01.dmp remap_tablespace=OSM_DAT_16K:OSM_DAT
Processing object type SCHEMA_EXPORT/USER
ORA-31684: Object type USER:"OSM_REPAPI" 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/TABLESPACE_QUOTA
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
Processing object type SCHEMA_EXPORT/TABLE/TABLE
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
. . imported "OSM_REPAPI"."CALL_DETAILS":"P2018" 1.139 GB 4740388 rows
. . imported "OSM_REPAPI"."CONTACTS" 883.8 MB 17213477 rows
. . imported "OSM_REPAPI"."COUNTCONTACTMV" 377.4 MB 17213477 rows
. . imported "OSM_REPAPI"."CALLS" 477.6 MB 6229053 rows
. . imported "OSM_REPAPI"."CALL_DETAILS":"P2017" 48.21 MB 198884 rows
. . imported "OSM_REPAPI"."CALL_DETAILS":"P2016" 48.18 MB 198760 rows
. . imported "OSM_REPAPI"."CALL_DETAILS":"P2013" 48.22 MB 198946 rows
. . imported "OSM_REPAPI"."CALL_DETAILS":"P2011" 48.14 MB 198639 rows
. . imported "OSM_REPAPI"."CALL_DETAILS":"P2015" 48.07 MB 198330 rows
. . imported "OSM_REPAPI"."CALL_DETAILS":"P2012" 48.04 MB 198193 rows
. . imported "OSM_REPAPI"."CALL_DETAILS":"P2014" 47.91 MB 197671 rows
. . imported "OSM_REPAPI"."CALL_DETAILS":"P2010" 24.05 MB 99242 rows
. . imported "OSM_REPAPI"."POLICIES" 13.03 MB 235882 rows
. . imported "OSM_REPAPI"."SIMS" 1.643 MB 34337 rows
. . imported "OSM_REPAPI"."CUSTOMERS" 689.0 KB 6249 rows
. . imported "OSM_REPAPI"."MOBILE_LINES" 1.315 MB 34337 rows
. . imported "OSM_REPAPI"."CALL_DETAILS_BY_CUST_CNTRY" 187.6 KB 6246 rows
. . imported "OSM_REPAPI"."CLAIMS" 112.6 KB 3400 rows
. . imported "OSM_REPAPI"."PACKAGE_TYPES" 7.570 KB 4 rows
. . imported "OSM_REPAPI"."DISCOUNT_TYPES" 5.992 KB 3 rows
. . imported "OSM_REPAPI"."COUNTRIES" 5.992 KB 3 rows
. . imported "OSM_REPAPI"."PROFESSIONS" 5.617 KB 7 rows
. . imported "OSM_REPAPI"."POLICY_TYPES" 5.593 KB 4 rows
. . imported "OSM_REPAPI"."EDU_LEVELS" 5.593 KB 5 rows
. . imported "OSM_REPAPI"."REGIONS" 5.578 KB 5 rows
. . imported "OSM_REPAPI"."MYCALL":"P2017" 5.523 KB 2 rows
. . imported "OSM_REPAPI"."CLAIM_STATUS" 5.523 KB 2 rows
. . imported "OSM_REPAPI"."RANDOM_LAST" 5.210 KB 11 rows
. . imported "OSM_REPAPI"."RANDOM_FIRST" 5.179 KB 11 rows
. . imported "OSM_REPAPI"."RANDOM_DOMAIN" 5.117 KB 5 rows
```

```

. . imported "OSM_REPAPI"."RANDOM_MIDDLE"          5.101 KB      6 rows
. . imported "OSM_REPAPI"."CALL_DETAILS":"P2019"    0 KB         0 rows
. . imported "OSM_REPAPI"."CITIES"                 0 KB         0 rows
. . imported "OSM_REPAPI"."MLOG$_CALL_DETAILS"      0 KB         0 rows
. . imported "OSM_REPAPI"."MLOG$_CONTACTS"          0 KB         0 rows
. . imported "OSM_REPAPI"."MYCALL":"P2015"          0 KB         0 rows
. . imported "OSM_REPAPI"."MYCALL":"P2016"          0 KB         0 rows
. . imported "OSM_REPAPI"."OSM_DOCUMENTS"           0 KB         0 rows
. . imported "OSM_REPAPI"."TRANS_LOG"               0 KB         0 rows
Processing object type SCHEMA_EXPORT/TABLE/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type SCHEMA_EXPORT/TABLE/COMMENT
Processing object type SCHEMA_EXPORT/PROCEDURE/PROCEDURE
Processing object type
SCHEMA_EXPORT/PROCEDURE/GRANT/OWNER_GRANT/OBJECT_GRANT
Processing object type SCHEMA_EXPORT/PROCEDURE/ALTER_PROCEDURE
Processing object type SCHEMA_EXPORT/TABLE/INDEX/INDEX
Processing object type SCHEMA_EXPORT/TABLE/INDEX/FUNCTIONAL_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/INDEX/STATISTICS/FUNCTIONAL_INDEX/INDEX_STATISTICS
Processing object type SCHEMA_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
Processing object type SCHEMA_EXPORT/STATISTICS/MARKER
Processing object type SCHEMA_EXPORT/TABLE/POST_TABLE_ACTION
Processing object type SCHEMA_EXPORT/MATERIALIZED_VIEW
Processing object type SCHEMA_EXPORT/TABLE/MATERIALIZED_VIEW_LOG

ORA-39082: Object type PROCEDURE:"OSM_REPAPI"."TRANS_LOAD" created with
compilation warnings

ORA-39082: Object type PROCEDURE:"OSM_REPAPI"."CALL_TRANS_LOAD" created with
compilation warnings

Job "OSM_REPAPI"."SYS_IMPORT_FULL_01" completed with 4 error(s) at Fri Apr
13 21:53:58 2018 elapsed 0 00:03:48

[oracle@edw01 dmp]$

```

## Drop Invalid Objects

These objects are not needed on the source (EDW) database, as the workload simulator runs from the source only.

```

SQL> drop procedure "OSM_REPAPI"."TRANS_LOAD";

Procedure dropped.

SQL> drop procedure "OSM_REPAPI"."CALL_TRANS_LOAD";

Procedure dropped.

SQL>

```



# Summary

The completion of the setup sections is essential to perform the workshop successfully. Depending on the resources available on the host system, the guest virtual machines resources allocated. The main resource management is the memory allocation among the guest operating system.

The list of configurations to prepare the workshop environment:

- Install Oracle VirtualBox
- Import the virtual machines
- Virtual machines memory settings
- Virtual machines network settings
- Power on the source machine
- Configure Domain Names System (DNS)
- Login using PuTTY terminals
- Configure Oracle SQLNET configuration files
- Deploy automated management

Depending on the physical memory available on the host operating system, use the following suggested settings.

| Host Memory Size | Source | Downstream | Target (EDW) |
|------------------|--------|------------|--------------|
| 8GB              | 2GB    | 2GB        | 3GB          |
| 16GB             | 3GB    | 3GB        | 8GB          |
| 32GB             | 6GB    | 8GB        | 16GB         |

The purpose of the automated management is to ensure the available resources are managed effectively and the database continues when running the workload on the source database. The most essential is monitoring is the following growth:

- The fast recovery area (FRA) growth
- The database file growth

FRA growth is optimized by automatically deleting the archive log files generated and sent to the FRA when running the workload on high-extreme levels. *Again, do not do this practice on production environment; rather configure RMAN archive log retention.*

Enabling the database file automatic expansion prevent from unable to allocate extents during the run of the workload simulator. The database file for the application schema grows until reaches the maximum limit of the disk space available.

# Oracle Software Download Policy

The Oracle software download from <http://edelivery.oracle.com> is strictly subject to Oracle Software download terms and restrictions. You must read and accept the terms and conditions before download the software shown in Figure X.

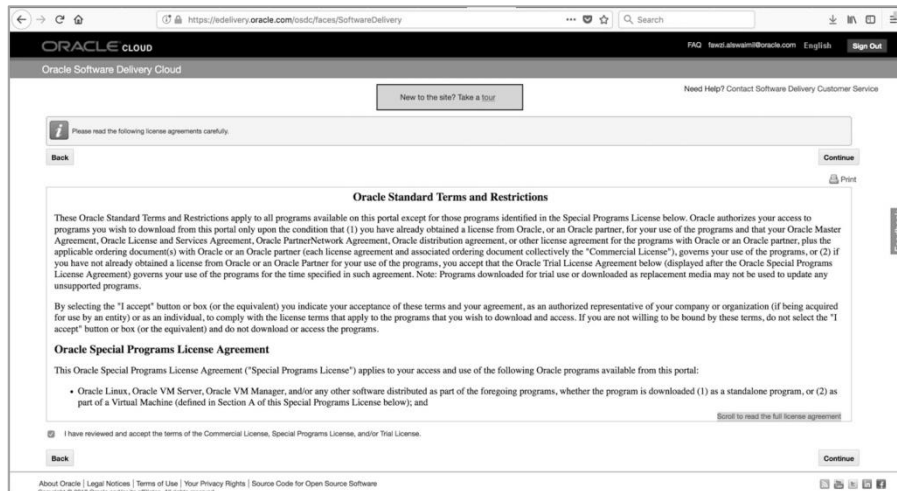


Figure x: Oracle Software Download Standard Terms and Conditions

## Oracle Software Installed

- Oracle Database 11g EE, Oracle Database 12c EE
- Oracle GoldenGate 12c
- Oracle Data Visualization Desktop
- Oracle SQL Developer
- Oracle VirtualBox
- Oracle Linux

## Workshop Virtual Machines Downloads

The virtual machines images have a restricted distribution. They are subject to Oracle Standard Terms and Conditions.

They can be only requested from the author by sending an email to:  
[fawzi.alswaimil@oracle.com](mailto:fawzi.alswaimil@oracle.com)

Other software is subject to their owner's terms and conditions.

# Database Design and Workload Simulation

## Overview

The purpose of the workload simulator (or simulator in short) is to generate transactions volumes sustainable for the duration of the project phases. The settings of the workload simulator is adjustable to match the real system if resources allows, or reduced proportionally in case it's running on limited resource testing environment. The workload simulator API developed using PL/SQL; hence it's stored in the database and accessible to users by granting the execute privilege.

This workload simulator is built on top of the data model shown in Figure 1. The workload simulator designed to simulate telecom mobile operators, this enables generating massive transactions volumes suitable for demonstrating an Enterprise Data Warehouse (EDW) operations.

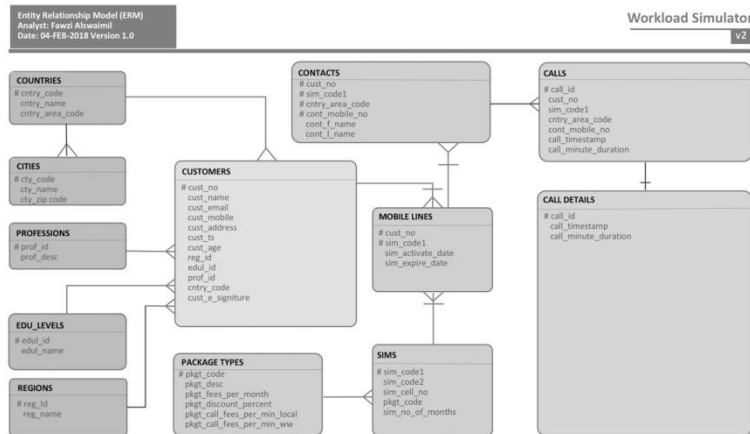


Figure 1: Workload Simulator Data Model

Figure 1 depicts the data model of the source database; the target database (EDW) has a dissimilar data model, which is explained in later sections.

# Workload levels, an Introduction

There are four workload levels; each designed to suite the minimum available system resources. Running a workload beyond the available resources of processors and disk space, may cause system slowness or crash. Always start with low workload, and uplift the workload as needed. This workload simulator provides the following levels:

- Low-level workload
- Medium-level Workload
- High-level Workload
- Extreme-level Workload

In order to support the non-default workload transactions volumes, ensure the transactions logs archived log files are periodically deleted using the Oracle Recovery Manager (RMAN). The following demonstrate using the DBMS\_SCHEDULER to delete the archive log file every 15 minutes.

```
begin
  dbms_scheduler.create_job(
    job_name=>'ClearArch',
    job_type=>'EXECUTABLE',
    job_action=>'/home/oracle/ClearArch.sh',
    repeat_interval=>'FREQ=MINUTELY; interval=15; bysecond=0;',
    enabled=>TRUE
  );
end;
```

Also, there must enough online redo log groups to support the log switch event for high volumes transactions. Figure 2 shows the configuration of the online redo log groups and online standby redo log groups for the Enterprise Data Warehouse.

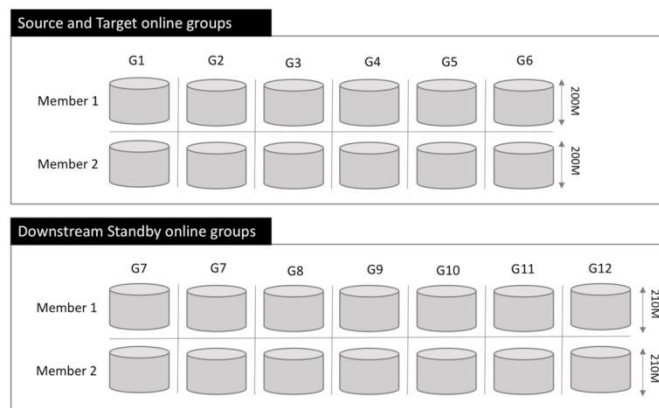


Figure 2: Online Redo Log Group Configuration

## Low-level Workload

The low-level workload simulator is the default. This workload session runs with the minimum resource required. The gap between transactions is 0.1 seconds. It generates 100,000 rows and last for approximately 2.8 hours.

## Medium Workload

The medium-level workload simulator is non-default. This workload session runs with above the minimum resource required. The gap between transactions is 0.01 seconds. It generates 1,000,000 rows and last for approximately 2.8 hours.

## High Workload

The high-level workload simulator is non-default. This workload session runs with high resource requirement. The gap between transactions is 0.001 seconds. It generates 100,000,000 rows and last for approximately 27.8 hours.

## Extreme Workload

The extreme-level workload simulator is non-default. This workload session runs with very high level resource requirement. Do not run the workload level unless you have a system with large number of processors. No gap between transactions, it generates 10,000,000,000 rows and last for approximately 11.5 hours.

*The above workloads have been benchmarked on 20-cores, Intel i7 Extreme processors system.*

## Architecture

The workload simulator is run on the source database only. Then it's Oracle GoldenGate 12c that move the transactions logs from the source, to downstream system for change data capture, then applied to the target – EDW as shown in Figure 3. Oracle GoldenGate 12c instances are running on the downstream and EDW systems.

The execution flow is from left to right. The workload simulator is started on the source database, by the call to `DBMS_GG_WL.CALL_TRANS_LOAD`. The source database is configured to ship the transactions logs on SYNC mode to the downstream system.

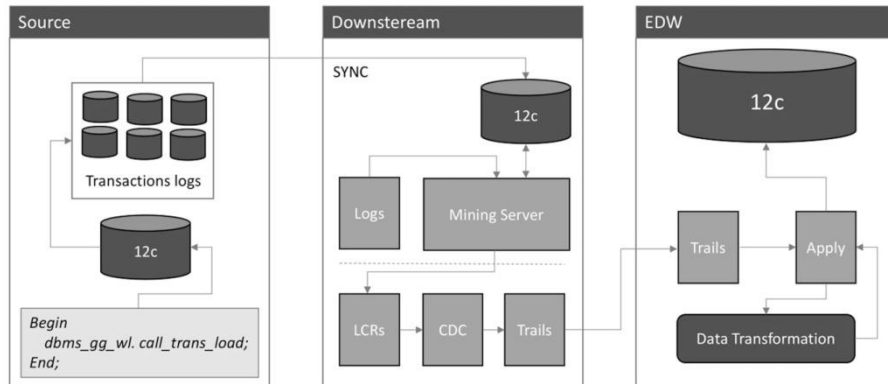


Figure 3: Workload Simulator Components

The downstream received the transactions logs; it uses a GoldenGate mining engine to produce the Logical Change Records (LCRs) which is used as input to the integrated capture extract to produce the extract trails. These trails are then sent to the remote system using the data pump process.

The remote trails received by the remote GoldenGate instance, and Replicat call-out to the data transformation procedure prepare the final data for the Replicat. The Replicat receive the output parameters and apply them to the database.

The data transformation is executed on the target database, because its PL/SQL stored procedure, call-out to Java and PL/SQL stored functions are always applicable to perform advanced and complex data transformation.

What is unique about the architecture is the in-transit compression and encryption is built-in. The data pump performs both functions when sending the trails.

## Workload Simulator API

The following is the workload simulator PL/SQL API; it uses cache in memory technique to generate random values stored as PL/SQL table, which are bind values to the DML statements to manipulate the database tables. The following are the list of tables manipulated or referenced by the workload simulator.

| Table Name    | Description                   |
|---------------|-------------------------------|
| CUSTOMERS     | Subscribers to mobile service |
| PACKAGE_TYPES | Package types                 |
| SIMS          | Sims mobile cards             |

|              |                                      |
|--------------|--------------------------------------|
| MOBILE_LINES | Mobile lines for subscribers         |
| CONTACTS     | Contact of a subscribers             |
| CALLS        | Calls made by a subscribers          |
| CALL_DETAILS | Call details made by the subscribers |

EDI> desc CUSTOMERS

| Name             | Null?    | Type           |
|------------------|----------|----------------|
| -----            | -----    | -----          |
| CUST_NO          | NOT NULL | NUMBER         |
| CUST_NAME        | NOT NULL | VARCHAR2 (128) |
| CUST_EMAIL       | NOT NULL | VARCHAR2 (128) |
| CUST_ADDRESS     |          | VARCHAR2 (128) |
| CUST_TS          |          | TIMESTAMP (6)  |
| CUST_AGE         |          | NUMBER         |
| REG_ID           |          | VARCHAR2 (8)   |
| EDUL_ID          |          | VARCHAR2 (8)   |
| PROF_ID          |          | VARCHAR2 (8)   |
| CNTRY_CODE       |          | VARCHAR2 (8)   |
| CUST_E_SIGNATURE |          | BLOB           |

EDI> desc PACKAGE\_TYPES

| Name                         | Null?    | Type           |
|------------------------------|----------|----------------|
| -----                        | -----    | -----          |
| PKGT_CODE                    | NOT NULL | VARCHAR2 (16)  |
| PKGT_DESC                    | NOT NULL | VARCHAR2 (128) |
| PKGT_FEES_PER_MONTH          |          | NUMBER (6, 2)  |
| PKGT_DISCOUNT_PERCENT        |          | NUMBER (2)     |
| PKGT_CALL_FEES_PER_MIN_LOCAL |          | NUMBER (4, 2)  |
| PKGT_CALL_FEES_PER_MIN_WW    |          | NUMBER (5, 2)  |

EDI> desc SIMS

| Name             | Null?    | Type          |
|------------------|----------|---------------|
| -----            | -----    | -----         |
| SIM_CODE1        | NOT NULL | VARCHAR2 (32) |
| SIM_CODE2        | NOT NULL | VARCHAR2 (32) |
| SIM_CELL_NO      | NOT NULL | VARCHAR2 (16) |
| PKGT_CODE        |          | VARCHAR2 (16) |
| SIM_NO_OF_MONTHS |          | NUMBER        |

EDI> desc MOBILE\_LINES

| Name              | Null?    | Type          |
|-------------------|----------|---------------|
| -----             | -----    | -----         |
| CUST_NO           | NOT NULL | NUMBER        |
| SIM_CODE1         | NOT NULL | VARCHAR2 (32) |
| SIM_ACTIVATE_DATE |          | DATE          |
| SIM_EXPIRE_DATE   |          | DATE          |

EDI> desc CONTACTS

| Name            | Null?    | Type          |
|-----------------|----------|---------------|
| -----           | -----    | -----         |
| CUST_NO         | NOT NULL | NUMBER        |
| SIM_CODE1       | NOT NULL | VARCHAR2 (32) |
| CNTRY_AREA_CODE | NOT NULL | VARCHAR2 (8)  |
| CONT_MOBILE_NO  | NOT NULL | VARCHAR2 (16) |
| CONT_F_NAME     |          | VARCHAR2 (64) |
| CONT_L_NAME     |          | VARCHAR2 (64) |

```

EDI> desc CALLS
Name                                     Null?      Type
-----
CALL_ID                                NOT NULL   VARCHAR2 (64)
CUST_NO                                NOT NULL   NUMBER
SIM_CODE1                              NOT NULL   VARCHAR2 (32)
CNTRY_AREA_CODE                        NOT NULL   VARCHAR2 (8)
CONT_MOBILE_NO                         NOT NULL   VARCHAR2 (16)
CALL_TIMESTAMP                         NOT NULL   TIMESTAMP (6)
CALL_MINUTE_DURATION                  NOT NULL   NUMBER (5,2)

EDI> desc CALL_DETAILS
Name                                     Null?      Type
-----
CALL_ID                                NOT NULL   VARCHAR2 (64)
CALL_TIMESTAMP                         NOT NULL   TIMESTAMP (6)
CALL_MINUTE_DURATION                  NOT NULL   NUMBER (5,2)
CUST_NO                                NOT NULL   NUMBER
CUST_NAME                             VARCHAR2 (128)
SIM_CODE1                              NOT NULL   VARCHAR2 (32)
SIM_EXPIRE_DATE                        NOT NULL   DATE
SIM_CELL_NO                            NOT NULL   VARCHAR2 (16)
DAILED_TO_AREA_CODE                   NOT NULL   VARCHAR2 (8)
DAILED_TO_NO                           NOT NULL   VARCHAR2 (16)
DAILED_TO_NAME                         NOT NULL   VARCHAR2 (128)
CALL_FEES                              NOT NULL   NUMBER (6,2)
REG_NAME                               NOT NULL   VARCHAR2 (128)
EDUL_NAME                              NOT NULL   VARCHAR2 (128)
PROF_DESC                              NOT NULL   VARCHAR2 (128)
CNTRY_NAME                             NOT NULL   VARCHAR2 (128)
PKGT_DESC                              NOT NULL   VARCHAR2 (128)
REP_GROUP                              NOT NULL   VARCHAR2 (8)
OS_USER                                NOT NULL   VARCHAR2 (32)
HOST_NAME                              NOT NULL   VARCHAR2 (32)
OPERATION_TYPE                         NOT NULL   VARCHAR2 (16)
OPERATION_TIMESTAMP                    NOT NULL   TIMESTAMP (6)

EDI>

```

The workload simulator package has three members. It's own by the schema OSM\_REPAPI.

```

SQL> desc dbms_gg_wl
PROCEDURE CALL_TRANS_LOAD
Argument Name                                     Type          In/Out  Default?
-----
N                                NUMBER        IN      DEFAULT
S                                NUMBER        IN      DEFAULT
PROCEDURE DISCONNECT_TRANS_LOAD
PROCEDURE TRANS_LOAD
Argument Name                                     Type          In/Out  Default?
-----
PLOAD_LEVEL                       NUMBER        IN      DEFAULT
PSTARTFROM                         NUMBER        IN      DEFAULT

SQL>

```



| Sub-Program           | Description                                                                         |
|-----------------------|-------------------------------------------------------------------------------------|
| Trans_Load            | Generates the random values and insert into tables according to the workload level. |
| Call_Trans_Load       | Controls the workload level by passing the parameters to Trans_Load sub-program     |
| Disconnect_Trans_Load | Terminates the workload sessions owned by the schema OSM_REPAPI.                    |

The PL/SQL code for the workload simulator:

```

create or replace procedure
  trans_load (pLoad Level in number default 10, pStartFrom in number default 1000000)
is
  vlast_name varchar2(32);
  vmiddle_name varchar2(32);
  vfirst_name varchar2(32);
  vlast_cnt number;
  vmiddle_cnt number;
  vfirst_cnt number;
  --
  vpt_code varchar2(8);
  vpt_cnt number;
  --
  vdt code varchar2(8);
  vdt cnt number;
  --
  vreg_id varchar2(8);
  vreg_cnt number;
  --
  vedul_id varchar2(8);
  vedul_cnt number;
  --
  vprof_id varchar2(8);
  vprof_cnt number;
  --
  vcntry_area_code varchar2(8);
  vcntry_area_cnt number;
  vcntry_code varchar2(8);
  vcntry cnt number;
  --
  vclm_status varchar2(8);
  vclm_cnt number;
  --
  vrd_name varchar2(32);
  vrd_cnt number;
  --
  vpkg_code varchar2(8);
  vpkg_cnt number;
  --
  vcont_mobile_no varchar2(16);
  vcont cnt number;
  --
  vsim code1 number;
  vline_cnt number;
  --
  vcty_code varchar2(8);
  vcty_cnt number;
  vcty_name varchar2(128);
  vcty_zip_code number;
  --

```

```

type pt_code_type$typ is table of varchar2(8) index by binary_integer;
vpt_code_type pt_code_type$typ;
--
type dt_code_type$typ is table of varchar2(8) index by binary_integer;
vdt_code_type dt_code_type$typ;
--
type rd name type$typ is table of varchar2(32) index by binary_integer;
vrd name type rd name type$typ;
--
type last_name_type$typ is table of varchar2(32) index by binary_integer;
vlast_name_type last_name_type$typ;
--
type middle_name_type$typ is table of varchar2(32) index by binary_integer;
vmiddle name type middle name type$typ;
--
type first_name_type$typ is table of varchar2(32) index by binary_integer;
vfirst_name_type first_name_type$typ;
--
type reg_id_type$typ is table of varchar2(8) index by binary_integer;
vreg_id_type reg_id_type$typ;
--
typeedul id type$typ is table of varchar2(8) index by binary_integer;
vedul_id_typeedul id type$typ;
--
type prof_id_type$typ is table of varchar2(8) index by binary_integer;
vprof id type prof id type$typ;
--
type cntry_area_code_type$typ is table of varchar2(8) index by binary_integer;
vcntry_area_code_type cntry_area_code_type$typ;
--
type cntry_code_type$typ is table of varchar2(8) index by binary_integer;
vcntry_code_type cntry_code_type$typ;
--
type cty code type$typ is table of varchar2(8) index by binary_integer;
vcty code type cty code type$typ;
type cty name type$typ is table of varchar2(8) index by binary_integer;
vcty_name_type cty_name_type$typ;
type cty_zip_code_type$typ is table of varchar2(8) index by binary_integer;
vcty_zip_code_type cty_zip_code_type$typ;
--
type clm status type$typ is table of varchar2(8) index by binary_integer;
vclm_status_type clm_status_type$typ;
--
type pkg_t_code_type$typ is table of varchar2(8) index by binary_integer;
vpkg_t_code_type pkg_t_code_type$typ;
--
type cont_mobile_no_type$typ is table of varchar2(16) index by binary_integer;
vcont mobile no type cont mobile no type$typ;
--
type sim1_code_type$typ is table of varchar2(32) index by binary_integer;
vsim1_code_type sim1_code_type$typ;
--
type sim1 duration type$typ is table of number index by binary_integer;
vsim1 duration type sim1 duration type$typ;
--
cursor c3 is select rf_name from random_first;
cursor c4 is select rm_name from random_middle;
cursor c5 is select rl_name from random_last;
cursor c6 is select rd_name from random_domain;
cursor c7 is select reg_id from regions;
cursor c8 is selectedul id from edu levels;
cursor c9 is select prof_id from professions;
cursor c11 is select cntry_code, cntry_area_code from countries;
cursor c16(pcntry_code varchar2) is select cty_code, cty_name, cty_zip_code from
cities where cntry_code=pcntry_code;
cursor c10 is select CLM STATUS from claim_status;
cursor c12 is select pkg_t code from package types;

```

```

    cursor c13(pcust_no number) is select CENTRY_AREA_CODE, cont_mobile_no from contacts
    where cust_no=pcust_no;
    cursor c14(pcust_no number) is select sim_codel from mobile_lines where
    cust_no=pcust_no;
    cursor c15(pcust_no number, psim_codel varchar2) is select cntry_area_code,
    cont_mobile_no from contacts where cust_no=pcust_no and sim_codel=psim_codel;
    --
    vcust seq number;
    vj number:=0;
    vk number:=0;
    vz number:=0;
    vsim1 varchar2(32);
    vsim2 varchar2(32);
    vsim1 duration number;
    vcall_id varchar2(32);
    vcont_mobile_no_cnt number;
    vcont_mobile_no2 varchar2(16);
    vname varchar2(128);
    vemail varchar2(128);
    vfname varchar2(64);
    vlname varchar2(64);
    vmname varchar2(1);
    vcountry_code varchar2(8);
    vage number;
    vind number;
    vcall timestamp timestamp;
    vcall duration number(5,2);
    --
begin
    select count(*) into vrd_cnt from random_domain;
    select count(*) into vlast_cnt from random_last;
    select count(*) into vmiddle_cnt from random_middle;
    select count(*) into vfirst_cnt from random_first;
    select count(*) into vreg_cnt from regions;
    select count(*) into vedul_cnt from edu_levels;
    select count(*) into vprof_cnt from professions;
    select count(*) into vcntry_cnt from countries;
    select count(*) into vpkgcnt_cnt from package_types;
    --
    vsim1 duration type(1):= 1;
    vsim1 duration type(2):= 3;
    vsim1_duration_type(3):= 6;
    vsim1_duration_type(4):= 12;
    --
    for i in 1..pLoad_Level loop
        --
        open c3;
        loop
            fetch c3 into vfirst_name;
            exit when c3%notfound;
            vfirst_name_type(c3%rowcount) := vfirst_name;
        end loop;
        close c3;
        --
        open c4;
        loop
            fetch c4 into vmiddle_name;
            exit when c4%notfound;
            vmiddle_name_type(c4%rowcount) := vmiddle_name;
        end loop;
        close c4;
        --
        open c5;
        loop
            fetch c5 into vlast_name;
            exit when c5%notfound;
            vlast_name_type(c5%rowcount) := vlast_name;

```

```

end loop;
close c5;
--
open c6;
loop
    fetch c6 into vrd_name;
    exit when c6%notfound;
    vrd_name_type(c6%rowcount) := vrd_name;
end loop;
close c6;
--
open c7;
loop
    fetch c7 into vreg_id;
    exit when c7%notfound;
    vreg_id_type(c7%rowcount) := vreg_id;
end loop;
close c7;
--
open c8;
loop
    fetch c8 into vedul_id;
    exit when c8%notfound;
    vedul_id_type(c8%rowcount) := vedul_id;
end loop;
close c8;
--
open c9;
loop
    fetch c9 into vprof_id;
    exit when c9%notfound;
    vprof_id_type(c9%rowcount) := vprof_id;
end loop;
close c9;
--
open c11;
loop
    fetch c11 into vcntry_code, vcntry_area_code;
    exit when c11%notfound;
    vcntry_code_type(c11%rowcount) := vcntry_code;
    vcntry_area_code_type(c11%rowcount) := vcntry_area_code;
end loop;
close c11;
--
open c12;
loop
    fetch c12 into vpkg_code;
    exit when c12%notfound;
    vpkg_code_type(c12%rowcount) := vpkg_code;
end loop;
close c12;
--
select count(*) into vcust_seq from customers;
if vcust_seq = 0 then
    vcust_seq := pStartFrom;
else
    select max(cust_no) into vcust_seq from customers;
    if pStartFrom > vcust_seq then
        vcust_seq := pStartFrom;
    else
        select max(nvl(cust_no, 0))+1 into vcust_seq from customers;
    end if;
end if;
--
vfname := vfirst_name_type(dbms_random.value(1, vfirst_cnt));
vmname := vmiddle_name_type(dbms_random.value(1, vmiddle_cnt));
vlname := vlname_type(dbms_random.value(1, vlname_cnt));

```

```

vname := vfname||' '||vmname||'. '||vlname;
vemail:= vfname||'. '||vlname||'@'||vrd_name_type(dbms_random.value(1,
vrd_cnt))||'.com';
vage := round(dbms_random.value(18, 90));
--
vcountry_code := vcntry_code_type(dbms_random.value(1, vcntry_cnt));
select count(*) into vcty cnt from cities where cntry code=vcountry code;
open cl6(vcountry code);
loop
    fetch cl6 into vcty_code, vcty_name, vcty_zip_code;
    exit when cl6%notfound;
    vcty_code_type(cl6%rowcount) := vcty_code;
    vcty_name_type(cl6%rowcount) := vcty_name;
    vcty_zip_code_type(cl6%rowcount) := vcty_zip code;
end loop;
close cl6;
--
insert into osm_repapi.customers values (vcust_seq,
vname,
vemail,
'PO Box '||round(dbms_random.value(100,
100000), 0)||' '||vcty name type(dbms_random.value(1, vcty cnt))||'
'||vcty_zip_code_type(dbms_random.value(1, vcty_cnt)),
localtimestamp,
vAge,
vreg id type(dbms_random.value(1,
vreg cnt)),
vedul id type(dbms_random.value(1,
vedul_cnt)),
vprof_id_type(dbms_random.value(1,
vprof_cnt)),
vcountry_code,
empty_blob());

commit;
--
-- Create SIM card
--
for i in 1..round(dbms_random.value(1, 10)) loop
    vsim1 := to_char(round(dbms_random.value(1000000000000, 9000000000000)));
    vsim2 := to_char(round(dbms_random.value(1000000000000, 9000000000000)));
    vsim1 duration := vsim1 duration type(dbms_random.value(1,
vsim1_duration_type.count));
    insert into sims values (vsim1,
vsim2,
'05'||to_char(round(dbms_random.value(1,
9)))||to_char(round(dbms_random.value(1, 9)))||to_char(round(dbms_random.value(1,
9)))||to_char(round(dbms_random.value(1, 9)))||to_char(round(dbms_random.value(1,
9)))||to_char(round(dbms_random.value(1, 9))),
vpkg_code_type(dbms_random.value(1, vpkg_cnt)),
vsim1_duration);

    commit;
    insert into mobile_lines values (vcust seq, vsim1, sysdate,
add months(sysdate,vsim1 duration));
    commit;
    --
    -- Create contacts range: 1 -1000
    --
    vcont_mobile_no_cnt number;
    --
    vcont_mobile_no2 varchar2(16);
    for i in 1..round(dbms_random.value(1, 1000)) loop
        vcont mobile no2 := '5'||lpad(round(dbms_random.value(1000000,
99999999)), 8, 0);
        select count(*) into vcont_mobile_no_cnt from contacts where
cust_no=vcust_seq and cont_mobile_no = vcont_mobile_no2;
        if vcont_mobile_no_cnt = 0 then
            insert into contacts values (vcust_seq,
vsim1,

```

```

vcntry_area_code_type(dbms_random.value(1, vcntry_cnt)),
                                vcont_mobile_no2,
vfirst_cnt)),
                                vfirst_name_type(dbms_random.value(1,
vlast cnt))),
                                vlast_name_type(dbms_random.value(1,
                                commit;
                                end if;
                                end loop;
                                end loop;
                                --
                                -- Make a call 1) Select Mobile Line - SIM
                                --                               2) Select contact
                                --                               3) Make the call
                                --
                                select count(*) into vline_cnt from mobile_lines where cust_no=vcust_seq;
                                open c14(vcust_seq);
                                loop
                                    fetch c14 into vsim_code1;
                                    exit when c14%notfound;
                                    vsim1 code type(c14%rowcount) := vsim code1;
                                end loop;
                                close c14;
                                vsim_code1 := vsim1_code_type(dbms_random.value(1, vline_cnt));
                                --
                                select count(*) into vcont cnt from contacts where cust no=vcust seq and
                                sim code1=vsim code1;
                                open c15(vcust_seq, vsim_code1);
                                loop
                                    fetch c15 into vCNTRY_AREA_CODE, vcont_mobile_no;
                                    exit when c15%notfound;
                                    vCNTRY_AREA_CODE_type(c15%rowcount) := vCNTRY_AREA_CODE;
                                    vcont mobile no type(c15%rowcount) := vcont mobile no;
                                end loop;
                                close c15;
                                --
                                -- make the call
                                for i in 1..round(dbms_random.value(1, 2000)) loop
                                    vind:=dbms_random.value(1, vcont cnt);
                                    vcall id := to char(round(dbms_random.value(1000000000000,
                                    9000000000000))||'-'||round(dbms_random.value(1000000000000, 9000000000000)));
                                    vj:= dbms_random.value(1, 100);
                                    vcall_duration := round(dbms_random.value(1, 180), 2);
                                    if vj > 50 then
                                        vcall_timestamp := to_timestamp(round(dbms_random.value(1,28))||'-'
                                        ||round(dbms_random.value(1,12))||'-'||round(dbms_random.value(2010,2018)), 'DD-MM-
                                        YYYY HH:MI:SS');
                                    else
                                        vcall_timestamp := localtimeStamp;
                                    end if;
                                    insert into calls values (vcall_id, vcust_seq, vsim_code1,
                                    vCNTRY AREA CODE type(vind), vcont mobile no type(vind), vcall_timestamp,
                                    vcall duration);
                                    commit;
                                    vj:= dbms_random.value(1, 100);
                                    if vj > 50 then
                                        insert into call_details values (vcall_id, vcall_timestamp,
                                    vcall_duration);
                                    else
                                        insert into call details values (vcall id, localtimeStamp,
                                    vcall_duration);
                                    end if;
                                    commit;
                                end loop;
                                end loop;
                                vj := 0;

```

```

vk := 0;
end;

```

## Working with the Workload Simulator

The PL/SQL API DBMS\_GG\_WL package is owned by the database schema OSM\_REPAPI. The following invoke the available workload levels.

| Argument | Name | Type   | In/Out | Default? |
|----------|------|--------|--------|----------|
| N        |      | NUMBER | IN     | DEFAULT  |
| S        |      | NUMBER | IN     | DEFAULT  |

Passing two parameters, N refers to the number of multiplied iterations, and S refers to the gap between two transactions in seconds.

### Starting Low-Level Workload

Connect to the database and run the workload simulator as shown next. Always start with low-level workload, and move to the next level as necessary.

```

S1C1>conn osm_repapi/oracle
Connected.
S1C1>exec dbms_gg_wl.call_trans_load(100000, 0.1);

```

### Starting Medium-Level Workload

Connect to the database and run the workload simulator as shown next.

```

S1C1>conn osm_repapi/oracle
Connected.
S1C1> exec dbms_gg_wl.call_trans_load(1000000, 0.01);

```

### Starting High-Level Workload

Connect to the database and run the workload simulator as shown next.

```

S1C1>conn osm_repapi/oracle
Connected.
S1C1> exec dbms_gg_wl.call_trans_load(10000000, 0.001);

```

## Starting Extreme-Level Workload

Connect to the database and run the workload simulator as shown next. Use this workload with cautious, as it requires enough system resources.

```
S1C1>conn osm_repapi/oracle
Connected.
S1C1> exec dbms_gg_wl.call_trans_load(1000000000, 0);
```

## Terminate the workload Simulator

Regardless of the workload level, terminate the workload database session by invoking the procedure `disconnect_trans_load`. Grant the following one-time privileges to OSM\_REPAPI.

```
SQL> grant select on sys.v_$session to osm_repapi;

Grant succeeded.

SQL> grant select on sys.v_$sqlarea to osm_repapi;

Grant succeeded.

SQL> grant alter system to osm_repapi;

Grant succeeded.

SQL>
```

From another terminal, Connect to the database and terminate the workload simulator as shown next.

```
S1C1>conn osm_repapi/oracle
Connected.
S1C1> exec dbms_gg_wl.disconnect_trans_load;
```

## Workload Simulator for MS Windows

Figure 4 shows the workload simulator for MS Windows. This is a general purpose workload simulator designed for Oracle GoldenGate 12c. The simulator allows more workload levels, and support of bi-directional data replication. The workload is available for any combination of DML statements: INSERT, UPDATE and DELETE.



The workload for MS Windows presents in-depth details regarding the source and target database SQL shared areas. The Replicat lag is determines and displays current Replicat's lag details.

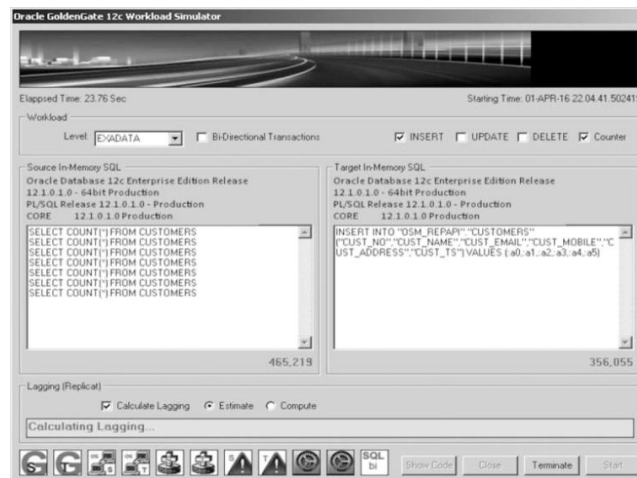


*Figure 4: Workload Simulator for MS Windows*

From within this workload simulator, source and target GoldenGate 12c instance and Oracle Database 12c utilities are accessible: GGSCI, Linux terminals, SQL\*Plus sessions, Log dump, and Oracle GoldenGate 12c log file.

## Summary

The workload simulator is a major component of the workshop, the purpose of the workload simulator is to generate the volume of transactions similar to the level of transactions volumes of the customer site, making the workshop as real-as-possible.

The workload simulator has four levels:

- Low-Level workloads
- Medium-Level workloads
- High-Level workloads
- Extreme-Level workloads

Unless you're aware of the system resources available, select the default workload level is low-level workload, higher workload may cause system performance related issues because of the contention of resources.

There are two interface of the workload simulator:

- SQL command interface
- MS Windows Interface

Both interfaces invoke the same API, DBMS\_GG\_WL. However; the MS Windows interface extends the DML operation to UPDATE and DELETE, while the SQL command interface is for INSERT only.

## Disclaimer

The workload simulator is strictly developed and used with this workshop only, do not migrate and run the workload on production environment or any other systems, as it may have negative impact on the system resources and performance.

### Warning!

This program may cause resource contention when executed on non-default mode

# Oracle Linux Basics for Microsoft Windows Users



## Overview

The workshop is built on Oracle Linux release 6 update 9. The purpose of this appendix is to ease the user's transition from Microsoft to Linux operating system. Linux running on Intel® chipset has dominated the information technology; nowadays non-stoppable systems operating 24/7 are commonly powered by Linux, which is a convincing reason the vast majority of the cloud service platforms and millions of commodities devices are operated by Linux distributions.

Beginner will notice Linux has similarities with MS Windows, both operating systems has an intuitive Graphical User Interface (GUI). Administering users, groups, directories, files and devices all can be performed from a GUI without remembering complex commands and attributes.

## Oracle Linux

Oracle Linux is an open source; it's compatible with Red Hat kernel. Currently, Oracle Linux distribution is version 7, which is backward compatible with previous releases, but with enhanced exiting features and added new functionalities. With the introduction of Linux Ksplice, the Linux Kernel is updatable online without the need to reboot the system; allowing systems to remain online during Linux patching.

Oracle Linux is unbounded by hardware vendors; it's the customer choice to select the hardware to host Oracle Linux, delivering the same level of reliability, security, performance and scalability found with enterprise systems. Oracle public cloud service is powered by Oracle Linux, the operating system of choice by customers, supporting all types of applications. Big data empowers Linux for elastic compute by adding Linux servers to the big data cluster, the primary reasons it's free of charge, and becoming the industry open source operating system. And, the business benefits of using Oracle Linux for Oracle Database for the single point of accountability, integrated support and advanced service are from one vendor, *Oracle Corporation*.

# Commonly Executed Linux Commands

The following are some of the most commonly executed commands by system administrators. Table 1 lists the commands and a brief description.

| Linux Command                     | Command Description                       |
|-----------------------------------|-------------------------------------------|
| \$mkdir -p /u01/app/oracle        | Create full path directory                |
| \$chown -R oracle:oinstall /u01   | Change the directory owner recursively    |
| \$ls-ltr                          | List files sorted by the creation date    |
| \$du .                            | Display disk usage                        |
| \$id                              | Display current user details              |
| \$chmod 777 .bash profile         | Change permission                         |
| \$scp * root@192.168.1.30:/tmp    | Remote secure copy                        |
| \$touch file1                     | Create a file                             |
| \$su                              | Switch to super user                      |
| \$rm file1                        | Remove file                               |
| #reboot                           | Reboot the operating system               |
| \$df -k                           | Report file system usage                  |
| \$ps -edf   grep pmon             | Display running processes                 |
| #sysctl -p                        | Re-activate the current kernel parameters |
| \$export DISPLAY=192.168.1.20:0.0 | Enable shell environment variable         |
| \$hostname                        | Display the host name                     |
| \$/sbin/ifconfig                  | Display network interface                 |
| \$cp f1 f2                        | Copy file                                 |
| \$cd                              | Change to home directory                  |
| \$vi                              | Start the editor                          |

Table 1: Linux Commands

## Linux Editor

Table 2 lists key-combination is for opening, modifying and saving files using the Linux Editor; vi.

| Keys           | Description                     |
|----------------|---------------------------------|
| vi             | Open the editor                 |
| i              | Insert                          |
| Save           | :wq!                            |
| :wq! f1.txt    | Create the file f1.txt          |
| :%s/DHDB/DXDB/ | Search and replace for a string |
| Small k        | Go up                           |
| Small j        | Go down                         |
| Small l        | Go right                        |
| Small h        | Go left                         |

|     |                                               |
|-----|-----------------------------------------------|
| \$  | End of Current Line                           |
| 0   | Beginning of current line                     |
| G   | Move to the end of the file                   |
| 1G  | Move to the beginning of the file             |
| 20G | Move to the 20 <sup>th</sup> Line of the file |

*Table 2: Linux Editor vi*

## Shell Scripting

The shell scripting (programming) is an essential asset for the database administrators (DBAs). The primary purpose is to auto-pilot the daily routine system administration tasks. A common database related example shell scripting is starting and stopping the database and listener up on system start up and shutdown as shown next.

```
#!/bin/sh
# description: Oracle auto start-stop script.
#
# Set ORACLE_HOME to be equivalent to the $ORACLE_HOME
# from which you wish to execute dbstart and dbshut;
#
# Set ORA_OWNER to the user id of the owner of the
# Oracle database in ORACLE_HOME.

ORA_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
ORA_OWNER=oracle

case "$1" in
'start')
    # Start the Oracle databases:
    # The following command assumes that the oracle login
    # will not prompt the user for any values
    # Remove "&" if you don't want startup as a background process.
    su - $ORA_OWNER -c "$ORA_HOME/bin/dbstart $ORA_HOME" &
    touch /var/lock/subsys/dbora
    ;;

'stop')
    # Stop the Oracle databases:
    # The following command assumes that the oracle login
    # will not prompt the user for any values
    su - $ORA_OWNER -c "$ORA_HOME/bin/dbshut $ORA_HOME" &
    rm -f /var/lock/subsys/dbora
    ;;
esac
```

Frequently automated tasks by the database administrator (DBA) include:

- Purging and cycling log and alert files
- Decision-based RMAN backup scripts (full and incremental based on week-day)
- Automation of database complex tasks such as database online upgrade

# Linux Kernel Shells

Oracle Linux supports several shells. They differ somehow when developing shell programming. Oracle Linux supports the following well-known shells: Bourne, C and Korn shells.

To find out the session current shell, type

```
#echo $SHELL
/bin/bash
#
```

To find out the current Kernel release, use the operating system command: `uname`

```
[root@linux4 init.d]# uname -a
Linux linux4.oracle.com 2.6.39-400.215.10.el5uek #1 SMP Tue Sep 9 22:51:46
PDT 2014 x86_64 x86_64 x86_64 GNU/Linux
[root@linux4 init.d]#
```

## Kernel Parameters

The Kernel parameters related to Oracle Database 12c are listed below. After editing the kernel parameter file `/etc/sysctl.conf`, re-activate the parameters settings without re-booting the operating system by executing the command, `sysctl -p`. Refer to the Oracle Database 12c installation guide for more details as the value for a Kernel parameter may change cross releases.

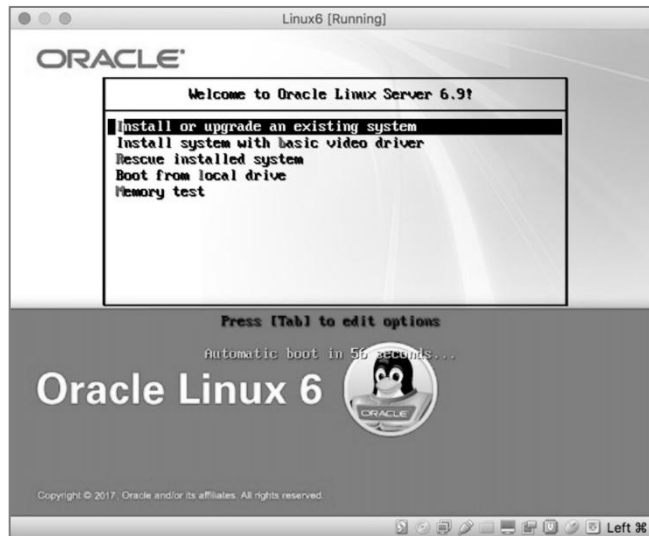
```
[root@linux4 etc]# sysctl -p
net.ipv4.ip_forward = 0
net.ipv4.conf.default.rp_filter = 2
net.ipv4.conf.default.accept_source_route = 0
kernel.core_uses_pid = 1
net.ipv4.tcp_syncookies = 1
kernel.msgmax = 65536
kernel.shmall = 4294967296
fs.file-max = 6815744
kernel.msgmni = 2878
kernel.msgmnb = 65536
kernel.sem = 250 32000 100 142
kernel.shmmni = 4096
kernel.shmmax = 4398046511104
kernel.sysrq = 1
net.core.rmem_default = 262144
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 1048576
fs.aio-max-nr = 3145728
net.ipv4.ip_local_port_range = 9000 65500
vm.min_free_kbytes = 51200
[root@linux4 etc]#
```

# Installing Oracle Linux

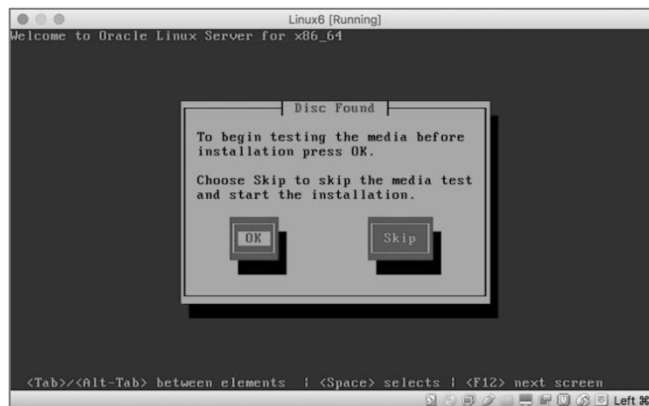
The following Linux installation is intended for research, test and educational purpose only. Production environment should allocate the file systems mount points based on the usage and resource availability, rather than using root “/” for all file systems.

The sequence of screenshots demonstrates installation of Oracle Linux release 6 update 9, running as Oracle VirtualBox guest machine.

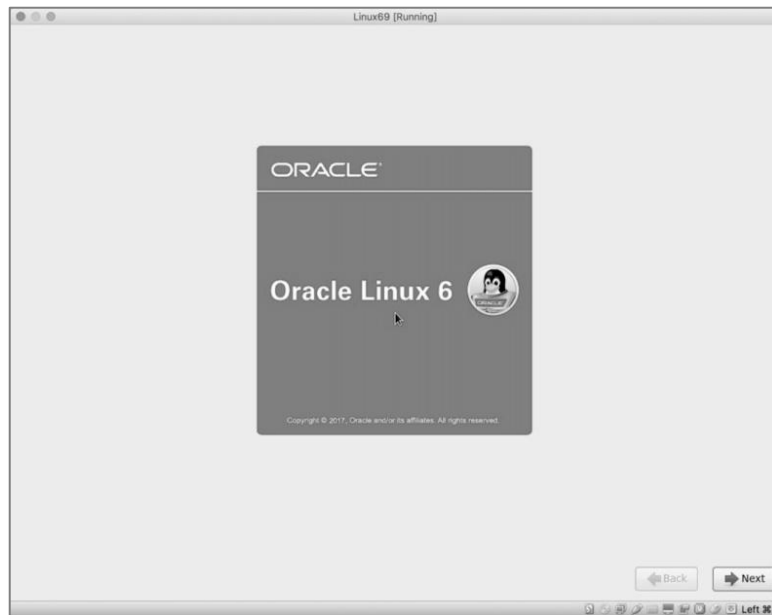
*Step 1: Select Install or upgrade an existing system, click “Enter”.*



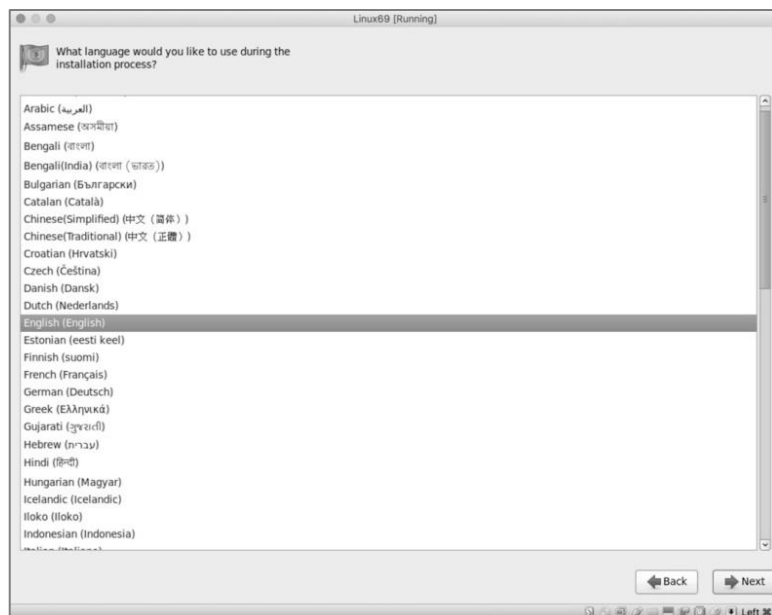
*Step 2: Tab to “Skip”, click “Enter”*



*Step 3: Click Next*

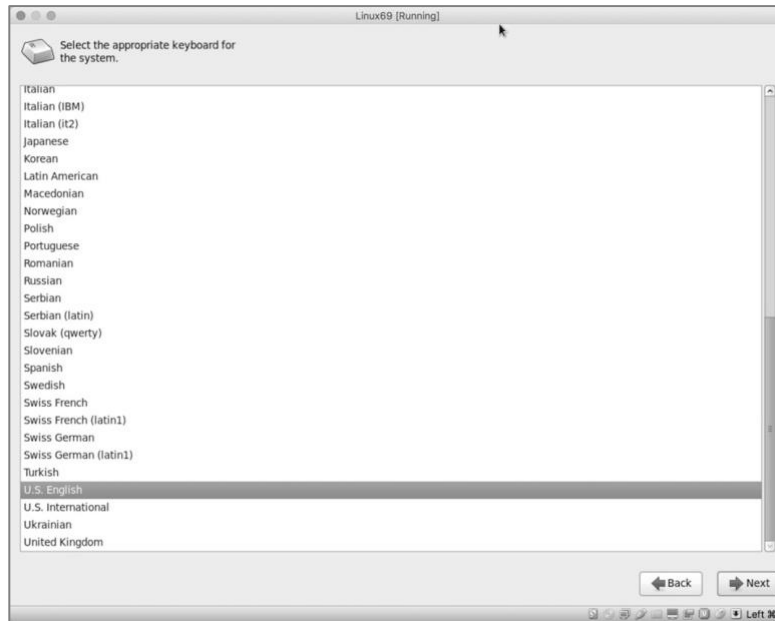


*Step 4: Installation language, select “English”, click “Next”*





*Step 5: System keyboard, select “US English”, click “Next”.*



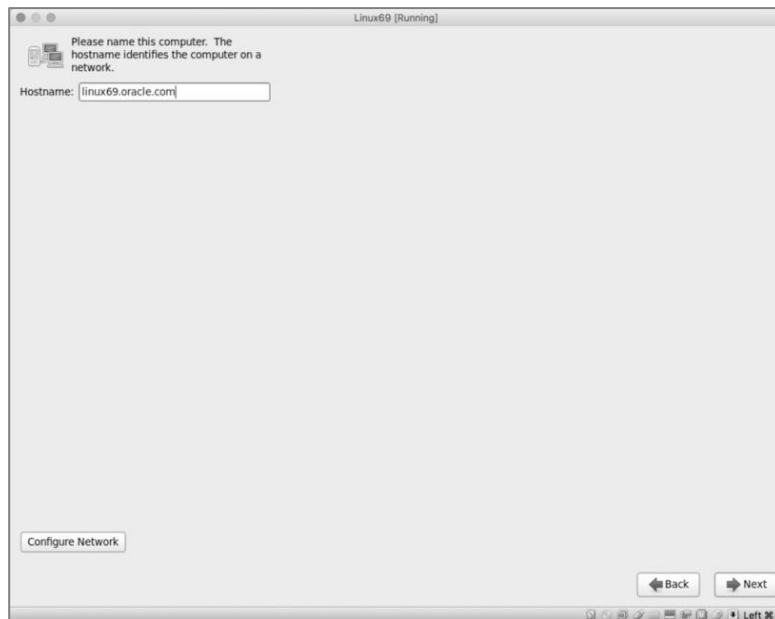
*Step 6: Select “Basic Storage Devices”, click “Next”*



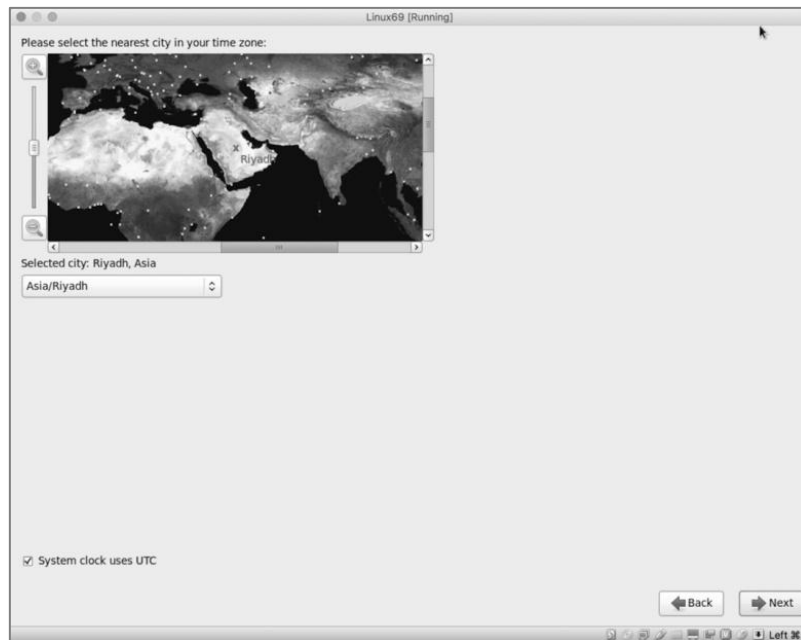
*Step 7: Click on “Yes, discard any data”, click “Next”*



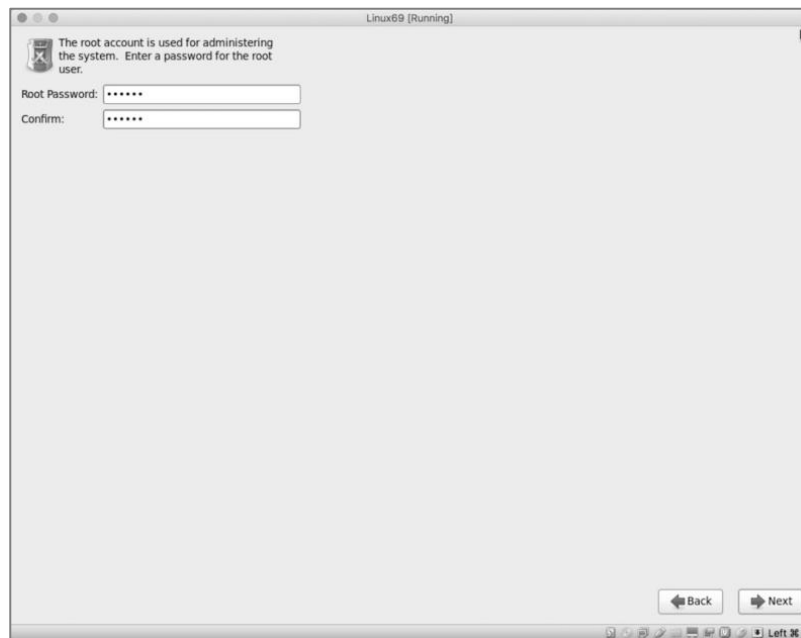
*Step 8: Enter the hostname, click “Next”*



*Step 9: Select the city time zone, Asia/Riyadh*



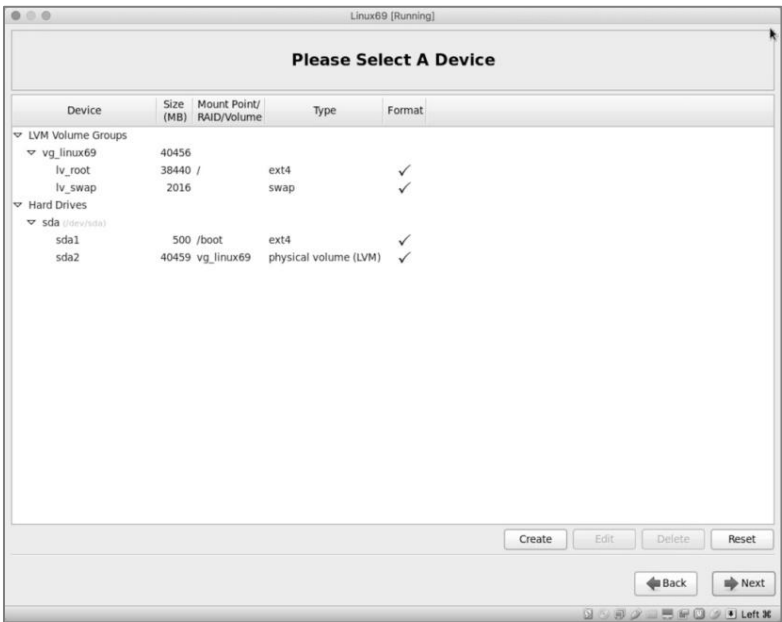
*Step 10: Enter the root password (oracle)*



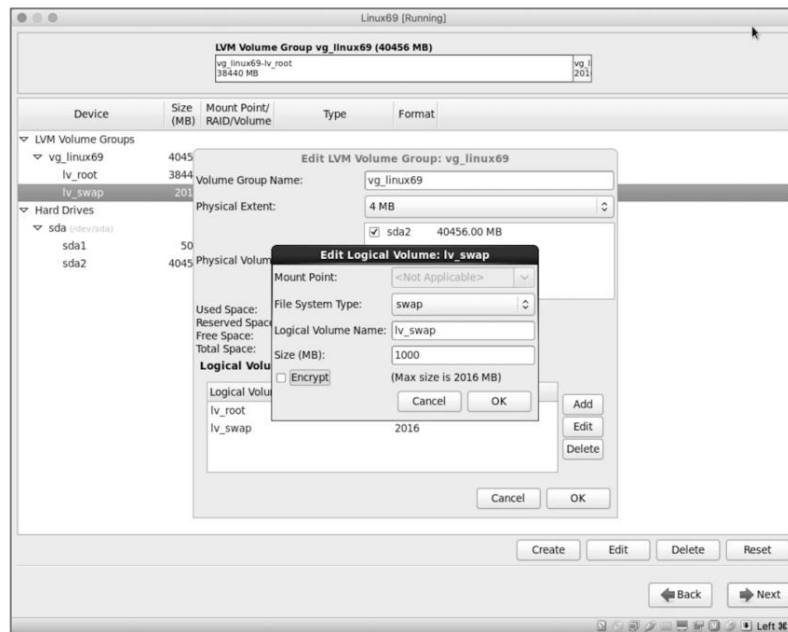
Step 11: Select installation type, Replace Existing Linux System(s). Enable Review and modify partitioning layout.



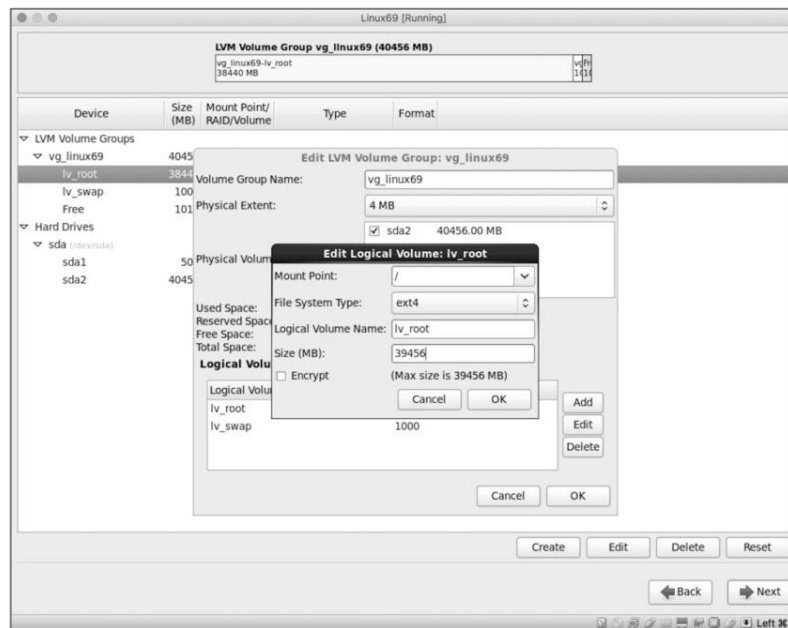
Step 12: Review the current devices and mount points. Changes them as shown next.



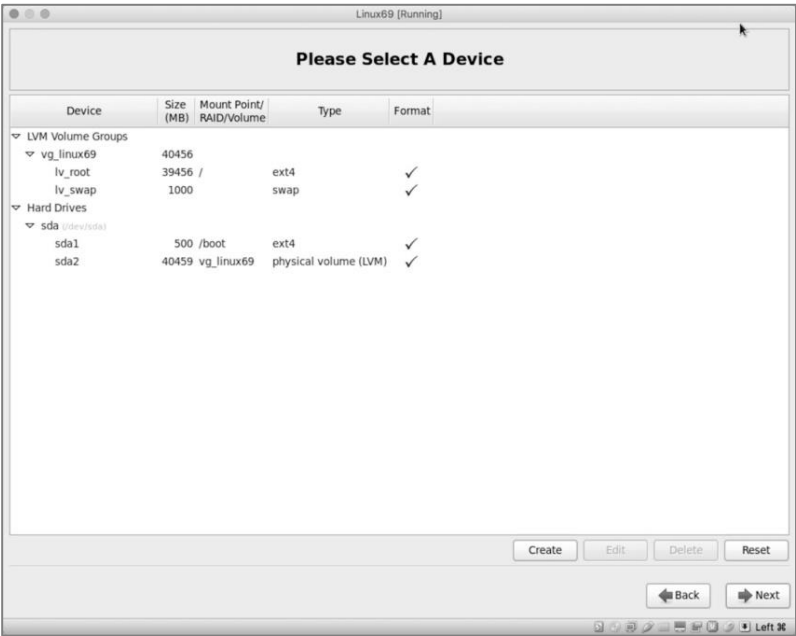
Step 13: Select “swap”, click on “Edit”, and enter Size (MB) 1000, click “OK”



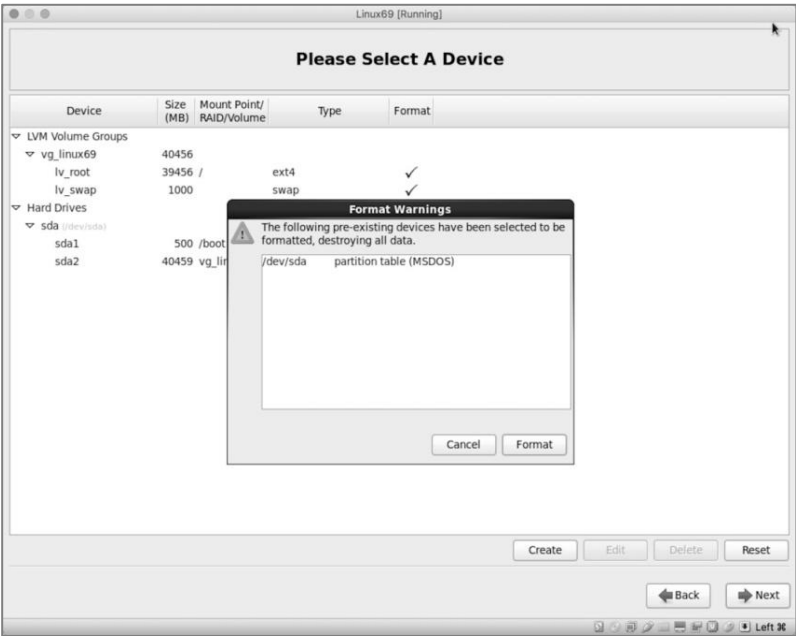
Step 14: Select “/”, click on “Edit”, enter the remaining maximum size.



Step 15: Review device settings. Swap and “/” are the only mount points.



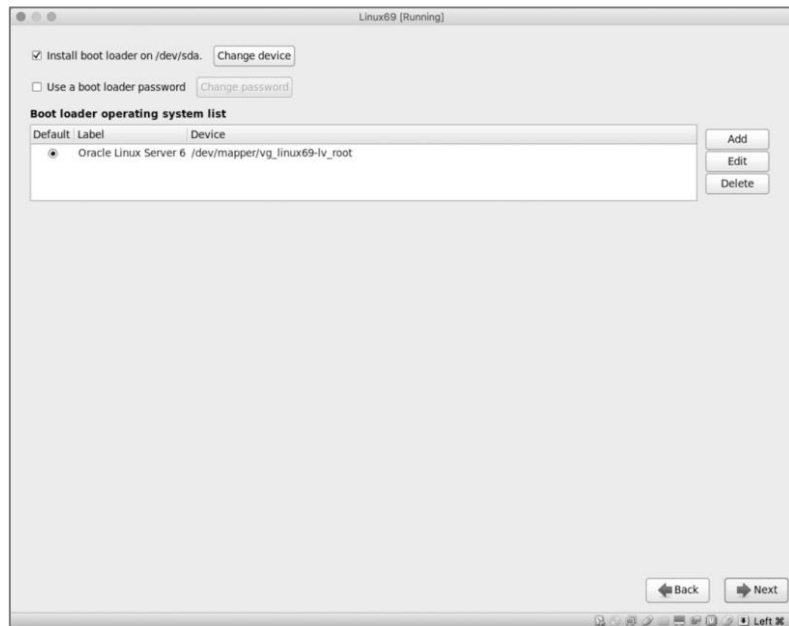
Step 16: Click “Format” to finalize and format device settings.



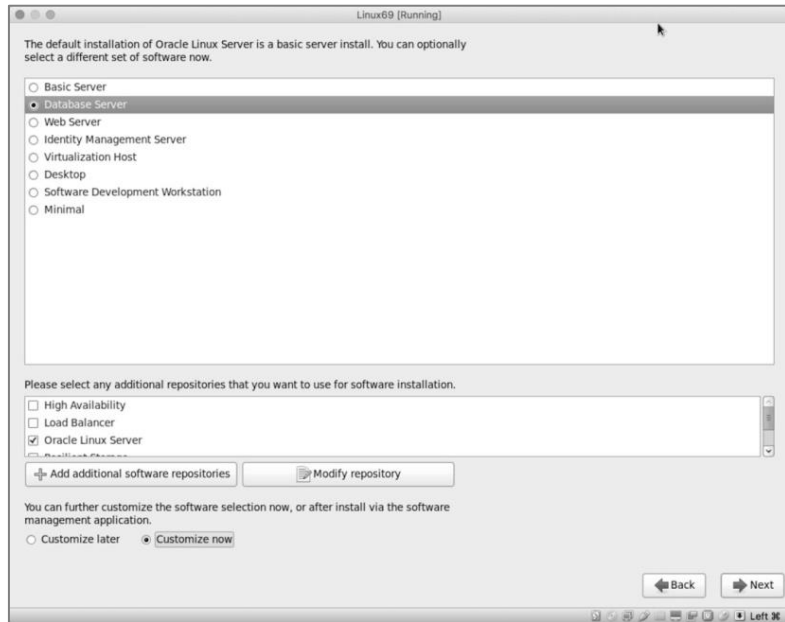
Step 17: Click on “Write device configuration to disk”



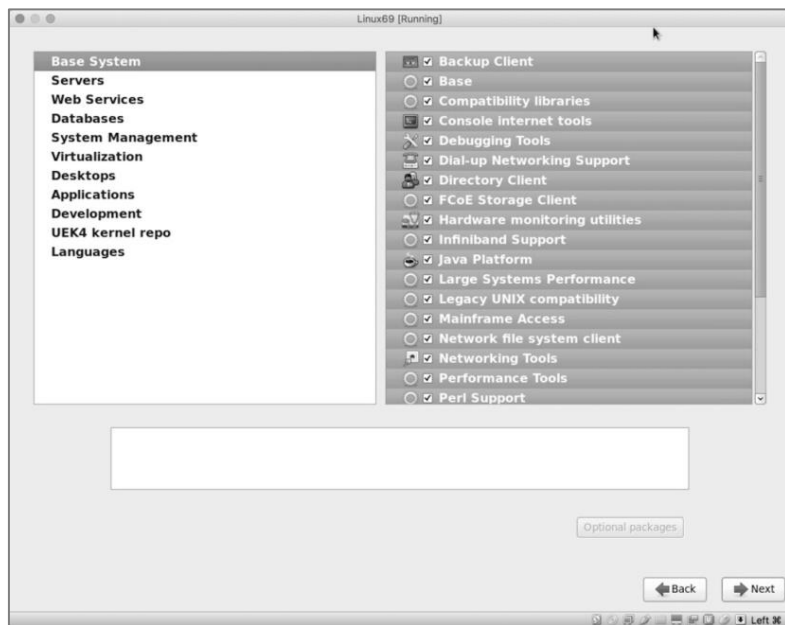
Step 18: Install boot loader on `/dev/sda`, click “Next”



*Step 19: Select “Database Server” and “Customize Now”, “Next”*

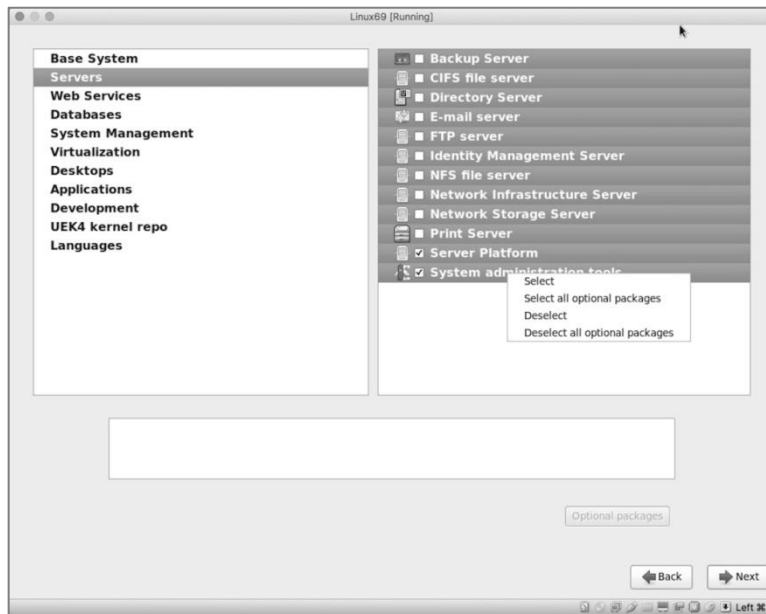


*Step 20: Click on Base System, click on the first entry, hold the shift key, click on the last entry, and then on the right mouse bottom.*

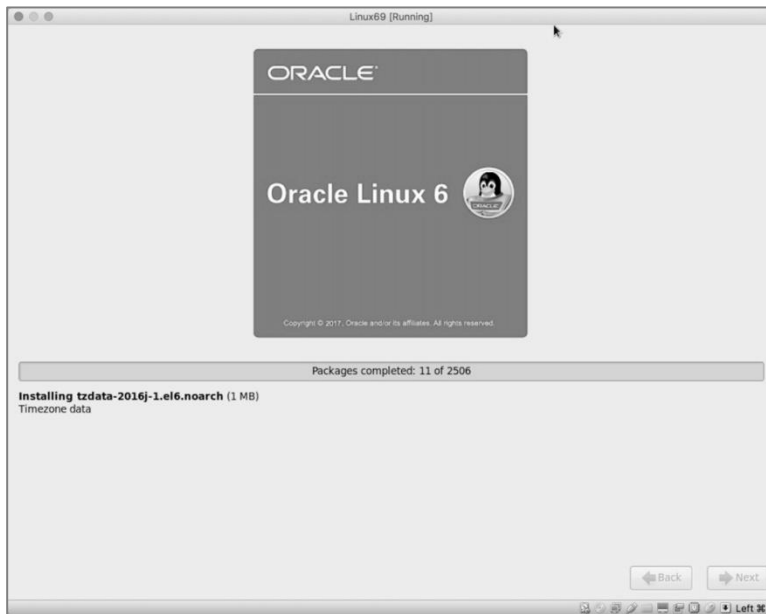




*Step 21: Select the second component – Servers, and repeat the same keys clicks. Click Select all optional packages when ready. Click “Next” after finishing Web Service, Databases, System Management, Desktop, Applications, and Development.*



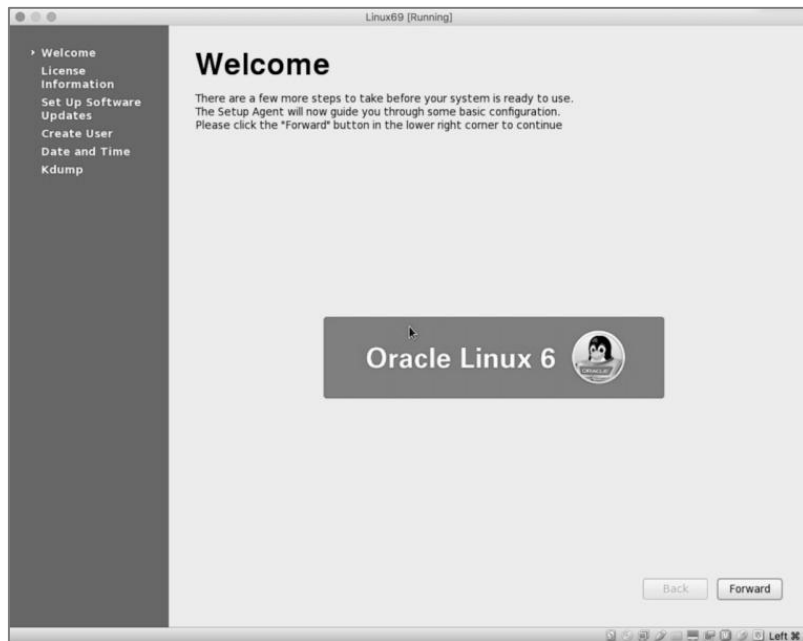
*Step 22: Wait until all packages installation complete.*



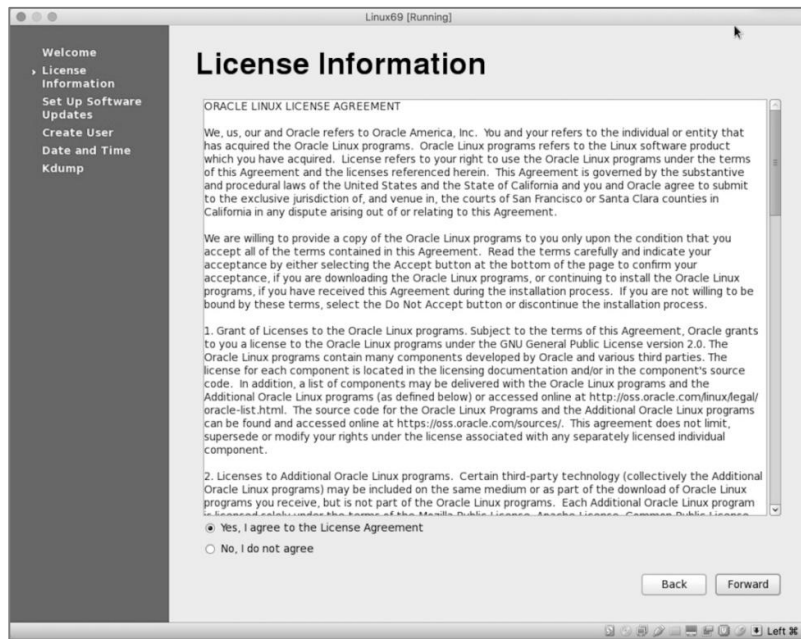
*Step 23: Click on “Reboot”*



*Step 24: Click on “Forward”*



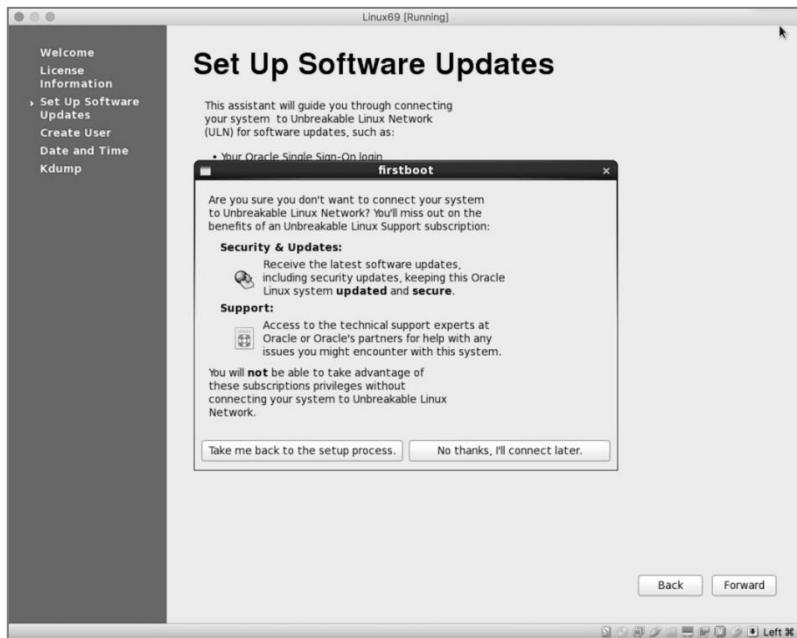
*Step 25: Read License Information, click “Forward”*



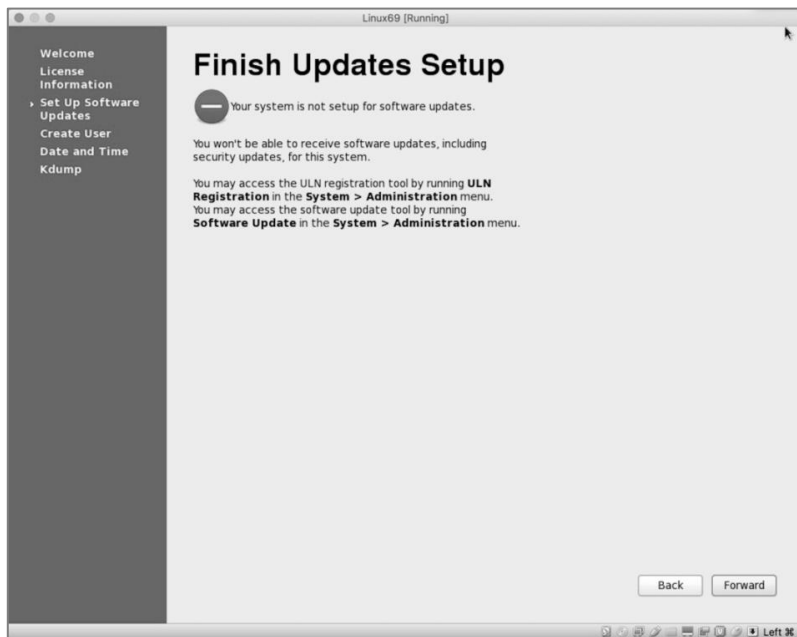
*Step 26: Setup Software Update, select “No, I prefer to register at a later time”, click “Forward”*



Step 27: Click on “No Thanks, I’ll connect later”



Step 28: Click “Forward”



*Step 29: Do not Create User, by click on “Forward”, “oracle” user will be created.*

The screenshot shows the 'Create User' window in the Oracle Linux installer. The window title is 'Linux69 [Running]'. On the left is a sidebar with a list of steps: 'Welcome', 'License Information', 'Set Up Software Updates', 'Create User' (highlighted with a right-pointing arrow), 'Date and Time', and 'Kdump'. The main area has the title 'Create User' and a paragraph: 'You must create a 'username' for regular (non-administrative) use of your system. To create a system 'username', please provide the information requested below.' Below this are four input fields: 'Username:', 'Full Name:', 'Password:', and 'Confirm Password:'. Further down, there is a paragraph about network authentication and a button 'Use Network Login...'. Below that is another paragraph about advanced options and a button 'Advanced...'. At the bottom right are 'Back' and 'Forward' buttons.

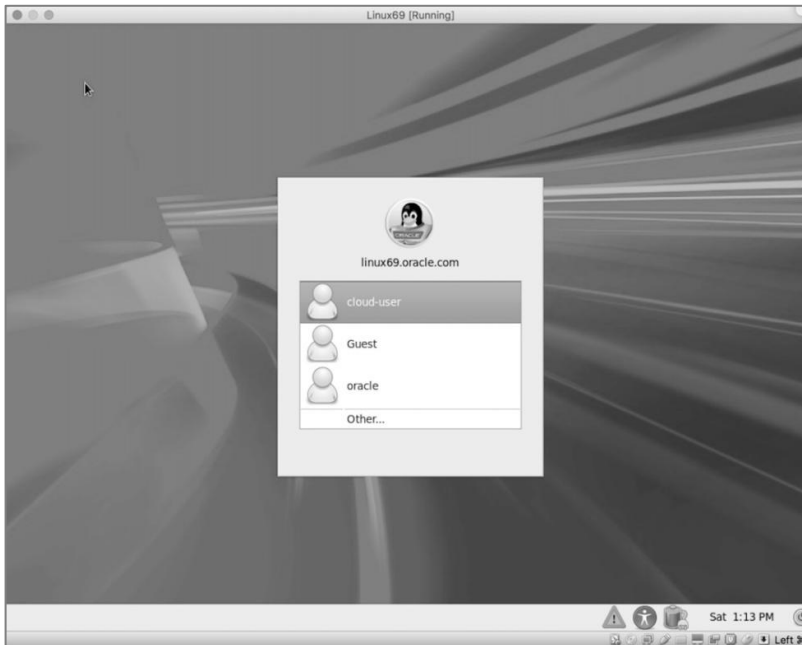
*Step 30: Verify the date, time and click “Forward”*

The screenshot shows the 'Date and Time' window in the Oracle Linux installer. The window title is 'Linux69 [Running]'. The sidebar on the left is the same as in the previous screenshot, with 'Date and Time' highlighted with a right-pointing arrow. The main area has the title 'Date and Time' and the text 'Please set the date and time for the system.' Below this is a section titled 'Date and Time' with a sub-header 'Current date and time: Sat 14 Apr 2018 01:12:04 PM AST'. There is an unchecked checkbox 'Synchronize date and time over the network'. Below that is the text 'Manually set the date and time of your system:'. This section contains two parts: 'Date' and 'Time'. The 'Date' part shows a calendar for April 2018, with the 14th selected. The 'Time' part has three spinners: 'Hour' (set to 13), 'Minute' (set to 9), and 'Second' (set to 37). At the bottom right are 'Back' and 'Forward' buttons.

*Step 31: Click “Finish”, and wait for the reboot to complete.*



*Step 32: Click on “Other...”, enter username (root) and password (oracle)*

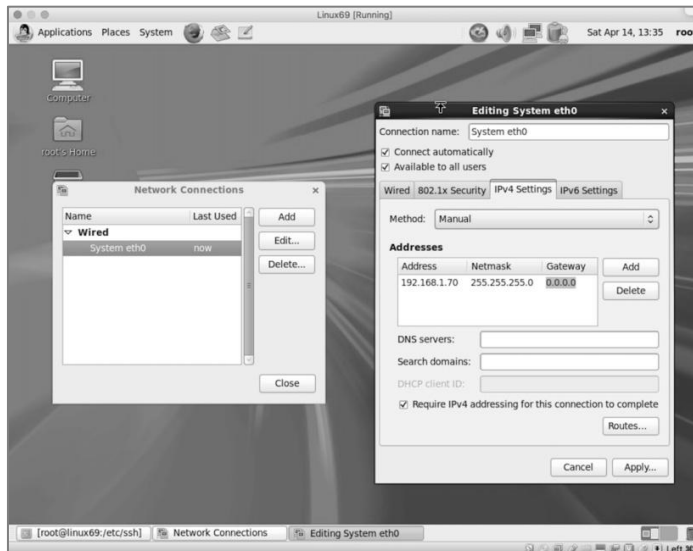


# Networking Setting

Navigate to Virtual Machine setting, select “Network”, Select “Bridged Adapter”, Allow All, Cable connected, and click “OK”



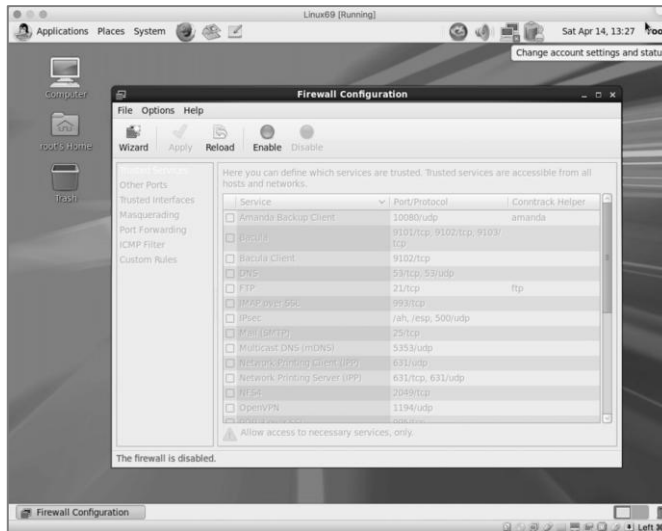
Click on System, Network Connection, click on System eth0, and select IPv4 Settings, Enter IP address, Netmask. Select Connect Automatically and click “Apply”.



# Disable Security Settings

The following steps are to allow maximum flexibility to work on this test environment. These steps are performed differently on production environments.

## Step 1: Disable Firewall



## Step 2: Disable iptables and ip6tables





Step 3: Edit ssh setting by editing /etc/ssh/sshd\_config file. Enable by un-commenting PermitRootLogin and PasswordAuthentication lines.



```

root@linux69:/etc/ssh
File Edit View Search Terminal Help
#LoginGraceTime 2m
PermitRootLogin yes
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

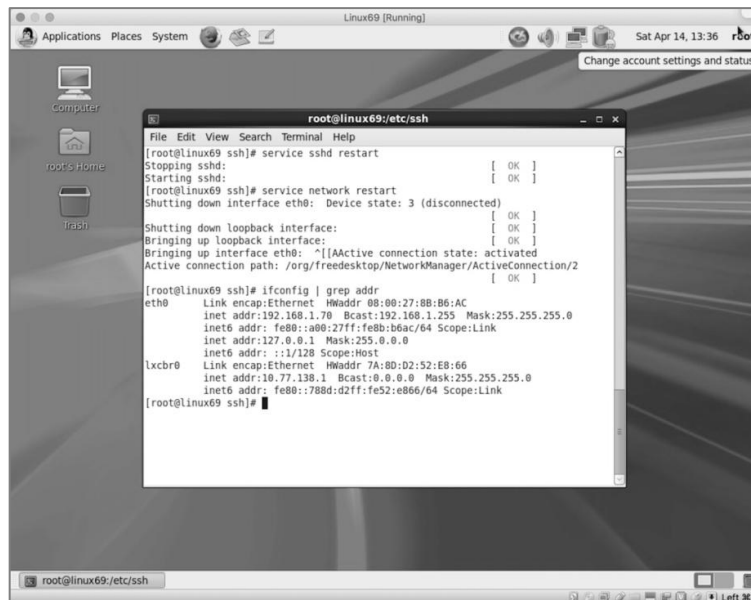
#RSAAuthentication yes
#PubkeyAuthentication yes
#AuthorizedKeysFile .ssh/authorized_keys
#AuthorizedKeysCommand none
#AuthorizedKeysCommandRunAs nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#RhostsRSAAuthentication no
# similar for protocol version 2
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
#RhostsRSAAuthentication and HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication no

```

Step 4: Restart the service for: sshd, network. And, verify the IP Address.



```

root@linux69:/etc/ssh
File Edit View Search Terminal Help
[root@linux69 ssh]# service sshd restart
Stopping sshd: [ OK ]
Starting sshd: [ OK ]
[root@linux69 ssh]# service network restart
Shutting down interface eth0: Device state: 3 (disconnected) [ OK ]
Shutting down loopback interface: [ OK ]
Bringing up loopback interface: [ OK ]
Bringing up interface eth0: ~[[Active connection state: activated
Active connection path: /org/freedesktop/NetworkManager/ActiveConnection/2 [ OK ]

[root@linux69 ssh]# ifconfig | grep addr
eth0      Link encap:Ethernet  HWaddr 08:00:27:08:B6:AC
          inet addr:192.168.1.70  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe8b:b6ac/64 Scope:Link
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
lxcbr0    Link encap:Ethernet  HWaddr 7A:80:D2:52:E8:66
          inet addr:10.77.138.1  Bcast:0.0.0.0  Mask:255.255.255.0
          inet6 addr: fe80::788d:d2ff:fe52:e866/64 Scope:Link
[root@linux69 ssh]#

```

Step 5: Open an ssh connection as shown next.

```
$ssh root@192.168.1.70
```

Are you sure you want to continue connecting (yes/no)? **yes**

Password, "**oracle**"

A terminal window titled 'falswaimil — root@linux69:~ — ssh root@192.168.1.70 — 80x24'. The window shows the following text:

```
fawzis-MacBook-Pro:~ falswaimil$ ssh root@192.168.1.70
The authenticity of host '192.168.1.70 (192.168.1.70)' can't be established.
RSA key fingerprint is SHA256:m9H/gjLZ9qPYubnRwFPCY4epgQiqSUy10/dwizfRRjY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.70' (RSA) to the list of known hosts.
root@192.168.1.70's password:
[root@linux69 ~]#
```

# Summary

Without a single doubt, Linux has dominated the operating systems industry, it's found every where. It's powering the security network cameras, the jetliner entertainment systems and the large-scale database servers running critical mission applications. It's an open source, hence it's cost-effective and a wise choice when bundled with first class support from Oracle corporation.

The transition of MS Windows operating system users to Linux is essential to manage an Enterprise Data Warehouse (EDW). The users are expected to be familiar with frequently executed commands and their parameters. MS Windows users who are familiar with MS power shell interface should find the transition to Linux a quick journey, and have so much in common in term of managing an Oracle Database.

The installation of Oracle Linux was illustrated for a research, test and educational environment. In a production environment file systems must be mounted accordingly to the administration guide, and kernel parameters sizing must be set to the minimum recommended by Oracle Database 12c installation guide.

Learning shell scripting is not mandatory, learning the basic is adequate. The shell scripting is all about automating Linux administration. For example, developing a shell script to purge logs and alerts file periodically rather than manually purging the files. The advantage it prevents human errors by executing the wrong command.

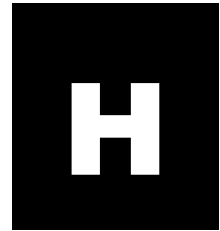
The operating system shell startup file must be configured to import the environment variables for Oracle GoldenGate 12c and Oracle Database 12c, the location of the startup file depends on the shell type.

Networking in the virtual world has multiple options; the bridge adapter was used to configure networking, assigning static IP address and submask, so that Linux SSH connections are established from external terminals. Because this is a test environment, the following has been conducted to enable ssh connections.

- Disabled firewall
- Disabled iptables and ip6tables
- Re-configure SSHD configuration file – `sshd_config`

The web is a good place to learn Linux fundamentals, search for Linux and start learning the basic administration commands, understand their purpose, their parameters, type them and interrupt their output, gradually moving to advanced commands and shell programming. Surely, you will find the transition to Linux interesting and worth learning journey.

# Glossary



## ABEND

This is the process status; it's the status when the process terminated abnormally. Extracts and Replicat status either: STOPPED, RUNNING or ABENDED. When it's ABENDED, the issue must be corrected before re-starting.

## AOL

Stands for Active Only Log. An Oracle GoldenGate change data capture mode where the primary capture mines the archive log file. The change data capture occurs after the database log switch event occurs.

## API

Stands for Application Programmatic Interface, examples of API are Oracle Database 12c PL/SQL API. API normally composed of members (procedures, functions, constructs), these members are invoked through a secure model.

## ASM

Stands for Automatic Storage Management, It's an Oracle vertical file system for storing and managing Oracle and non-Oracle database files. An ASM instance must be running on the database server.

## ASMM

It stands for Automatic Shared Memory Management, Oracle recommends enabling ASMM by setting the SGA parameters MAX\_TARGET and MAX\_SGA\_TARGET parameters to a value to support the instance workload. The majority of the SGA components are dynamically allocated based on workload distribution.

## **BI**

It stands for Business Intelligence; it composed of set of business analytics dashboards for presenting data from different prospective. An example of BI is building dashboard using Oracle Data Visualization.

## **Bi-directional**

When deploying Oracle GoldenGate 12c where the data flow is 2-way from source to target and from target to source, there is referred to as bi-directional. This is implemented for active-active database high-availability and load balancing.

## **CDC**

Stands for Change Data Capture, Oracle GoldenGate perform low-impact CDC. There are different types of CDC when using Oracle GoldenGate: Classic, integrated locally deployed and integrated downstream deployed.

## **CDR**

Stands for Conflict Detection and Resolution (CDR), when implementing active-active database, CDR applies business rules to enable runtime conflict handling. It occurs when network is disconnected between two more connected databases or when the same row updated nearly at the same time across two or more databases.

## **Checkpoint**

One of the component Oracle GoldenGate group is a checkpoint. The primary purpose of a checkpoint is to enable precise recovery in the event of failure. The primary and secondary process default uses checkpoint file, the Replicat default checkpoint uses database table.

## **Column Mapping**

The assigned of target column to one or more source column. The source column may go through transformation; multiple columns may be concatenated then mapped to a single target column.

## **DaaS**

It's stands for Database as a Service; it's normally listed under Platform as a Service (PaaS). DaaS enable customers to quickly online acquire and provision Oracle database instances, that matching their business needs.

## **Data Guard**

It's Oracle data protection solution. The database is protected remotely by a database replica running as standby database. The data guard is the framework to create and manage the primary and standby database configuration.

## **Database Clone**

It's executed using Oracle Recovery Manger (RMAN) to create an exact copy of the database. Database clone either performed from database backup or from an active database.

## **Deferred**

When the capture process is operating from the archive log file, it's referred to as non-real-time or deferred change data capture. The archive log file can be on the primary or standby database.

## **DEFGEN**

A utility used when operating Oracle GoldenGate 12c in a heterogeneous environment. It generates a definition files to translate the source data types to the corresponding target's data type.

## **DOP**

It stands for the Degree of Parallelism, when executing a data warehouse load operation, parallel processing with high DOP speed up the execution of the load phase. DOP specified at various levels: Session, SQL hint or database. For Oracle GoldenGate Replicat, the keyword PARALLEL is specified when defining the Replicat.

## **Downstream System**

A downstream is a system Oracle GoldenGate 12c is deployed. The system operates the database mining server for producing the logical change records (LCRs). On the downstream system change data capture processed and extract trail is created.

## **DSS**

It stands for Decision Support System; DSS enables business owners to make sound decision, faster and more accurate. When the DSS repository refreshed on real-time, it delivers a real-time view of the business making the decision-making process more effective for a growing competitive market place.

## **EL-T**

It stands for Extract, Load and Transform phases on a data warehouse. This is the modern technology delivered by Oracle Data Integrator. The advantage of EL-T is it eliminates the need for staging, transformation occurs at the target database system.

## **EDW**

It stands for Enterprise Data Warehouse, stores large volume of data to support the decision support systems. It receives data from multiple core systems, transform and load data using high-performance parallel processing. Using data warehouse best practice, there is no staging area, and the data processed through Extract-Transform-Load technique.

## **ETL**

It stands for Extract, Transform and Load phases on a data warehouse. The extracts capture the transaction from the core system and create a stage area, the transformation acts on the stage and load performs high-performance parallel insert into the data warehouse repository.

## **FRA**

It stands for Fast Recovery Area, the destination for different file types: Backup, archive log files, etc. when executing long running transactions, the rate of generating archive log files increase linearly, hence ensure its size large to support the workloads.

## **GGSCI**

Stands for GoldenGate Software Command Interface, used to submit and parse Oracle GoldenGate commands, and to interact with the database. Some of the commands requires login to the database from within GGSCI.

## **Heap Table**

This is the normal Oracle Database table. There are different types of tables: Heap table, index organized table, partitioned table, materialized view and external table. Create heap table using the SQL command CREATE TABLE.

## **Heterogeneous**

It's the database types within an Oracle GoldenGate deployment. When the source and target are of the same database type, then it's referred to as Homogenous, however; when the source and the target are of different database type, then it's referred to as Heterogeneous.

## **IaaS**

It stands for Infrastructure as a Service; with IaaS customers left and shift their workload from on-premise to cloud. With IaaS management and administration is the customer's accountability. The vendor provides the CPU, Memory and storage requirements.

## **Initial Data Load**

Moving data from the source to the target system, referred to as instantiation. The Initial Data Load uses Initial Data Load extract and Initial Data Load Replcat. The Initial Data Load extract scan the table, send the data to the target for the Replicat to apply. Initial Data Load may be processed using the database utilities such as Oracle Recovery Manager (RMAN) or export-import data pump.

## **Intermediate System**

A system placed between the source and target systems. The intermediate system functions as stage server or the trail for re-processing then sent to the target systems. The main function of the intermediate system is to perform massive data transformation. It may run with or without a database.



## **LCR**

Stands for Logical Change Record, used by Oracle GoldenGate integrated extract and Replicat. LCR is the database transactions stored as transient data produced by the database mining server and used as an input to the integrated extract.

## **LOGDUMP**

A utility for mining Oracle GoldenGate trail files. The utility allows navigating inside the internal of a trail file, browsing contents and managing the trail files such as concatenating two trail files.

## **MACRO**

MACRO is function-like structure. It has a name and it's a place holder for Oracle GoldenGate parameters. It's referenced and invoked from the parameter files, so that all parameters are copied over at runtime.

## **Materialized View**

A database objects type for storing the result of a query. Commonly used on Enterprise Data Warehouse (EDW) to enhance the performance of intrusive queries. The materialized is refreshed using materialized view logs.

## **Obey**

It's a file mime type and a command of Oracle GoldenGate 12c. An obey file consists of sequence of GGSCI commands executed serially. It's also the GGSCI command to run an obey file. Obey files are used to deploy Oracle GoldenGate 12c instances.

## **ODS**

Stands for Operational Data Store, it's a repository for current operational data, an online process extracts data from multiple core systems and feed the operational data store to stay current for operational reports.

## **Offloading**

Moving the workload from one system to another system, offloading the database means moving the reporting engine to a dedicated database that can be modified to

expedite the report runtime by introducing indexes and materialized view, for example.

## **OLAP**

Stands for Online Analytical Processing, it's found on the back-end systems running decision support application. These systems require more system resources to support long running transactions performing summaries and aggregations.

## **OLTP**

Stands for Online Transaction Processing, it's found on the core systems running short transactions. These systems include ERP and CRM. OLTP must support large number of users performing quick transactions such as online shopping transactions.

## **Optimizer**

The heart of the Relational Database Management System (RDBMS), as the name suggest, the primary function is to determine the most optimized execution tree for handling SQL statement runtime. An example, the optimizer diverts execution of complex SQL statement from materialized view and not from the base tables.

## **Partitioning**

It's a database method for dividing tables and indexes into segments. Partitioning introduces a management window for VLDB and transparently enhances the application performance. Oracle Database 12c supports 19 partition types.

## **Query Re-write**

When running a query, the database optimizer produces the SQL execution plan. Sometimes the best execution plan is to re-write the query to run from a materialized view which stores the result of the query. Query Re-write is enabled by default since Oracle Database 10g.

## **RAC**

It stands for Real Application Cluster, designed to support multi-instance database to guard the from instance failures. RAC deliver high-performance scalable computing without modifying the application. Data warehouse application greatly benefit from RAC for handling workload across nodes.

## **Range Function**

The range function is implemented to deliver parallel processing for extract and Replicats. Range function provide controlled parallel processing of extract, reading the extract trail and applying transaction to the target database.

## **RAT**

It stands for Real Application Testing, it composed of two packages: Database replay and SQL Performance Analyzer (SPA). Database Replay is commonly used to measure performance after database upgrade and SPA commonly used after application upgrade. RAT automates an end-to-end technical testing.

## **RBA**

RBA stands for the Relative Byte Address. Oracle GoldenGate uses RBA when reading and writing to a trail file.

## **Real-Time**

When the capture process is operating from the online redo log file, it's referred to real-time change data capture. It's also extended to operate from the standby online redo log file when Oracle GoldenGate operating on active data guard mode.

## **Redo-Transport**

The shipment of the transactions logs from the source to the downstream (or standby database) is referred to as Redo-Transport service. The shipping is done using the database connect string of the receiving database.

## **Response File**

Oracle universal installer has two modes, interactive and silent. When running the installer in silent mode it requires an input called response file. The response file stores the pair of attribute and value to provide the input to install the software.

## **RMAN**

It's the Oracle Database Recovery Manager, its one of the method to perform initial data load for Oracle-to-Oracle database and database clone. The primary function is to perform online backup, restore and recovery operation for Oracle database.

## SaaS

It stands for Software as a Service; customers run their applications such as HCM, ERP and CRM on the cloud. Upgrade and patching the application is the cloud provider accountability. The customers pushed to focus on the business rather than the platform in term of hardware, operating system and database.

## Schema

It's a database user who owns at least one database object. Oracle GoldenGate database administrator is a schema because it stores the checkpoint table, login to the database and invoke procedure and run queries.

## CSN

CSN stands for the Commit Sequence Number. It's the current position of the read or write of Oracle GoldenGate process. The structure of CSN depends on the database type, for Oracle database it's the same as System Change Number (SCN).

## Sequence

An Oracle database object, it stores sequenced number on memory. Primarily create to support artificial primary key. Because it's stored in memory, it delivers higher performance when working with sequential numbering.

## SGA

Stands for System Global Area (SGA), it's the memory allocated for Oracle Database 12c instance. The SGA is allocated when starting the instance; it's dynamically allocated using Automatic Shared Memory Management (ASMM).

## SQL

It stands from Structured Query Language, it's easy to learn and use to interact with Relational Database Management System (RDBMS). The major SQL DML statements are: SELECT, UPDATE, DELETE, INSERT and MERGE.

## **SQL Execution Plan**

The SQL execution plan depicts the steps to execute an SQL statement. The database optimizer evaluates multiple execution paths, each has a calculated weight, and the path with best execution plan is used to run the SQL statement.

## **SQL Loader**

The SQL Loader is a database utility for loading text data into the database tables. It's originally used for database migration across platforms. The input to SQL Loader is a control file which describes the data format and the data input file.

## **SQLEXEC**

It's an Oracle GoldenGate parameter for extending its capabilities. The purpose is to perform complex operations beyond that are found with Oracle GoldenGate functions by invoking stored database procedure and SQL queries.

## **Streams Pool Size**

The area of the SGA for Oracle streams and integrated extract, configured using STREAMS\_POOL\_SIZE parameter, the minimum recommended value is 750M, however; higher value depends on the transactions workload of the source database.

## **Synchronization**

It's the continuous application of the transactions to the target database to achieve real-time database synchronization. In the case of Oracle GoldenGate Replicat, the Replicat types are: Classic, coordinate, parallel, integrated and non-integrated.

## **Unidirectional**

When deploying Oracle GoldenGate 12c where the data flow is 1-way from source to target only, this is referred to unidirectional. Implemented for query offloading and zero downtime upgrade and migration.

## **User Exit**

It's an external program developed in C or C++. Oracle GoldenGate 12c call-out to user exit to perform complex tasks such as low arithmetic operations. A Common task of User Exit was conflict detection and resolutions, prior to introducing CDR.

## **Veridata**

The tool for comparing pairs of tables by performing low hash-level compares operation between column-row values. Veridata produces out-out-sync report and repair steps.

## **VLDB**

Stands for Very Large Database, its non-standard database growing exponentially, examples of VLDB include data warehouse, and operational data store. VLDB stores large objects data types with API as retrieval methods.

## **VPD**

It stands for Virtual Private Database, it enhances applications and database security and promote the performance. A table is logically partitioned at run based on fine-grain security model. Lookup values are stored on memory for faster reference.

## ABOUT THE AUTHOR

Fawzi A Alswaimil had purchased and started using Oracle version 6 for DOS Operating System in 1988 during his undergraduate study at the University of Arizona. Where he adapted Liner Programming techniques using Oracle PRO\*C to develop “Rivers Flood Simulator” for his senior project. He graduated with a Bachelor of Science in Engineering Mathematics, May 1989.

His first job was an Oracle DBA (1989-1994) with Oracle 7.3 and earned his Oracle Database Administration Master in 1994. Worked with Oracle CASE tool (Oracle Designer) extensively and published an Oracle Designer White Paper titled “Database Reverse Engineering Using Oracle Designer”. Since the release of Oracle 8, he has been working extensively with Oracle Database options and features for high-availability and scalability, including Advanced Replication, Streams, the Recovery Manager (RMAN), Real Applications Cluster (RAC), Data Guard (DG).

Soon after the acquisition of GoldenGate Inc. by Oracle Corporation in September 2009, he has been dedicated for Oracle GoldenGate. Developed “Oracle GoldenGate Implementer’s Workshop”, implemented Oracle GoldenGate software at major Oracle customer sites, develop utilities for Oracle GoldenGate and delivered the official Oracle University GoldenGate curriculums; For the timeframe 1995-2015, he has been teaching at Oracle University, delivering the vast majority of Oracle Technologies courses.

On the other side of his professional career as an Oracle Database administration, he is an advanced Oracle for .NET and Oracle Web PL/SQL toolkit developer. He developed a light weight PL/SQL API using PL/SQL Web toolkit used by several Portal applications, and several products and utilities using Oracle for .NET including: Xplain®, Internet Protocol Surveillance System (IPSS), Oracle GoldenGate Configuration Assistant (GGCA), Oracle GoldenGate Companion and Oracle GoldenGate load-balancer.

The most recent milestone of his career, he’s the author of Oracle GoldenGate 12c: Hands-On Guide to Enterprise Data Integration by Rampant. The book precisely covers his experience with Oracle GoldenGate 12c implementations and research. Jun 2016, he joined Oracle as Integration solution specialist, while still he remains one of the most skilled data integration specialist, he spends the majority of his time on application integration arena which include Oracle Cloud Service, Oracle Internet of Things, Oracle API and Data Integration Platform Cloud (DIPC).

## Implement Enterprise Real-Time Change Data Capture and Complex Data Transformation for In-Memory Analytical Applications

An Enterprise Data Warehouse (EDW) is a platform for the business decision makers to act confidently, and develop a predictive model for their business that is highly accurate. Today's competitors are very innovative, adaption to new trends and economy challenges demands the availability of real-time, illustrative and dynamic fit-for-purpose analytical view. That is when the Enterprise Data Warehouse (EDW) becomes a vital component for a dynamic enterprise. The agility of an Enterprise Data Warehouse (EDW) repository is critical, hence applying the industry best practices play a major role on deploying flexible, scalable and data centric design.

### Oracle GoldenGate 12c 12.3

The Enterprise Data Warehouse (EDW) is hostile by nature. Oracle GoldenGate 12c eliminates the impact of the Enterprise Data Warehouse (EDW) on the core systems by diverting processing to a well-designed database suitable for the EDW repository. Oracle GoldenGate 12c is capable of moving real-time data using downstream system and active data guard technologies which enhances the core systems security and avoid overhead by installing additional software. For an EDW, data transformations are not avoidable, Oracle



### What you will learn from this book

- ☐ Understand the principle of robust Enterprise Data Warehouse design.
- ☐ Learn advanced technique to install Oracle GoldenGate 12c on advanced topology using Oracle GoldenGate 12c downstream.
- ☐ Configure Redo-Transport for shipping of the transaction from the core system to Oracle GoldenGate 12c downstream system.
- ☐ Deploy the integrated extract, configure and defined the integrated extract, extract trail.
- ☐ Configure and define the data pump extract for data streaming to remote targets, and defining the remote trails.
- ☐ Configure coordinated, parallel, integrated apply to support high degree of parallelism (DOP) starting with DOP=25.