ORACLE
GoldenGate 12c



Design, Build and Deploy
EXTENDED DATABASE CLUSTER 12c (EDBC)

Powered By Oracle GoldenGate 12c Downstream Technology

Version 12.1.0.2.0-12.2.0.1.0 Release 1.0

Supporting EDBC adaptability for filed Patent: ORA170402

EXTENDED DATABASE CLUSTER, a method of clustering N active database instances using N Unidirectional Data Streaming

EXTENDED DATABASE CLUSTER 12c Workshop
Developed and Delivered by:
Fawzi. A. Alswaimil
Data Integration Solutions Sales Manager
Saudi Arabia | Kuwait | Bahrain
fawzi.alswaimil@oracle.com | +966 50 345 5094

Design, Build and Deploy EXTENDED DATABASE CLUSTER 12*c* (EDBC)

Powered by Oracle GoldenGate 12*c* Downstreaming Technology

Version 12.1.0.2.0-12.2.0.1.0
Release 1.0

January 2017

Copyright © 2016, 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 EDBC Developer



Fawzi A. Alswaimil is a citizen of Saudi Arabia, since 1984 he has been re-locating between Saudi Arabia in the United States of America. He earned his Bachelor of Science degree in Engineering Mathematics with emphasis in Computer Science, 1989 from the University of Arizona, Tucson. Began his Oracle carrier during Spring of 1988, when he had developed rivers flood simulator using Oracle 6 and PRO*C using Linear Programming techniques. He's the author of the book: Oracle GoldenGate 12c: Hands-on Guide to Enterprise Data Integration Solutions by Rampant TechPress. On April 1993 he earns his Oracle Master Certification.

1989, his first job was a DBA at the National Company for Cooperative Insurance (NCCI), where he was managing the core systems databases and supporting an in-house team of developers using Oracle 6 and Oracle 7 to deploy an integrated core systems. 1995, he joined Oracle Systems Ltd. as a support analyst for Oracle Database and Oracle Designer, during which he published his first Oracle White Paper – Designer/2000 Applications System; Technical Overview, March 1997.

1997, he relocated to Fairfax, VA, working for ARIS, a US based Oracle partner. He was a senior Oracle Consultant delivering Oracle services to customers across the US, with focus on database optimization, backup & recovery and database replication. 1999 he re-joined Oracle System Ltd. as support team leader in Saudi Arabia where he mostly was delivering on-site support services for telecomm and financial institutes' clients.

Mid of 2000, he joined STRATIKA. A Minneapolis, MN based IT company. He led the development of several projects including Remote-DBA and Xplain®. During his work with STRATIKA he developed his first database curriculum for DBAs and developers, while managing and delivering Oracle technology consulting projects. On 2003 he join Oracle Systems Ltd. as Oracle Education Manager, managing and delivering Oracle education business unit in Saudi Arabia.

In 1-Jan-2008, he started his own IT business and founded PreciseTrace. PreciseTrace specialized on Oracle high-availability solutions and delivered consulting mostly on Oracle core technologies: Active Data Guard, Recovery Manager, Real Application Cluster and Oracle GoldenGate. During which he has developed several Oracle GoldenGate toolset.

Since 1-Jun-2015, he is a member of a unique organization – Oracle Data Integration Solutions (DIS) sales team; positioning, selling, road-show and developed-deliver workshops of Oracle GoldenGate 12c in Saudi Arabia, Kuwait and Bahrain.

What is Extended Database Cluster

Extended Database Cluster (EDBC) is an elastic design for fit-for-purpose high-availability architecture; empowering and tightly integrating Oracle Database 12c and Oracle GoldenGate 12c technologies. EDBC rapid elasticity is the uniqueness for matching the computing requirements; when business expand, EDBC compute capacity linearly extended and when business shrink EDBC compute capacity de-extended; this complex restructure is performed online without interrupting running systems. Figure 1 depicts Extended Database Cluster (EDBC) components and data streaming for a compute capacity (CC) of degree 4; one primary node, three expended nodes and four downstream systems, hence the EDBC CC=4.

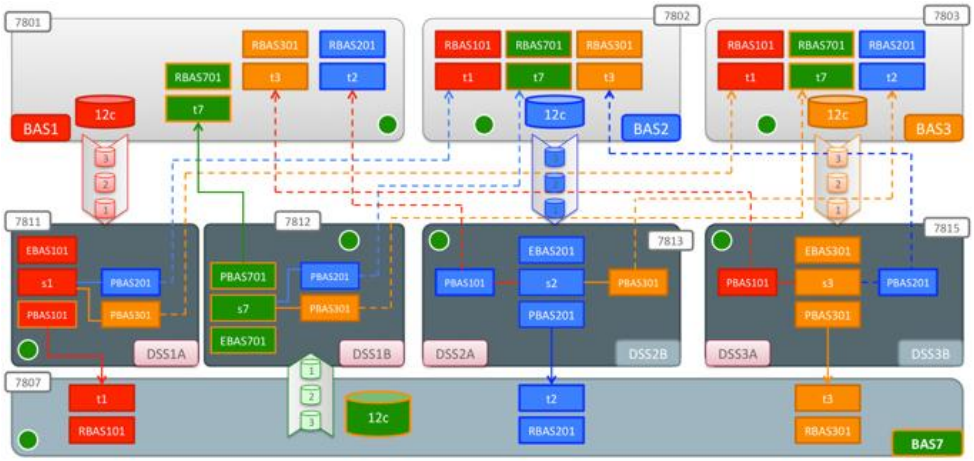


Figure 1: Extended Database Cluster (EDBC) components with compute capacity of degree 4

An Extended Database Cluster (EDBC) composed of two or more nodes, there will be one-and-only-one primary node, and one-or-more extended node(s). EDBC nodes are running active Oracle database instances and interconnected via secure TCP/IP, allowing optimized, compressed, real-time data streaming for high-performance cluster-wide synchronization for extreme-availability with uncompressed data integrity. When real-time data streaming is suspended, EDBC nodes continue to operate in non-clustered-mode, and automatically resumes clustered-mode, when networked.

For an enterprise; data compression, data encryption and security across EDBC nodes is an obligation. EDBC deployable using two-level of compression, encryption and security. Redo-Transport is compressed and encrypted when shipped, then compressed and encrypted when streamed, doubling data security and network optimization for up to 80%. The EDBC primary node, normally situated at the corporate data center, performs

centralized operations such as startup, shutdown, backup, etc. of the EDBC nodes and downstream systems.

EDBC for In-Cloud Compute

Instead of EDBC running on-premise, it's deployable as private cloud, public cloud and hybrid cloud. In-cloud allows the provisioning of EDBC environment rapidly as DBaaS instances running on Oracle cloud. EDBC on hybrid cloud, where the primary node is running on-premise and the extended nodes are running in-cloud, the Oracle GoldenGate Cloud Services (GGCS) performs in-bound and out-bound data synchronization. Surely, EDBC is deployable across cloud regions such as Oracle cloud and Amazon cloud.

EDBC for Maximum Availability Architecture (EDBC-MAA)

EDBC-MAA empowers Oracle Real-Application Cluster (RAC) on the primary node, extended nodes, and downstream systems eliminates single point of failures for mission critical systems. Oracle Data Guard (DG) for the primary and extended nodes is a rapidly deployable option. EDBC-MAA empowers Oracle Automatic Storage Management (ASM) for allocation of database files, configuration files and trail files. ASM failure groups automates data recoverability, hence further enhancing EDBC availability.

EDBC for Maximum Security and Performance Architecture

For enterprises demanding high degree of security, EDBC is deployable on maximum security architecture; combining Oracle GoldenGate 12c and Oracle Advanced Security delivers double security blue-print. Data encrypted when transmitted and when at storage. Similarly, combining Oracle GoldenGate 12c and Oracle Advanced Encryption delivers double compression blue-print. Data compression when shipped from origins, compressed when transmitted and when at storage.

EDBC for SMB

Small-Medium Businesses may adapt EDBC for minimum compute capacity and system resources, hence reducing complexity and cost. This environment typically runs on two TCP interconnected systems only; configured as locally deployed Oracle GoldenGate 12c integrated change data capture, where the local Oracle Database 12c instance is used as database mining engine. SMB running Oracle Database Standard Edition, are able to deploy EDBC using classic capture only, which is un-recommended approach.

EDBC Manager

A desktop application for creating and managing an EDBC environment. It provides a 360-degree view of the systems, and from EDBC console performs maintenance operations: startup, shutdown, backup, logs viewing and troubleshooting.

EDBC Data Protection

EDBC uses Oracle Recovery Manager (RMAN) for backup and restore operations. A recovery catalog database resides on the primary node database. An NFS is configured to enable RMAN writes backup from all EDBC system to a centralized backup location on the primary node. The stored backup scripts backup the database files, control files, parameters files and archived redo log files.

EDBC Certification

EDBC Version 12.1.0.2.0-12.2.0.1.0 Release 1.0 is developed on Oracle Linux 5 Update 9, running Oracle Database 12c Enterprise Edition version 12.1.0.2.0 and Oracle GoldenGate 12c version 12.2.0.1.0. However; the certification of EDBC is always subject to Oracle certification matrix found at: support.oracle.com.

EDBC Run-Time Options

EDBC has been certified to run on the following environments:

- Oracle Real-Application Cluster (RAC)
- Oracle Data Guard (DG) and Oracle Active Data Guard (ADG)
- Oracle Automatic Storage Management (ASM)
- Oracle Advanced Compression
- Oracle Advanced Security
- Oracle Partitioning

EDBC for MS Windows

Though the current EDBC release developed and tested on Oracle Linux version 5 update 9, it should be migratable to Microsoft Windows with minimal changes. Customers interested to run EDBC on MS Windows must verify the certification by visiting: support.oracle.com to obtain a certified combination of Oracle Database 12c and Oracle GoldenGate 12c.

Workshop Overview

The purpose of this hands-on workshop is to teach attendees design, build and deploy an Extended Database Cluster 12c (EDBC). The methodology is well-engineered to enable a consistent approach for designing a large yet scalable database cluster composed of n-interconnected active Oracle Database instances. Depending largely on the compute capacity of the cluster, an N-node active-active topology deployed using a set of N *unidirectional* data streaming Oracle GoldenGate processes. Figure 2 shows the architecture of an Extended Database Cluster 12c (EDBC) with a compute capacity of three nodes. The EDBC compute capacity is increased by adding additional extended nodes, online without impacting other nodes availability.

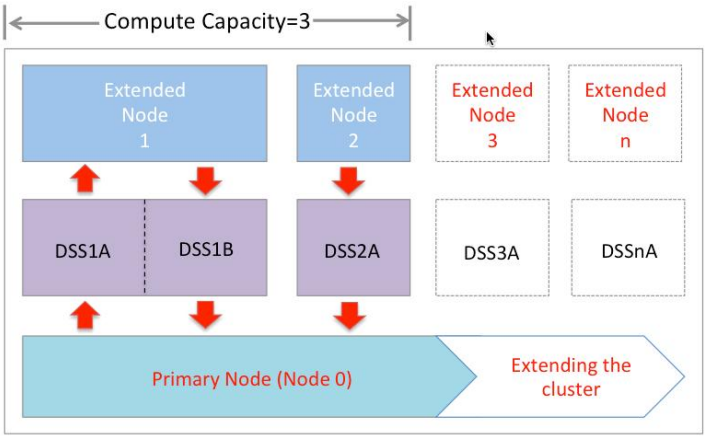


Figure 2: Extended Database Cluster 12c (EDBC)

Designing EDBC is influenced by the cluster compute capacity size, the workshop is built to support up to 3-node cluster; the cluster composed of three downstream systems, one primary node and two extended nodes. The Oracle GoldenGate 12c downstream systems are provided; there will be two Oracle VirtualBox machines for the downstream systems for each participant as shown in Figure 2.

The build phase begins by integrating the Oracle Database 12c with Oracle GoldenGate 12c by configuring the Redo-Transport and the Integrated Capture. The obey files are provided for deployment. They define and configure the interconnected set of Oracle GoldenGate 12c instances; configure the downstream system, deploy the primary node, extended node 1 and extended node 2. As part of the workshop, basic cluster administration tasks of the database cluster are executed.

Though the workshop is developed on a single database instance (non-RAC) environment, applying Oracle Real Application Cluster (RAC) is supported as intended.

EDBC high-availability increased by enabling automatic Oracle GoldenGate 12c instance re-location and restarting Oracle GoldenGate process on the surviving RAC node.

Workshops and Tasks

There are ten (11) workshops, the order of executing Workshops 1-6 is strict and must be adhered. Workshops 7- 11 are optional, but highly recommended. The workshops presume the audiences have an acceptable level of Oracle database and GoldenGate hands-on experience.

To learn Oracle GoldenGate 12c Fundamentals, refer to Appendix A.

Workshop No	Workshop Title	Workshop Timeframe
1	EDBC Workshop Preparation	Day 1
2	EDBC Environment Review	
3	Configuring EDBC Redo-Transport	
4	Configuring EDBC Downstream Systems	
5	Configuring EDBC Data Streaming	
6	Configuring EDBC Nodes Synchronization	
7	EDBC Workload Simulation	Day 2
8	EDBC Conflict Detection and Resolution	
9	EDBC Administration	
10	EDBC Online Database Upgrade	
11	Secure EDBC	
	Duration	**2 Days**

Workshop 1 EDBC Workshop Preparation

This workshop is mandatory and must be successfully completed and verified by the instructor. Please, do not continue if you're unable to complete Workshop 1. Prior to the workshop, it's recommended to install the appropriate Oracle VirtualBox, install Oracle Linux and test connectivity from the host Operating System (Laptop) to the guest machine, the reason is to avoid troubleshooting network connectivity issues ahead of time.

Task No	Task Description
1	Enable and verify Virtualization
2	Install EDBC workshop
3	Setting guest machine, primary node, extended nodes static TCP/IP addresses
4	Connect to cluster network and perform all systems login using PuTTY

Workshop 2 EDBC Environment Review

This workshop aims to familiarize candidates with the database environment across all EDBC systems. Login credentials are provided for root, oracle and the database. Refer Workshop 2 for further details.

Task No	Task Description
1	Review the primary node - EDBCN0
2	Review extended nodes - EDBCN1 and EDBCN2
3	Review downstream systems - C<n>DS1A, C<n>DS1B and C<n>DS2A

Workshop 3 Configuring EDBC Redo-Transport

Configure the Redo shipping from the primary node and extended nodes to the associated downstream systems. Prior to starting this workshop, it's necessary to verify TNS is configured and successful connections from the node to its downstream database mining system. The tasks is similar to configuring Oracle Data Guard instance.

Task No	Task Description
1	Configure Redo-Transport from the primary to downstream C<n>DS1A
2	Configure Redo-Transport from extended node 1 to downstream C<n>DS1B
3	Configure Redo-Transport from extended node 2 to downstream C<n>DS2A

Workshop 4 Configure EDBC Integrated Capture

Begin by registering the Integrated Extract (IE) in the downstream system mining database. Then configure the set of extracts, trails and Replicats. The downstream systems are provided by the instructor for each participant, the physical memory allocation for each downstream system database mining virtual machine is set to 4GB.

Task No	Task Description
1	Prepare the downstream systems
2	Configure integrated capture for the primary node
3	Configure integrated capture for extended node 1
4	Configure integrated capture for extended node 2
5	Review the Oracle GoldenGate 12c instance processes status
6	Perform transactions on the primary node

Workshop 5 Configuring EDBC Data Streaming

This configure a set of data pump processes for data streaming purpose to extended nodes, for enabling data synchronization to participated nodes in real-time. The data streaming is based on business requirements for data decentralization. The workshop designed to deliver decentralization across the cluster.

Task No	Task Description
1	Configure compressed, outbound data streaming of the Primary node
2	Configure compressed, outbound data streaming of extended node 1
3	Configure compressed, outbound data streaming of extended node 2

Workshop 6 Configuring EDBC Nodes Synchronization

This configures a set of data synchronization processes - Replicats. Data is applied to the participated nodes on real-time using the target database SQL native calls. The workshop does not introduce any form of transformations, which is planned for next release.

Task No	Task Description
1	Configure data synchronization for the Primary node
2	Configure date synchronization for extended node 1
3	Configure data synchronization for extended node 2
4	Perform basic data synchronization tests

Workshop 7 EDBC Workload Simulation

Run workloads transactions simulator across the cluster nodes. From the primary, extended node 1 and extended node 2. The PL/SQL API workload simulator already installed and configured for multiple workload levels and DML transaction types.

Task No	Task Description
1	Run workload on the primary node
2	Run workload on extended node 1
3	Run workload on extended node 2
4	Run workload on the cluster nodes

Workshop 8 EDBC Conflict Detection and Resolution

Implement automatic data Conflict Detection and Resolution (CDR) from the applications and Oracle GoldenGate 12c instance configuration. Applying database virtualization (Virtual Private Database – VPD) is an option for data segregation.

Task No	Task Description
1	Setup conflict detection and resolution on IE and Replicat
2	Deploy Virtual Private Database (VPD) for data segregation
3	Testing Insert, update and delete data conflicts

Workshop 9 EDBC Administration

Performing basic EDBC administration tasks. Database Administrators (DBAs) may use the supplied Linux Shell scripts, or the Oracle Enterprise Manager Cloud Control. EDBC empowers Oracle Backup best practice using the Recovery Manager (RMAN).

Task No	Task Description
1	Enable Remote Login
2	Start the Cluster
3	Enable Network File System (NFS)
4	Backup the Cluster
5	Stop the Cluster

Workshop 10 EDBC Online Database Upgrade

A feature of Extended Database Cluster 12c (EDBC) is zero downtime upgrade by performing rollout online database upgrade on EDBC nodes. This workshop upgrades extended node 2 from Oracle Database 11g to Oracle Database 12c.

Task No	Task Description
1	Divert applications to run from extended node 1
2	Stop data streaming of the primary and extended node 1 to extended node 2
3	Stop Replicats and verify no-record-found on extended node 2
4	Report EXTSEQNO, EXTRBA and SCN of extended node 2 Replicats
5	Shutdown GoldenGate 12c on extended node 2 for Oracle database 11g
6	Backup trail files and parameter files

7	De-install GoldenGate 12c on extended node 2 for Oracle database 11g
8	Install Oracle GoldenGate 12c on extended node 2 for Oracle Database 12c
9	Restore trail files and parameters files
10	Start Oracle GoldenGate 12c instance for Oracle Database 12c
11	Start data pumps of the primary and extended node 1 to extended node 2
12	Verify remote trails are generated
13	Disable Redo Transport
14	Backup the Database and Controlfile
15	Perform Oracle Database 12c pre-upgrade
16	Execute Oracle Database 12c Upgrade Assistance
17	Perform Oracle Database 12c post-upgrade
18	Enable database flashback
19	Test applications
20	Rewind the database and disable database flashback
21	Enable Redo Transport
22	Unregister and re-register extended node 2 Integrated Extract
23	Re-create the checkpoint table
24	Re-define the Replicats on extended node 2
25	Start the Replicats on the extended node 2
26	Verify data consistency
27	Divert applications to run from extended node 2
28	Reconfigure RMAN catalog scripts
29	Reconfigure Administration scripts

Workshop 11 Secure EDBC

This workshop applies security features and options to maximize the security of the Extended Database Cluster 12c (EDBC). Depending on the level of data sensitivity, only the specific feature and options are installed and configured.

Task No	Task Description
1	Enable Compression Redo-Transport
2	Enable Encryption of Redo-Transport
3	Enable Password Encryption
4	Command Authentication
5	Configure Secure Trail and Data Streaming
6	Configure Secure Manager
7	Configure EDBC Backup Encryption
8	Deploy Tablespace encryption

Workshop 1: EDBC Workshop Preparation

An Introduction

Workshop 1 is mandatory, it prepares and setup the environment to build and deploy Extended Database Cluster 12c (EDBC) for up to 3-node compute capacity. In order to meet the workshop hardware requirements, the workshop hardware (Oracle VirtualBox machines) is provisioned as follows:

- EDBC Downstream systems: Provisioned for each participant. The Oracle database instance is configured to support a reasonable transactional workload.

- EDBC Primary/Extended nodes: Oracle VirtualBox imported on the candidate laptop. The setup utility will fail to install if the minimum resources are unavailable. Figure 2 shows the EDBC Oracle VirtualBox provisioning.

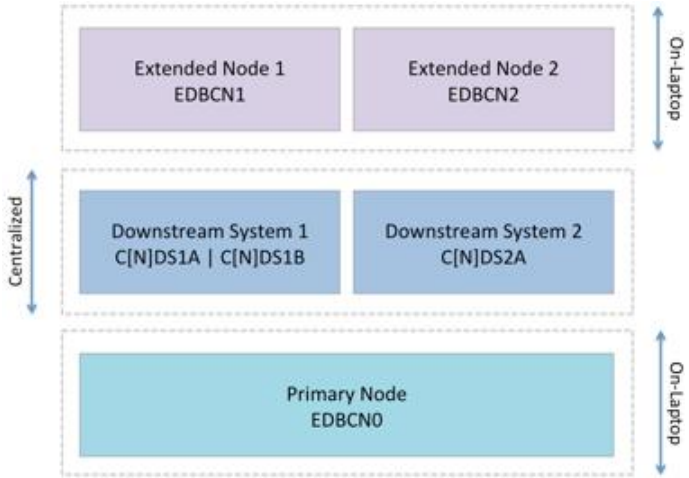


Figure 3: EDBC Hardware Provisioning

Table 1 lists workshop 1 tasks.

Task No	Task Description
1	Enable and Verify Virtualization Technology
2	Installing EDBC Workshop
3	Setting Static IP Addresses and Verify Systems Login
4	Verify Database Environment and Pre-requisites

Table 1: Workshop 1 Tasks

Task 1: Enable and Verify Virtualization Technology

Prior to installing the workshop, the laptop must support virtualization technology. Enabling virtualization depends on the laptop model, but generally while booting the laptop, interrupt the startup by pressing F1 to enter BIOS mode, navigate to Security option to enable Virtualization Technology. Figure 3 illustrates virtualization setting for Lenovo ThinkPad laptop.

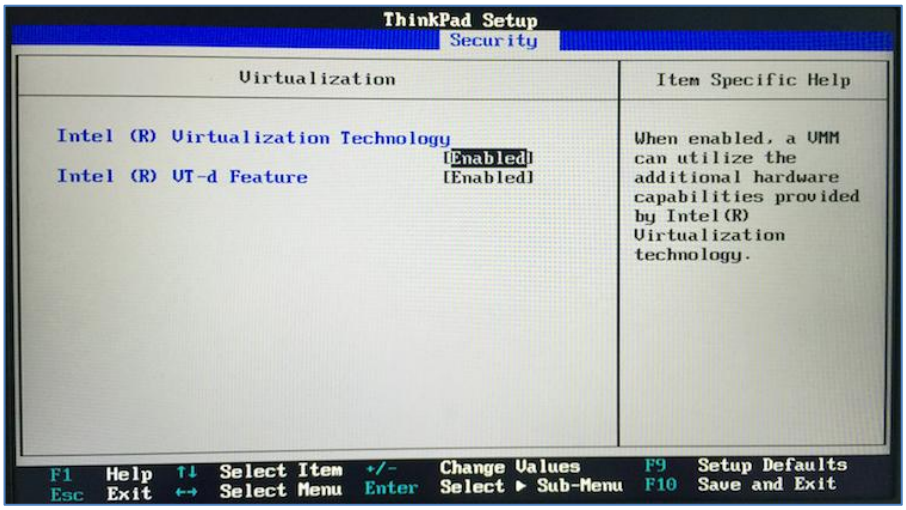
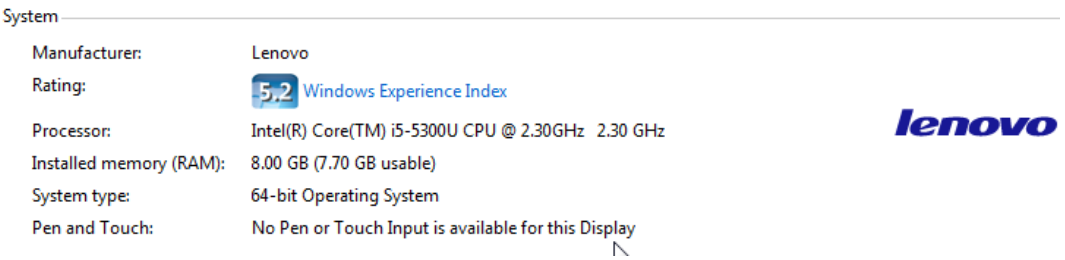


Figure 4: Enabling Virtualization

System Architecture

The workshop requires running MS Windows operating system 64-bit architecture. Verify the operating system support for 64-bit. Navigate to Control Panel ▸ All Control Panel Items ▸ System



The use of MS Windows 32-bit architecture requires manual EDBC installation. The memory allocation of the host operating system will be limited to 2.8GB, however; the guest operating system will be running 64-bit Oracle Linux architecture.

Task 2: Installing EDBC Workshop

- Insert the external disk into USB 3.0 port; double click on the program edbc.exe to start the setup install program shown in Figure 4. The utility create a unique directory formatted as follow: <drive>:\edbc_<machine serial#>\labs. The default destination targets the disk drive with maximum free disk space.

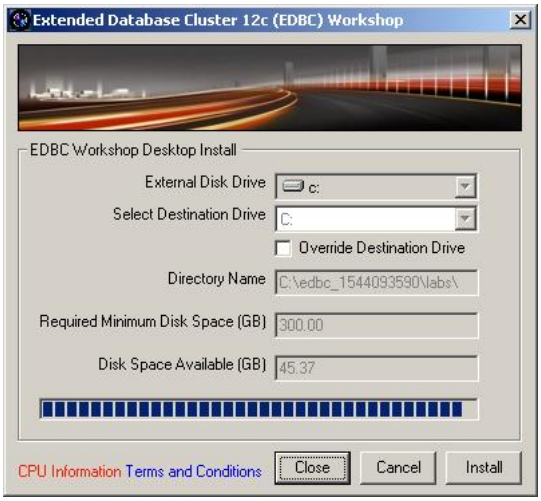


Figure 5: EDBC Workshop Setup Utility

The utility ensures pre-requisites are met, otherwise, the install bottom remain disabled, until the minimum requirements (Disk and Memory) are achieved.

- Determine CPU architecture for 32-bit or 64-bit
- Create the install root directory
- Remove static IP Address, if assigned
- Confirm the minimum free disk space must be available
- Confirm the minimum memory requirements must be available
- Lunch the appropriate Oracle VirtualBox Version 5.1 installer

➲ Make sure no applications are running, start the "Task Manager" to confirm no applications are running, terminate running applications, if any.

➲ Change the destination if necessary. Ensure the new destination has the minimum free disk space available. Uncheck to enable the Install bottom.

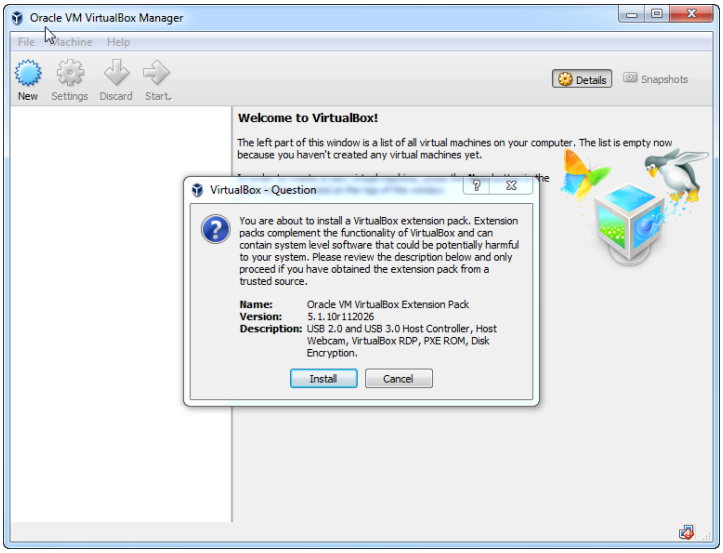
➲ Click on the "Install" bottom. To begin the installation, you must confirm reading and accepting the Terms and Conditions.

Install Oracle VirtualBox Version 5.1

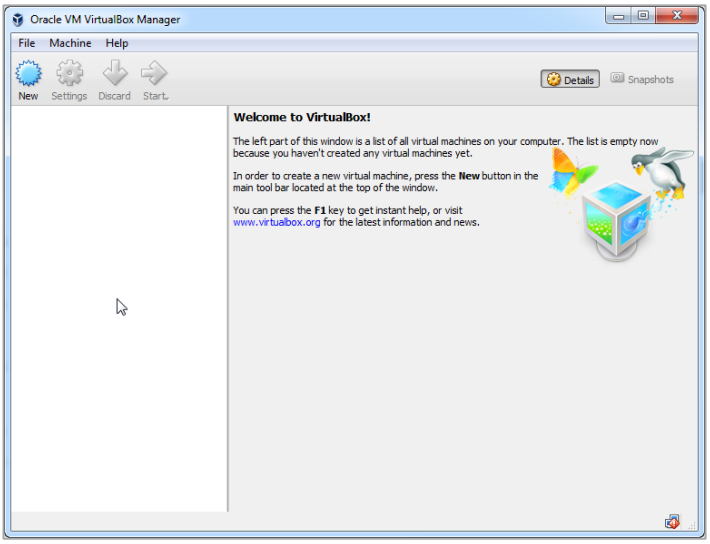
- For MAC OS X users, though the workshop is developed in MS Windows, MAC OS X is supported. Install Oracle VirtualBox Version 5.1 for MAC OS X and follow the same instructions to import the workshop VirtualBox machines.



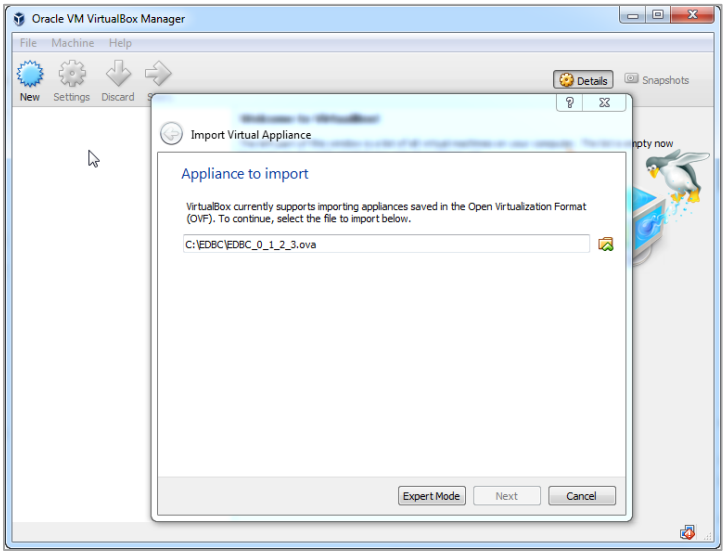
- Install Oracle VirtualBox Version 5.1 Extension Pack



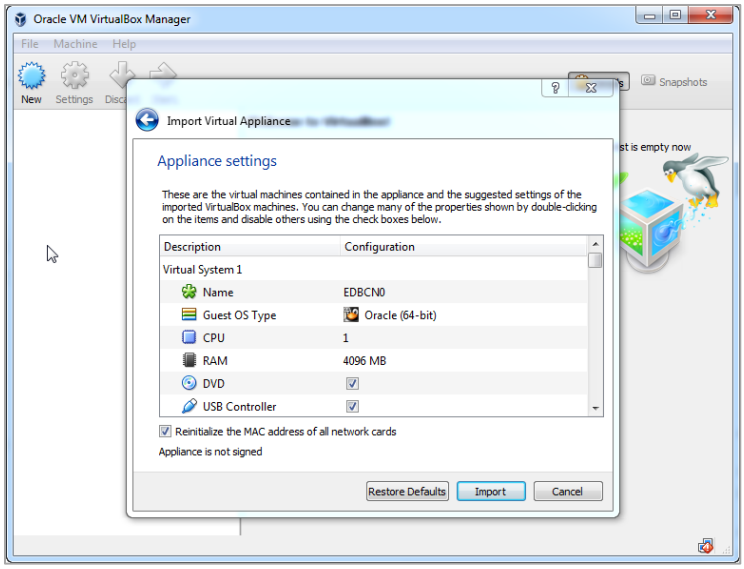
- Start the Oracle VirtualBox Manager.



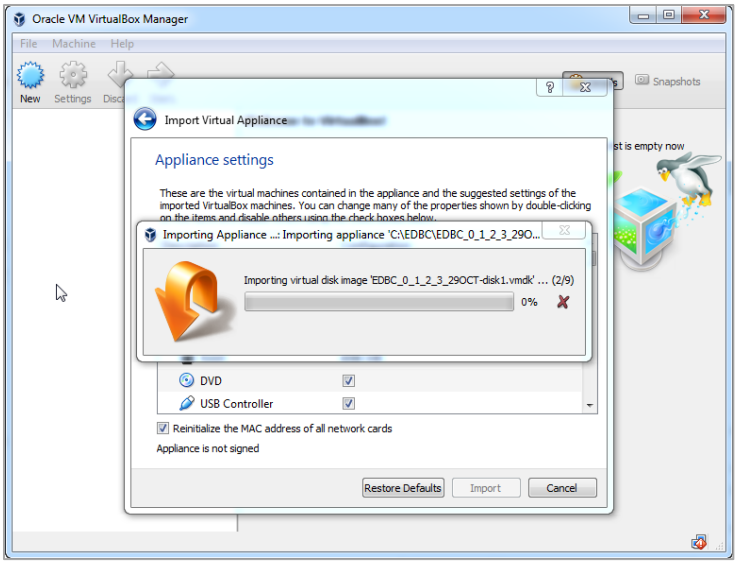
Import the Oracle VirtualBox machines



***Step 1**: Navigate File ▸ Import Appliance, select the OVA file EDBC_0_1_2_3 to import the VirtualBox Machines EDBC0, EDBC1, EDBC2 and EDBC3*

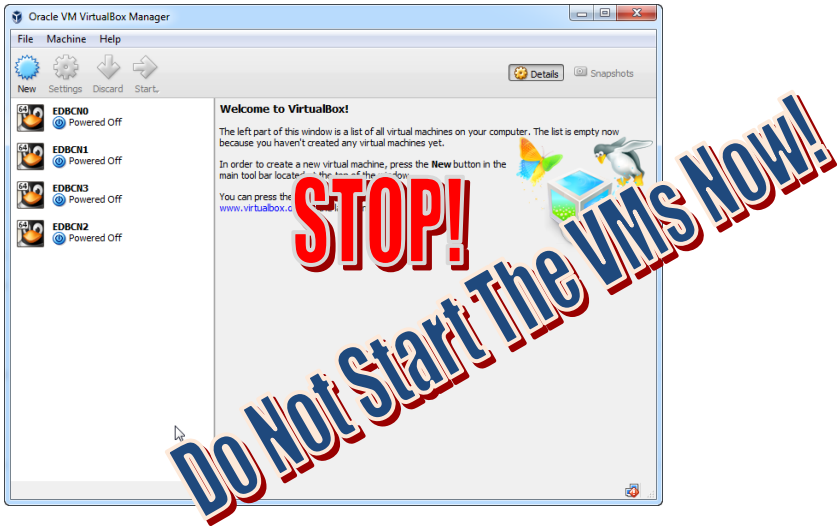


Step 2**: Apply Settings.* ***<u>Check Reset the MAC address of all network cards</u>



***Step 3**: Start Import of VirtualBox Machines*

The Oracle VirtualBox machines import takes approximately 45-90 minute. Do not interrupt or run other applications while the import is in-progress. When the import is complete, the Oracle VirtualBox machines EDBCN0, EDBCN1, EDBCN2 and EDBCN3 are imported. Please, do not start the VirtualBox machines yet.



After import completed successfully. The next step is to configure the Oracle VirtualBox machines memory settings and networking.

VirtualBox Machines Memory Settings

- Change the memory setting for Oracle VirtualBox machines for EDBC0, EDBC1 and EDBC2 according the physical RAM size.

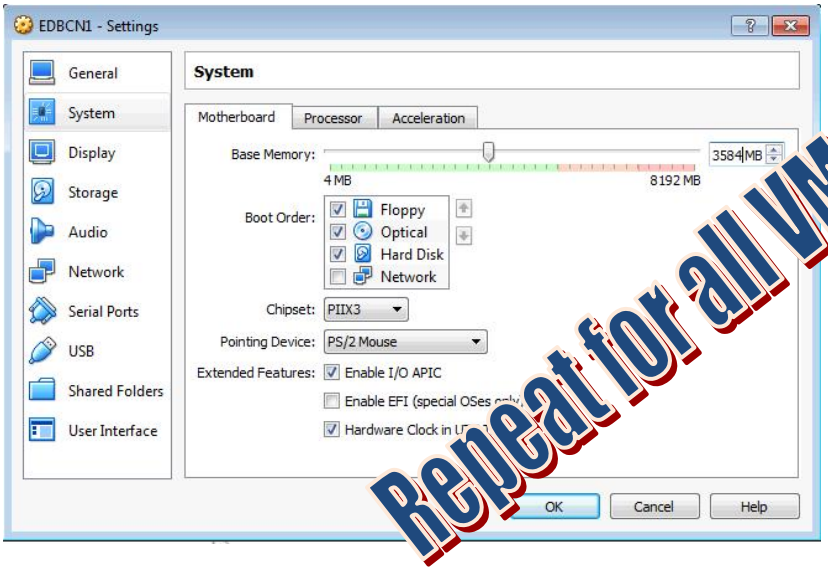
  - For laptop machines with 8GB RAM, start VirtualBox EDBC0, EDBC1 only, (2-Node EDBC), resize the VirtualBox machine memory as shown below:

Operating System	Memory Size (GB)
Host MS Windows OS	1GB (1024MB)
EDBC0	3.5GB (3584MB)
EDBC1	3.5GB (3584MB)

  - For laptop machines with 12-16GB RAM, start VirtualBox EDBC0, EDBC1 and EDBC2 only (3-Node EDBC), resize the VirtualBox machine memory as shown below:

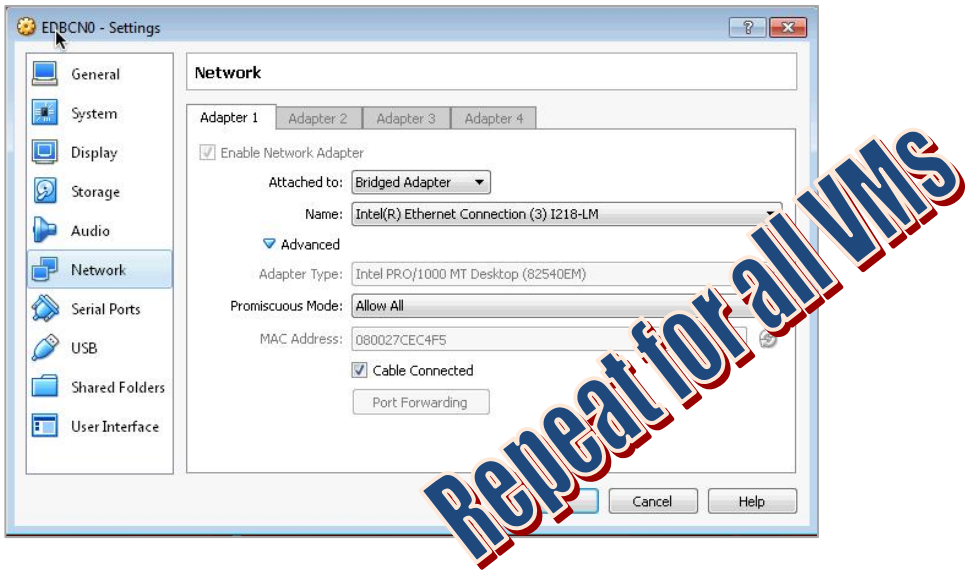
Operating System	Memory Size (12GB)	Memory Size (16GB)
Host MS Windows OS	1.5GB (1536MB)	2GB (2048MB)
EDBC0	3.5GB (3584MB)	5GB (5120MB)
EDBC1	3.5GB (3584MB)	5GB (5120MB)
EDBC2	3.5GB (3584MB)	4GB (4096MB)

The next step shows you how to re-size memory and perform setting for the cluster network. Click on Settings, select System and change the Base Memory accoring to the recommended values.



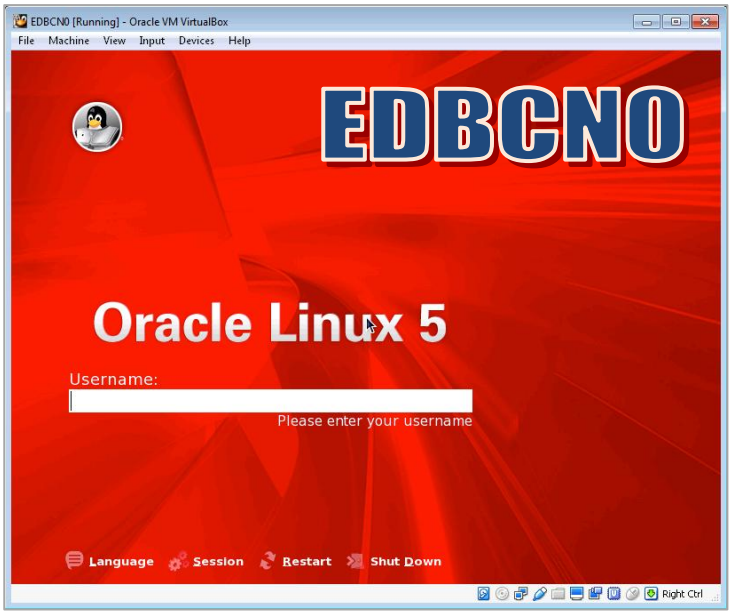
VirtualBox Machines Network Settings

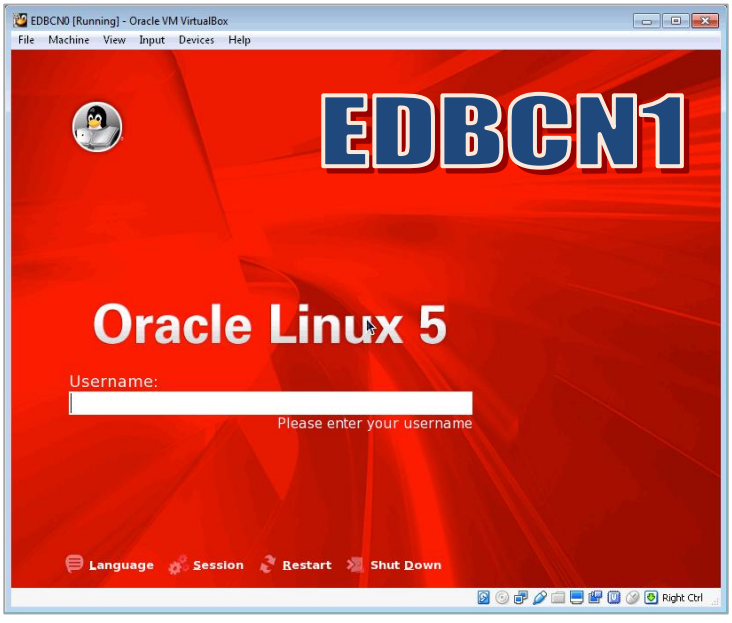
Select the following options: Attached to: Bridged Adapter, Name Intel Wireless, Allow All, Advanced ▸ Cable Connected (Unchecked)

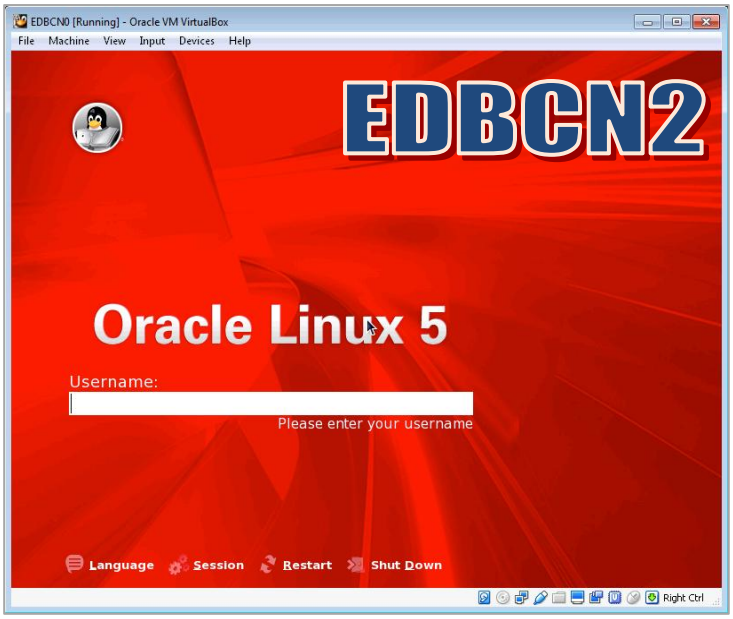


Start VirutalBox Console

Start the Oracle VirtualBox Console for EDBCN0, EDBCN1, and EDBCN2. Console Login as 'root', password 'oracle'.

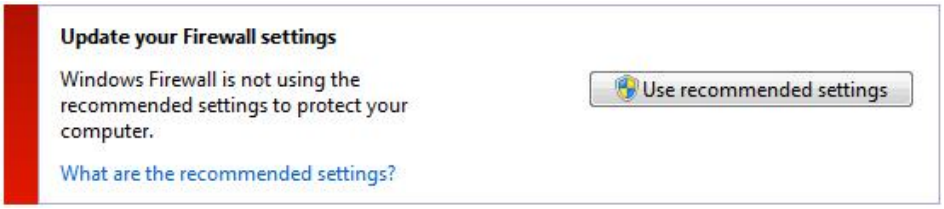






Task 2: Setting Guest Machine, the Primary node, Extended Nodes Static IP Addresses

- Turn off MS Windows Firewall. Navigate to: Control Panel ▸ All Control Panel Items ▸ Windows Firewall



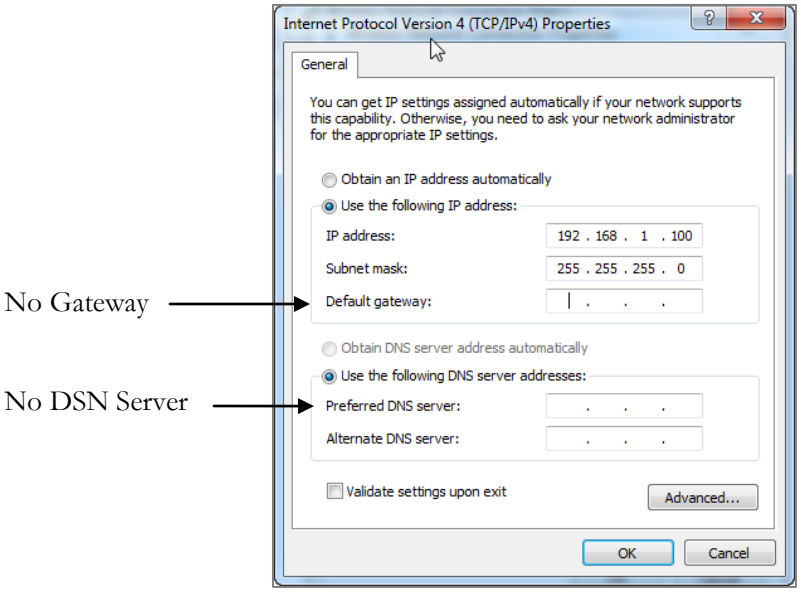
- Connect to EDBC Wireless network

Candidate No	Wireless Network Name	Password (if required)
1-5	EDBC-WIRELESS1	edbc_12c
6-10	EDBC-WIRELESS2	edbc_12c

- Assign guest machine (local host) IP/Address as shown in Table 2.

Candidate No	IP Address	Subnet
1	192.168.1.91	255.255.255.0
2	192.168.1.92	255.255.255.0
3	192.168.1.93	255.255.255.0
4	192.168.1.94	255.255.255.0
5	192.168.1.95	255.255.255.0
6	192.168.1.96	255.255.255.0
7	192.168.1.97	255.255.255.0
8	192.168.1.98	255.255.255.0
9	192.168.1.99	255.255.255.0
10	192.168.1.100	255.255.255.0

Table 2: Guest Machine IP address



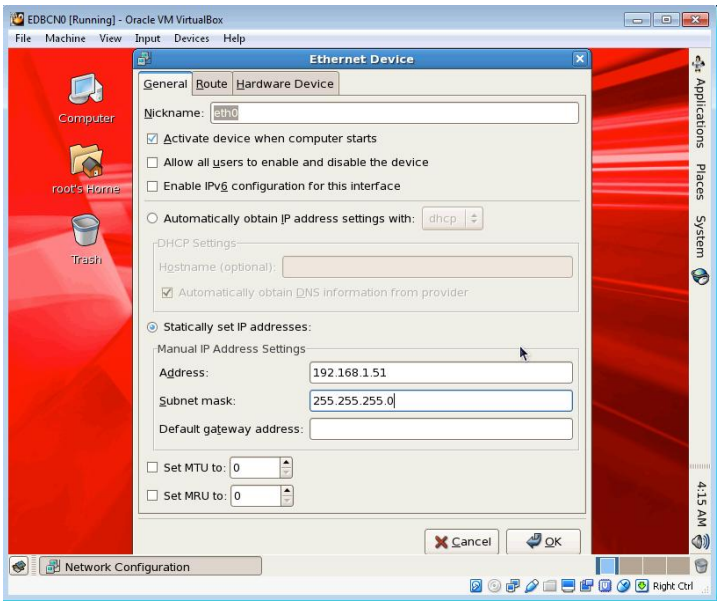
- ☐ Check the IP address for the primary and extended nodes. The format of the TCP/IP address as shown in Table 3.

Select	Extended Node IP Address	Extended Node Hostname	Downstream Hostname	Downstream Systems IP Address
C1	192.168.1.51	edbcn0.oracle.com	c1ds1.oracle.com	192.168.1.31
☐	192.168.1.61	edbcn1.oracle.com	c1ds2.oracle.com	192.168.1.41

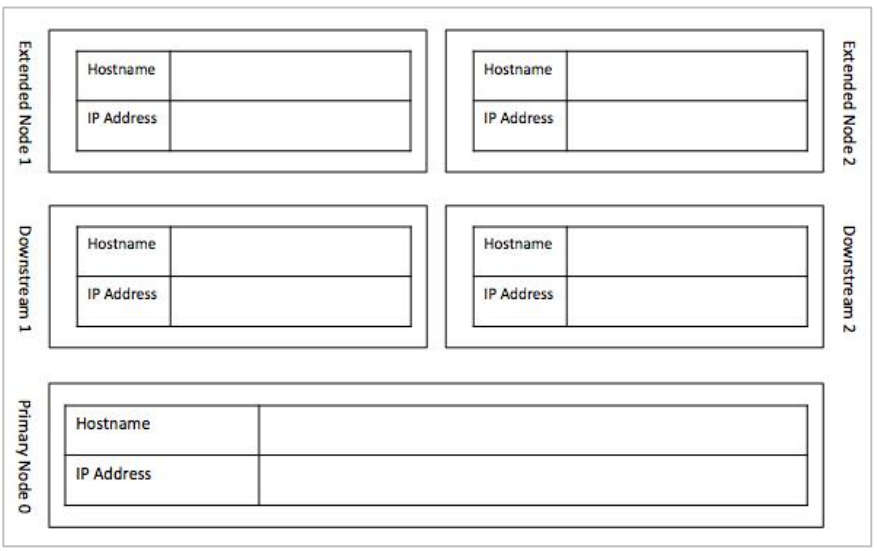
	192.168.1.71 192.168.1.81	edbcn2.oracle.com edbcn3.oracle.com		
C2 □	192.168.1.52 192.168.1.62 192.168.1.72 192.168.1.82	edbcn0.oracle.com edbcn1.oracle.com edbcn2.oracle.com edbcn3.oracle.com	c2ds1.oracle.com c2ds2.oracle.com	192.168.1.32 192.168.1.42
C3 □	192.168.1.53 192.168.1.63 192.168.1.73 192.168.1.83	edbcn0.oracle.com edbcn1.oracle.com edbcn2.oracle.com	c3ds1.oracle.com c3ds2.oracle.com	192.168.1.33 192.168.1.43
C4 □	192.168.1.54 192.168.1.64 192.168.1.74 192.168.1.84	edbcn0.oracle.com edbcn1.oracle.com edbcn2.oracle.com edbcn3.oracle.com	c4ds1.oracle.com c4ds2.oracle.com	192.168.1.34 192.168.1.44
C5 □	192.168.1.55 192.168.1.65 192.168.1.75 192.168.1.85	edbcn0.oracle.com edbcn1.oracle.com edbcn2.oracle.com edbcn3.oracle.com	c5ds1.oracle.com c5ds2.oracle.com	192.168.1.35 192.168.1.45
C6 □	192.168.1.56 192.168.1.66 192.168.1.76 192.168.1.86	edbcn0.oracle.com edbcn1.oracle.com edbcn2.oracle.com edbcn3.oracle.com	c6ds1.oracle.com c6ds2.oracle.com	192.168.1.36 192.168.1.46
C7 □	192.168.1.57 192.168.1.67 192.168.1.77 192.168.1.87	edbcn0.oracle.com edbcn1.oracle.com edbcn2.oracle.com edbcn3.oracle.com	c7ds1.oracle.com c7ds2.oracle.com	192.168.1.37 192.168.1.47
C8 □	192.168.1.58 192.168.1.68 192.168.1.78 192.168.1.88	edbcn0.oracle.com edbcn1.oracle.com edbcn2.oracle.com edbcn3.oracle.com	c8ds1.oracle.com c8ds2.oracle.com	192.168.1.38 192.168.1.48
C9 □	192.168.1.59 192.168.1.69 192.168.1.79 192.168.1.89	edbcn0.oracle.com edbcn1.oracle.com edbcn2.oracle.com edbcn3.oracle.com	c9ds1.oracle.com c9ds2.oracle.com	192.168.1.39 192.168.1.49
C10 □	192.168.1.60 192.168.1.70 192.168.1.80 192.168.1.90	edbcn0.oracle.com edbcn1.oracle.com edbcn2.oracle.com edbcn3.oracle.com	c10ds1.oracle.com c10ds2.oracle.com	192.168.1.40 192.168.1.50

Table 3: Candidates EDBC IP addresses

- Set the TCP/IP Address for EDBC0, EDBC1 and EDBC2 Oracle VirtualBox machines. Navigate System ▸ Network and change the IP address as per the instruction.



- Complete the diagram; enter the hostname and IP address for Downstream 1, Downstream 2, Primary Node 0, Extended Node 1 and Extended Node 2.



Configure Linux Local DNS /etc/hosts

Update the local Linux DNS file /etc/hosts using the above diagram.

The local DNS host file /etc/hosts lists the TCP/IP address, hostname and the short host name. The following is an example from system for candidate 1 (C1).

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1               localhost.localdomain localhost
::1             localhost6.localdomain6 localhost6
192.168.1.51    edbcn0.oracle.com       edbcn0
192.168.1.61    edbcn1.oracle.com       edbcn1
192.168.1.71    edbcn2.oracle.com       edbcn2
192.168.1.31    c1ds1.oracle.com        c1ds1
192.168.1.41    c1ds2.oracle.com        c1ds2
```

Using EDBC0 console (Primary node), verify connectivity to your extended nodes and downstream systems are reachable. Do not use the IP Address, you must use the hostname.

```
[root@edbcn0 ~\# ping edbcn1

[root@edbcn0 ~\# ping edbcn2

[root@edbcn0 ~\# ping c<n>ds1

[root@edbcn0 ~\# ping c<n>ds2
```

Ping Command	Command Successful?
ping edbcn1	☐
ping edbcn2	☐
ping c<n>ds1	☐
ping c<n>ds2	☐

Where n=1..10, Login details for EDBCN0:

Username: root
Password: oracle

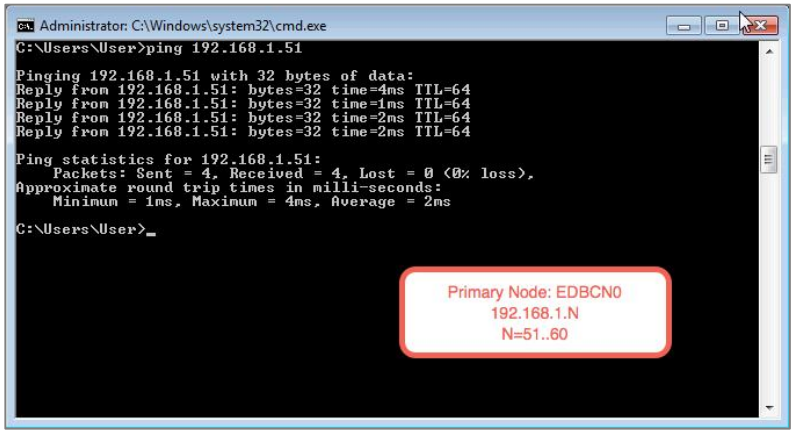
Validate Networking

Ping the following Oracle VirtualBox machines from MS Windows command-line (cmd). Use the IP address assigned to you in Table 3.

IP Address	Description	Successful Ping?
192.168.1.<n>, n=51..60	The primary node (Node 0)	☐
192.168.1.<n>, n=61..70	Extended node 1	☐
192.168.1.<n>, n=71..80	Extended node 2	☐
192.168.1.<n>, n=31..40	Downstream system 1	☐
192.168.1.<n>, n=41..50	Downstream system 2	☐

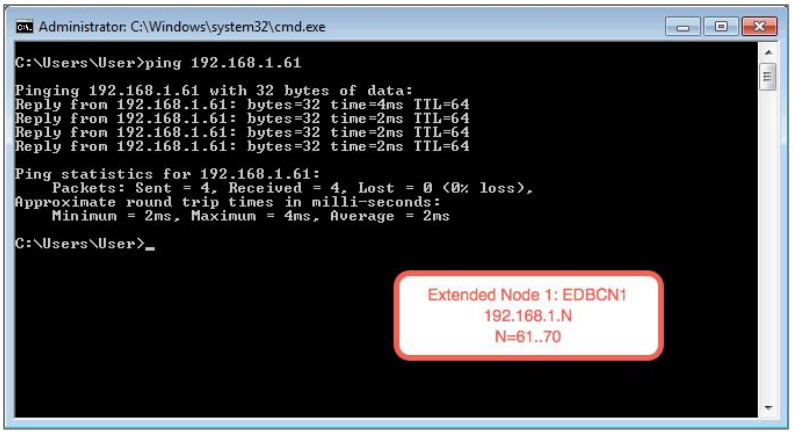
Primary Node

192.168.1.<n>
N=51..60



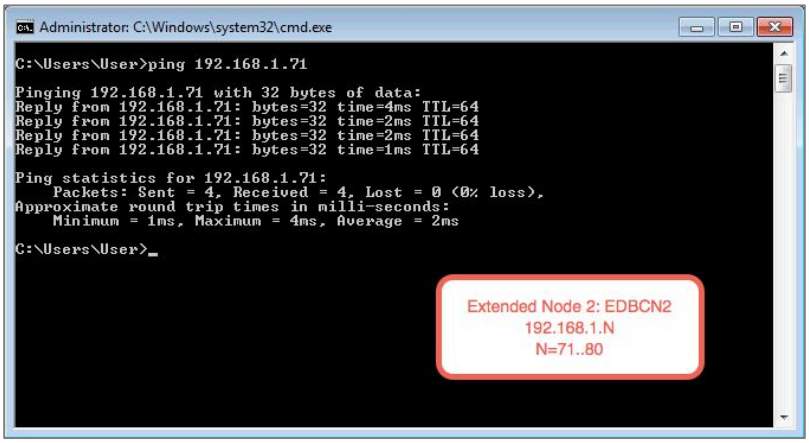
Extended node 1

192.168.1.<n>
N=61..70



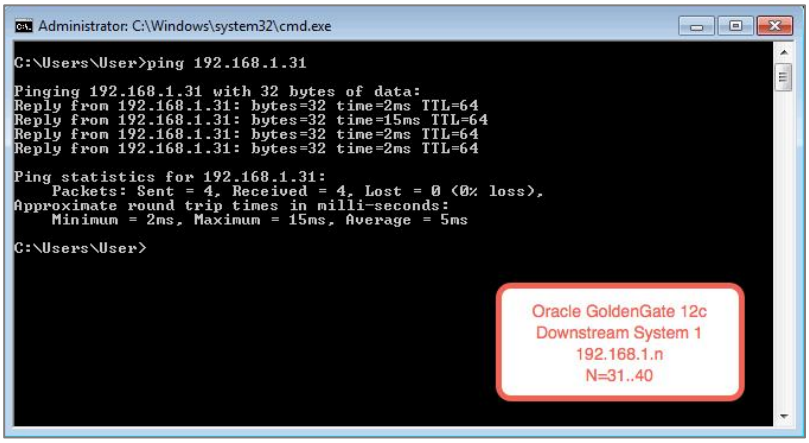
Extended node 2

192.168.1.<n>
N=71..80



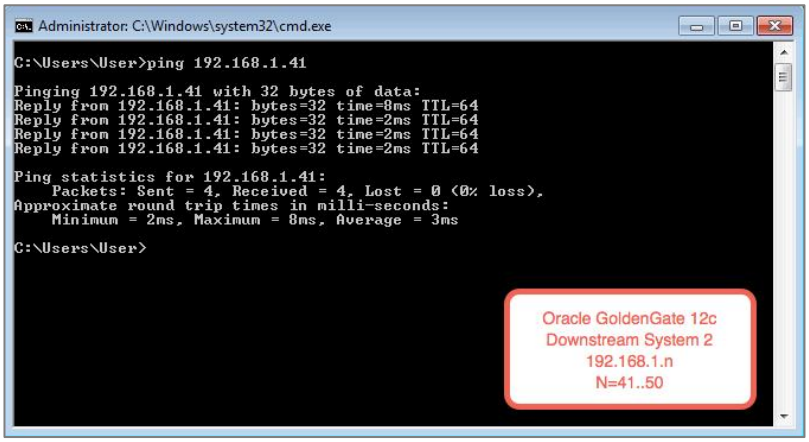
Downstream System 1

192.168.1.<n>
N=31..40



Downstream System 2

192.168.1.<n>
N=41..50



Task 3: Perform EDBC Systems Login

Use the below credentials to Login to the primary node, extended nodes and downstream systems.

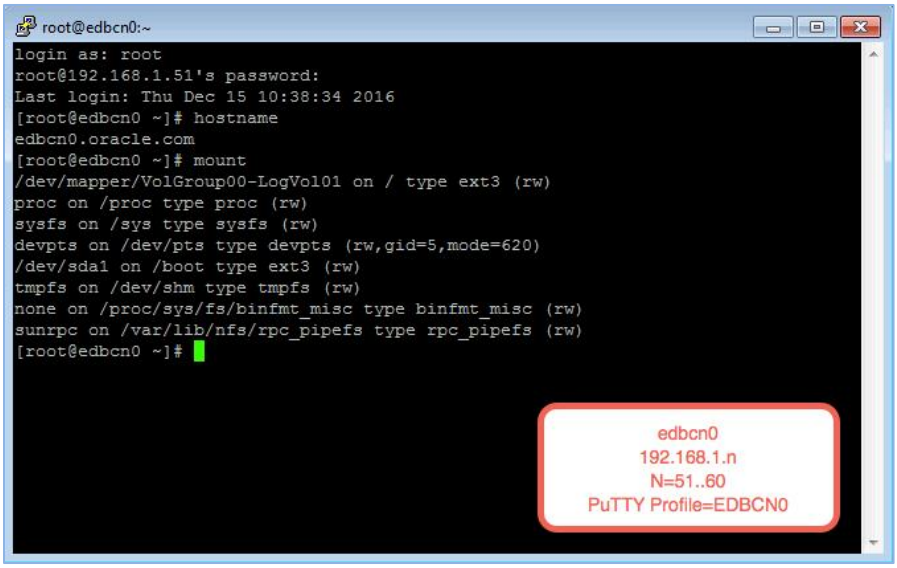
Username: root
Password: oracle, for downstream oracle_c<n>

- From the host machine (MS Windows), start PuTTY terminal for the following systems:

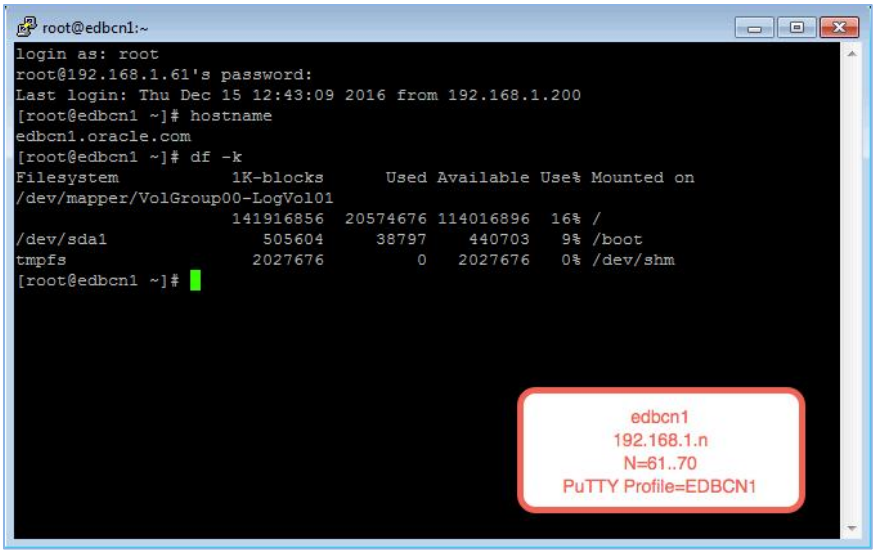
  1. Primary node system – edbcn0.oracle.com
  2. Extended node 1: edbcn1.oracle.com
  3. Extended node 2: edbcn2.oracle.com
  4. Downstream system: c<n>ds1.oracle.com
  5. Downstream system: c<n>ds2.oracle.com

  Where <n> is the candidate number assigned to you; n=1..10 only.

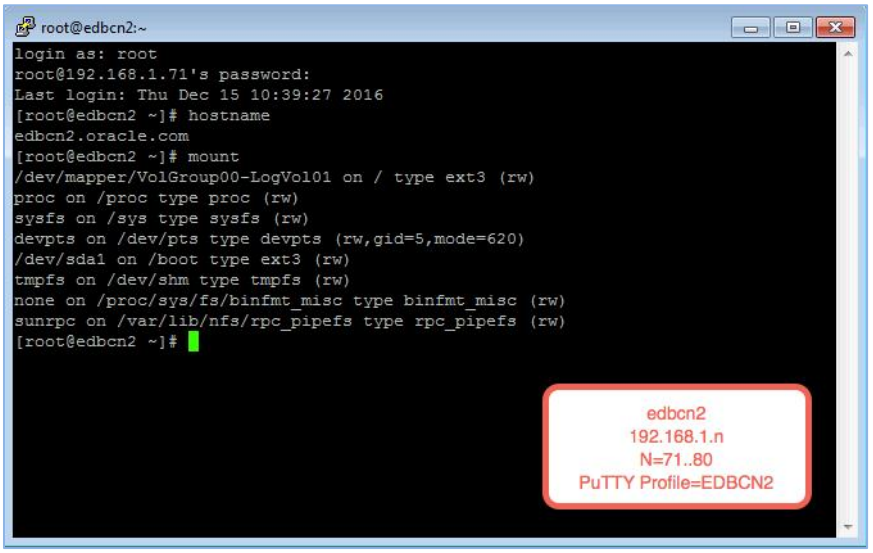
  Store the Linux PuTTY profile to each system: edbc0, edbcn1, edbc2, c<n>ds1 and c<n>ds2.



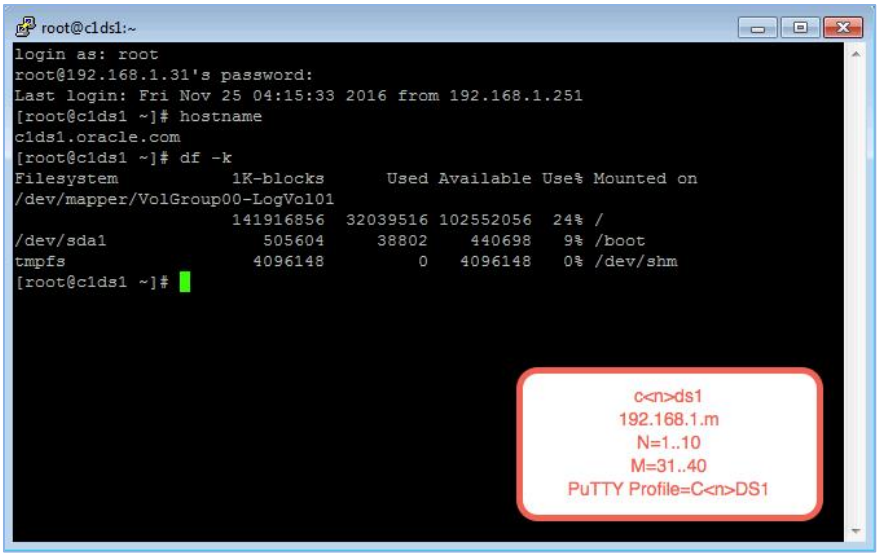
Oracle VirtualBox Name	EDBCN0
Unix Hostname	edbcn0.oracle.com
IP Address	192.168.1.<n>, 51-60



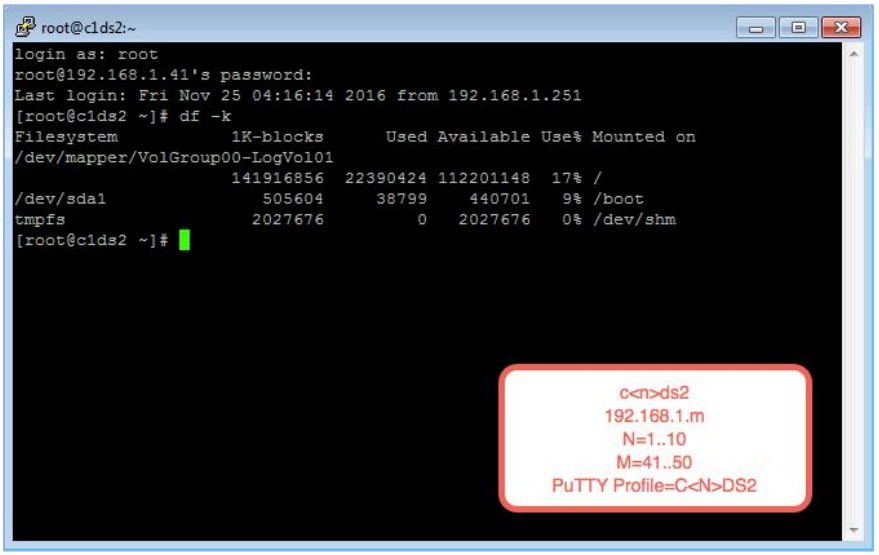
Oracle VirtualBox Name	EDBCN1
Unix Hostname	edbcn1.oracle.com
IP Address	192.168.1.<n>, 61-70



Oracle VirtualBox Name	EDBCN2
Unix Hostname	edbcn2.oracle.com
IP Address	192.168.1.<n>, 71-80



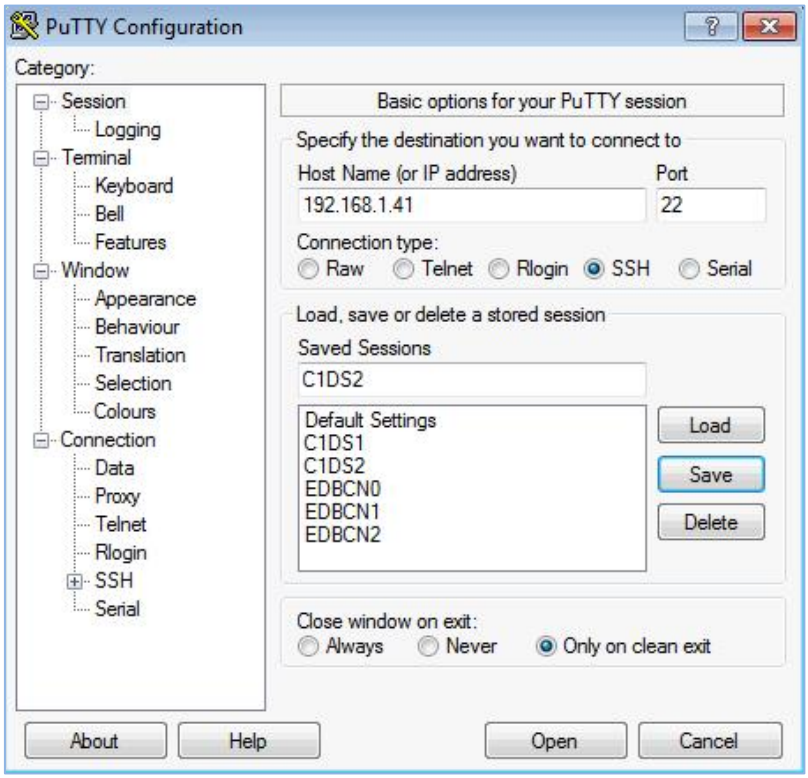
Oracle VirtualBox Name	C<n>DS1, n=1-10
Unix Hostname	c<n>ds1.oracle.com, n=1-10
IP Address	192.168.1.<n>, 31-40



Oracle VirtualBox Name	C<n>DS2, n=1-10
Unix Hostname	c<n>ds2.oracle.com, n=1-10
IP Address	192.168.1.<n>, 41-50

PuTTY Profiles

Create PuTTY profiles names as shown below.



If you need to assign the terminals descriptive names, pull-down PuTTY menu ▸ Change settings ▸ Windows ▸ Behavior, enter the Window Title and click Apply.

Profile	Window Title
C<n>DS1	Downstream System 1
C<n>DS2	Downstream System 2
EDBCN0	Primary Node (Node 0)
EDBCN1	Extended Node 1
EDBCN2	Extended Node 2

Workshop 2: EDBC Environment Review

An Introduction

The purpose of this workshop is to confirm your understanding of the EDBC components. An EDBC composed of one-and-only-one primary node, and one-or-more-extended node(s). The sequence of deploying EDBC systems is performed in the following order:

1. The primary node, to extended node 1
2. Extended node 1 to the primary node
3. Extended node 2 to the primary node

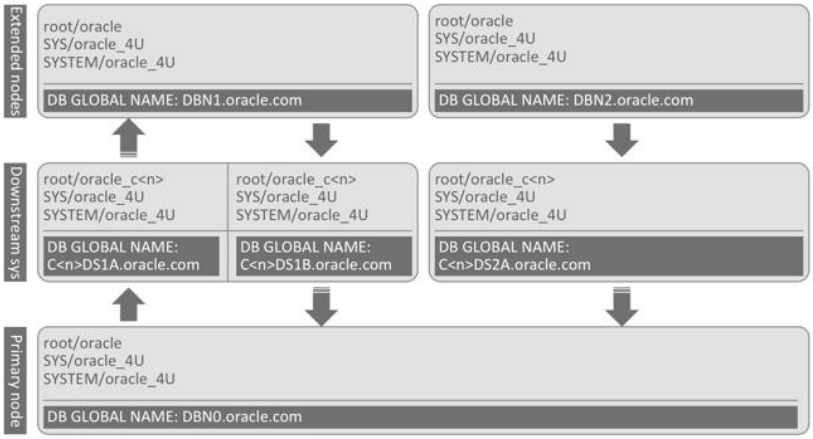


Figure 6: Hostnames, Global Names and Login Credentials

Table 4 lists the login credentials for your EDBC. Figure 6 depicts the same details in a diagram format. The EDBC is extended by adding extended nodes and the associated downstream systems from real-time change data capture and data streaming.

Privileged User	Password
Primary super user (root)	oracle
Extended super user (root)	oracle
Primary software owner (oracle)	oracle
Extended software owner (oracle)	oracle
Downstream System (root)	oracle_c<n>
Database SYS	oracle_4U
Database SYSTEM	oracle_4U

Table 4: Login Credentials

Where n is the candidate #; Table 5 lists Workshop 2 tasks.

Task No	Task Description
1	Start/Restart the database listener & instances
2	Verify TNS connect strings
3	Review Downstream systems C<n>DS1A, C<n>DS1B and C<n>DS2A
4	Review primary node EDBCN0
5	Review extended nodes EDBCN1 and EDBCN2

Table 5: Workshop 3 Tasks

Login to each of the Oracle GoldenGate 12c downstream mining database, the primary node and extended nodes and complete, verify the systems configuration.

Task 1: Restart the Database Listener and Instances

Each database instance is statically registered with its database listener. Verify the listener has the database service name is registered.

- ☐ Start/Restart the primary node (DBN0) listener & database

```
[oracle@edbcn0 ~]$ lsnrctl

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 18-DEC-2016 04:55:46

Copyright (c) 1991, 2014, Oracle.  All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> start
Starting /u01/app/oracle/product/12.1.0/dbhome 1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.2.0 - Production
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
Log messages written to
/u01/app/oracle/diag/tnslsnr/edbcn0/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn0.oracle.com)(PORT=1521)))

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edbcn0.oracle.com)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.2.0 - Production
Start Date                18-DEC-2016 04:55:48
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
```

```
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/edbcn0/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn0.oracle.com)(PORT=1521)))
Services Summary...
Service "DBN0.oracle.com" has 1 instance(s).
  Instance "DBN0", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
LSNRCTL>
```

Starting the primary node database DBN0…

```
[oracle@edbcn0 ~]$ . oraenv
ORACLE_SID = [DBN0] ?
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@edbcn0 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sun Dec 18 04:31:43 2016

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

SQL> conn / as sysdba
Connected to an idle instance.
SQL> startup
ORACLE instance started.

Total System Global Area          1660944384 bytes
Fixed Size                           2925072 bytes
Variable Size                     1056968176 bytes
Database Buffers                   587202560 bytes
Redo Buffers                        13848576 bytes
Database mounted.
Database opened.
SQL>
```

- Start/Restart extended node 1 listener & database

```
[oracle@edbcn1 ~]$ lsnrctl

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 18-DEC-2016 05:03:11

Copyright (c) 1991, 2014, Oracle.  All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> start
Starting /u01/app/oracle/product/12.1.0/dbhome_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.2.0 - Production
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
```

```
Log messages written to
/u01/app/oracle/diag/tnslsnr/edbcn1/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn1.oracle.com)(PORT=1521)))

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edbcn1.oracle.com)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.2.0 - Production
Start Date                18-DEC-2016 05:03:14
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/edbcn1/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn1.oracle.com)(PORT=1521)))
Services Summary...
Service "DBN1.oracle.com" has 1 instance(s).
  Instance "DBN1", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
LSNRCTL>
```

Starting extended node 1 database DBN1…

```
[oracle@edbcn1 ~]$ . oraenv
ORACLE_SID = [DBN1] ?
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@edbcn1 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sun Dec 18 04:35:16 2016

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

SQL> conn / as sysdba
Connected to an idle instance.
SQL> startup


ORACLE instance started.

Total System Global Area          1660944384 bytes
Fixed Size                           2925072 bytes
Variable Size                     1056968176 bytes
Database Buffers                   587202560 bytes
Redo Buffers                        13848576 bytes
Database mounted.
Database opened.
SQL>
```

- Start/Restart extended node 2 listener & database

```
[oracle@edbcn2 ~]$ lsnrctl

LSNRCTL for Linux: Version 11.2.0.4.0 - Production on 18-DEC-2016 05:08:58

Copyright (c) 1991, 2013, Oracle.  All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> start
Starting /u01/app/oracle/product/11.2.0/dbhome_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 11.2.0.4.0 - Production
System parameter file is
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/listener.ora
Log messages written to
/u01/app/oracle/diag/tnslsnr/edbcn2/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn2.oracle.com)(PORT=1521)))
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edbcn2.oracle.com)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 11.2.0.4.0 - Production
Start Date                18-DEC-2016 05:08:59
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/edbcn2/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn2.oracle.com)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
Services Summary...
Service "DBN2.oracle.com" has 1 instance(s).
  Instance "DBN2", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
LSNRCTL>
```

Starting extended node 2 database DBN2…

```
[oracle@edbcn1 ~]$ . oraenv
ORACLE SID = [DBN1] ?
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@edbcn1 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sun Dec 18 04:35:16 2016

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

SQL> conn / as sysdba
```

```
Connected to an idle instance.
SQL> startup


ORACLE instance started.

Total System Global Area          1660944384 bytes
Fixed Size                           2925072 bytes
Variable Size                     1056968176 bytes
Database Buffers                   587202560 bytes
Redo Buffers                        13848576 bytes
Database mounted.
Database opened.
SQL>
```

- Start/Restart downstream database mining system 1 listener & database

```
[oracle@c<n>ds1 ~]$ lsnrctl

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 18-DEC-2016 05:14:56

Copyright (c) 1991, 2014, Oracle.  All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> start
Starting /u01/app/oracle/product/12.1.0/dbhome_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.2.0 - Production
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome 1/network/admin/listener.ora
Log messages written to
/u01/app/oracle/diag/tnslsnr/c1ds1/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=c1ds1.oracle.com)(PORT=1521)))

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=c1ds1.oracle.com)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.2.0 - Production
Start Date                18-DEC-2016 05:14:58
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome 1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/c1ds1/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=c1ds1.oracle.com)(PORT=1521)))
Services Summary...
Service "C<n>DS1A.oracle.com" has 1 instance(s).
  Instance "C1DS1A", status UNKNOWN, has 1 handler(s) for this service...
Service "C<n>DS1B.oracle.com" has 1 instance(s).
```

```
  Instance "C<n>DS1B", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
LSNRCTL>
```

Starting downstream mining database C<n>DS1A…

```
[oracle@c<n>ds1 ~]$ . oraenv
ORACLE_SID = [C<n>DS1A] ?
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@c1ds1 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sun Dec 18 05:16:32 2016

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

SQL> connect / as sysdba
Connected to an idle instance.
SQL> startup
ORACLE instance started.

Total System Global Area          1660944384 bytes
Fixed Size                           2925072 bytes
Variable Size                     1056968176 bytes
Database Buffers                   587202560 bytes
Redo Buffers                        13848576 bytes
Database mounted.
Database opened.
SQL>
```

Starting downstream mining database C<n>DS1B…

```
[oracle@c<n>ds1 ~]$ . oraenv
ORACLE_SID = [C<n>DS1A] ? C<n>DS1B
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@c1ds1 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sun Dec 18 05:18:17 2016

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

SQL> conn / as sysdba
Connected to an idle instance.
SQL> startup
ORACLE instance started.

Total System Global Area          1660944384 bytes
Fixed Size                           2925072 bytes
Variable Size                     1056968176 bytes
Database Buffers                   587202560 bytes
Redo Buffers                        13848576 bytes
Database mounted.
Database opened.
SQL>
```

- Start/Restart downstream database mining system 2 listener & database

```
[oracle@c<n>ds2 ~]$ lsnrctl

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 18-DEC-2016 05:20:11

Copyright (c) 1991, 2014, Oracle.  All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> start
Starting /u01/app/oracle/product/12.1.0/dbhome_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.2.0 - Production
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome 1/network/admin/listener.ora
Log messages written to
/u01/app/oracle/diag/tnslsnr/c1ds2/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=c1ds2.oracle.com)(PORT=1521)))

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=c1ds2.oracle.com)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.2.0 - Production
Start Date                18-DEC-2016 05:20:13
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome 1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/c1ds2/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=c1ds2.oracle.com)(PORT=1521)))
Services Summary...
Service "C<n>DS2A" has 1 instance(s).
  Instance "C<n>DS2A", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
LSNRCTL>
```

Starting downstream mining database C<n>DS2A…

```
[oracle@c<n>ds2 ~]$ . oraenv
ORACLE_SID = [C<n>DS2A] ?
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@c1ds2 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sun Dec 18 05:21:11 2016

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

SQL> conn / as sysdba
Connected to an idle instance.
```

```
SQL> startup
ORACLE instance started.

Total System Global Area          1660944384 bytes
Fixed Size                           2925072 bytes
Variable Size                     1056968176 bytes
Database Buffers                   587202560 bytes
Redo Buffers                        13848576 bytes
Database mounted.
Database opened.
SQL>
```

Task 2: Verify TNS Database Connect Strings

Because EDBC Redo-Transport and integrated extract (IE) use explicit database connect string, verify the database connect strings across the cluster are successfully connected. Table 6 lists the source and target for the connect string.

Source (FROM)	Target (TO)	Connect String	TNS Success?
Primary Node	Primary Node	DBN0.oracle.com	☐
Extended Node 1	Extended Node 1	DBN1.oracle.com	☐
Extended Node 2	Extended Node 2	DBN2.oracle.com	☐
Downstream 1A	Downstream 1A	C<n>DS1A.oracle.com	☐
Downstream 1B	Downstream 1B	C<n>DS1B.oracle.com	☐
Primary Node	Downstream 1A	C<n>DS1A.oracle.com	☐
Downstream 1A	Extended Node 1	DBN1.oracle.com	☐
Extended Node 1	Downstream 1B	C<n>DS1B.oracle.com	☐
Downstream 1B	Primary Node	DBN0.oracle.com	☐
Extended Node 2	Downstream 2A	C<n>DS2A.oracle.com	☐
Downstream 2A	Primary Node	DBN0.oracle.com	☐
Downstream 1A	Extended Node 2	DBN2.oracle.com	☐
Downstream 1B	Extended Node 2	DBN1.oracle.com	☐
Downstream 2A	Extended Node 1	DBN2.oracle.com	☐

Table 6: Database Connect Strings

- ☐ Verify TNS from Primary node to Primary node (Loopback)

```
[oracle@edbcn0 ~]$ tnsping dbn0.oracle.com

TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 19-DEC-2016
15:15:09

Copyright (c) 1997, 2014, Oracle.  All rights reserved.

Used parameter files:
```

```
Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = edbcn0.oracle.com)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
DBN0.oracle.com)))
OK (0 msec)
[oracle@edbcn0 ~]$
```

```
SQL> conn system/oracle_4U@dbn0.oracle.com
Connected.
SQL>
```

- Verify TNS from Extended Node 1 to Extended Node 1 (Loopback)

```
[oracle@edbcn1 ~]$ tnsping dbn1.oracle.com

TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 20-DEC-2016
00:03:24

Copyright (c) 1997, 2014, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = edbcn1.oracle.com)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
DBN1.oracle.com)))
OK (0 msec)
```

```
SQL> conn system/oracle_4U@dbn1.oracle.com
Connected.
SQL>
```

- Verify TNS from Extended Node 2 to Extended Node 2 (Loopback)

```
[oracle@edbcn2 ~]$ tnsping dbn2.oracle.com

TNS Ping Utility for Linux: Version 11.2.0.4.0 - Production on 20-DEC-2016
00:08:17

Copyright (c) 1997, 2013, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = edbcn2.oracle.com)(PORT = 1521))) (CONNECT DATA = (SERVICE NAME =
dbn2.oracle.com)))
OK (0 msec)
```

```
SQL> conn system/oracle_4U@dbn2.oracle.com
Connected.
```

```
SQL>
```

- Verify TNS from Downstream 1A to Downstream 1A (Loopback)

```
[oracle@c1ds1 ~]$ tnsping c1ds1a.oracle.com

TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 20-DEC-2016
00:12:22

Copyright (c) 1997, 2014, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = c<n>ds1.oracle.com)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
C<n>DS1A.oracle.com)))
OK (20 msec)
```

```
SQL> conn system/oracle_4U@c<n>ds1a.oracle.com
Connected.
SQL>
```

- Verify TNS from Downstream 1B to Downstream 1B (Loopback)

```
[oracle@c1ds1 ~]$ tnsping c<n>ds1b.oracle.com

TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 20-DEC-2016
00:13:26

Copyright (c) 1997, 2014, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = c<n>ds1.oracle.com)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
C<n>DS1B.oracle.com)))
OK (0 msec)
```

```
SQL> conn system/oracle_4U@c<n>ds1b.oracle.com
Connected.
SQL>
```

- Verify TNS from Downstream 2A to Downstream 2A (Loopback)

```
[oracle@c1ds2 ~]$ tnsping c<n>ds2a.oracle.com

TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 20-DEC-2016
00:14:31
```

```
Copyright (c) 1997, 2014, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = c<n>ds2.oracle.com)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
C<n>DS2A.oracle.com)))
OK (10 msec)
```

```
SQL> conn system/oracle_4U@c<n>ds2a.oracle.com
Connected.
SQL>
```

- Verify TNS from Primary node to Downstream 1A

```
[oracle@edbcn0 admin]$ tnsping c<n>ds1a.oracle.com

TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 20-DEC-2016
00:19:27

Copyright (c) 1997, 2014, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = c<n>ds1.oracle.com)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
C<n>DS1A.oracle.com)))
OK (0 msec)
```

```
SQL> conn system/oracle_4U@c<n>ds1a.oracle.com
Connected.
SQL>
```

- Verify TNS from Downstream 1A to Extended Node 1

```
[oracle@c<n>ds1 ~]$ tnsping dbn1.oracle.com

TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 20-DEC-2016
00:24:37

Copyright (c) 1997, 2014, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = edbcn1.oracle.com)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
DBN1.oracle.com)))
OK (0 msec)
```

```
SQL> conn system/oracle_4U@dbn1.oracle.com
Connected.
SQL>
```

- Verify TNS from Extended Node 1 to Downstream 1B

```
[oracle@edbcn1 ~]$ tnsping c<n>ds1b.oracle.com

TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 20-DEC-2016
00:30:19

Copyright (c) 1997, 2014, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = c<n>ds1.oracle.com)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
C<n>DS1B.oracle.com)))
OK (10 msec)
```

```
SQL> conn system/oracle_4U@c1ds1b.oracle.com
Connected.
SQL>
```

- Verify TNS from Downstream 1B to Primary

```
[oracle@c<n>ds2 ~]$ tnsping dbn0.oracle.com

TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 20-DEC-2016
00:35:12

Copyright (c) 1997, 2014, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = edbcn0.oracle.com)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
DBN0.oracle.com)))
OK (0 msec)
```

```
SQL> conn system/oracle_4U@dbn0.oracle.com
Connected.
SQL>
```

- Verify TNS from Extended Node 2 to Downstream 2A

```
[oracle@edbcn2 ~]$ tnsping c1ds2a.oracle.com
```

```
TNS Ping Utility for Linux: Version 11.2.0.4.0 - Production on 20-DEC-2016
01:06:01

Copyright (c) 1997, 2013, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = c<n>ds2.oracle.com)(PORT = 1521))) (CONNECT DATA = (SERVICE NAME =
C<n>DS2A.oracle.com)))
OK (0 msec)
```

```
SQL> conn system/oracle_4U@c<n>ds2a.oracle.com
Connected.
SQL>
```

- Verify TNS from Downstream 2A to Primary

```
[oracle@c<n>ds2 admin]$ tnsping dbn0.oracle.com

TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 20-DEC-2016
01:08:32

Copyright (c) 1997, 2014, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = edbcn0.oracle.com)(PORT = 1521))) (CONNECT DATA = (SERVICE NAME =
DBN0.oracle.com)))
OK (0 msec)
```

```
SQL> conn system/oracle_4U@dbn0.oracle.com
Connected.
SQL>
```

- Verify TNS from Downstream 1A to Extended Node 2

```
[oracle@c<n>ds1 ~]$ tnsping dbn2.oracle.com

TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 20-DEC-2016
00:26:57

Copyright (c) 1997, 2014, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
```

```
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = edbcn2.oracle.com)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
DBN2.oracle.com)))
OK (10 msec)
```

```
SQL> conn system/oracle_4U@dbn2.oracle.com
Connected.
SQL>
```

- Verify TNS from Downstream 1B to Extended Node 2

```
[oracle@c<n>ds1 ~]$ tnsping dbn2.oracle.com

TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 20-DEC-2016
00:39:18

Copyright (c) 1997, 2014, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = edbcn2.oracle.com)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
DBN2.oracle.com)))
OK (10 msec)
```

```
SQL> conn system/oracle_4U@c<n>ds1a.oracle.com
Connected.
SQL> conn system/oracle_4U@dbn2.oracle.com
Connected.
SQL>
```

- Verify TNS from Downstream 2A to Extended Node 1

```
[oracle@c<n>ds2 ~]$ tnsping dbn1.oracle.com

TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 20-DEC-2016
01:12:49

Copyright (c) 1997, 2014, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = edbcn1.oracle.com)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
DBN1.oracle.com)))
OK (0 msec)
```

```
SQL> conn system/oracle_4U@dbn1.oracle.com
Connected.
SQL>
```

Task 3: Review Downstream Database Mining Systems

The designation of the downstream systems must be considered carefully, then the association of the primary node with its downstream system, and the extended node with its downstream is must be considered carefully. This allows consistent data streaming from the downstream system to the extended nodes within the cluster. The first downstream system is divided into two downstream stream instances, instance A and instance B.

C<n>DS1A: Downstream system of the primary
C<n>DS1B: Downstream system of extended node 1

Then, associate the remaining extended nodes to downstream systems as follows:

C<n>DS12: Downstream system of extended node 2
C<n>DS1<m>: Downstream system of extended node <m>

Where; n is the candidate # and m is the downstream system #

☐ Downstream C<n>DS1A, n=1..10

Hostname		☐
IP Address		☐
Global Database Name		☐
Database Version		☐
Oracle Database Home	cd $ORACLE_HOME	☐
Oracle GoldenGate Home	cd $OGG_HOME1	☐
GoldenGate (GG) Version		☐
GG Manager Port		☐
GG Administrator	ogg_admin	☐
TNS DB Connection	ogg_admin/oracle@C<n>DS1A.oracle.com	☐
GGSCI DB Connection		☐

☐ Downstream C<n>DS1B, n=1..10

Hostname		☐
IP Address		☐
Global Database Name		☐

Database Version		☐
Oracle Database Home	$ORACLE_HOME	☐
Oracle GoldenGate Home	$OGG_HOME2	☐
GoldenGate (GG) Version		☐
GG Manager Port		☐
GG Administrator	ogg_admin	☐
TNS DB Connection	ogg_admin/oracle@C<n>DS1B.oracle.com	☐

☐ Downstream - C<n>DS2A, n=1..10

Hostname		☐
IP Address		☐
Global Database Name		☐
Database Version		☐
Oracle Database Home	$ORACLE_HOME	☐
Oracle GoldenGate Home	$OGG_HOME1	☐
GoldenGate (GG) Version		☐
GG Manager Port		☐
GG Administrator	ogg_admin	☐
TNS DB Connection	ogg_admin/oracle@C<n>DS2A.oracle.com	☐
Number of log file		☐
Size of log file (MB)		☐

Task 4: Review the Primary Node System

☐ Primary Node - edbc0.oracle.com

Hostname		☐
IP Address		☐
Global Database Name		☐
Database Version		☐
Oracle Database Home	$ORACLE_HOME	☐
Oracle GoldenGate Home	$OGG_HOME	☐
GoldenGate (GG) Version		☐

GG Manager Port		☐
GG Administrator	ogg_admin	☐
TNS DB Connection	ogg_admin/oracle@EDBCN0.oracle.com	☐
GGSCI DB Connection		☐
OSM_REPAI DB Connection	osm_repapi/oracle@ EDBCN0.oracle.com	☐
Number of log file		☐
Size of log file (MB)		☐
Database Archive Mode		☐
Database Supplemental log		☐

Task 5: Review Extended Nodes Systems

☐ Extended Node 1 - edbc1.oracle.com

Hostname		☐
IP Address		☐
Global Database Name		☐
Database Version		☐
Oracle Database Home	$ORACLE_HOME	☐
Oracle GoldenGate Home	$OGG_HOME	☐
GoldenGate (GG) Version		☐
GG Manager Port		☐
GG Administrator	ogg_admin	☐
TNS DB Connection	ogg_admin/oracle@EDBCN1.oracle.com	☐
GGSCI DB Connection		☐
OSM_REPAI DB Connection	osm_repapi/oracle@ EDBCN1.oracle.com	☐
Number of log file		☐
Size of log file (MB)		☐
Database Archive Mode		☐
Database Supplemental log		☐

☐ Extended Node 2 – edbc2.oracle.com

Hostname		☐

IP Address		☐
Global Database Name		☐
Database Version		☐
Oracle Database Home	$ORACLE_HOME	☐
Oracle GoldenGate Home	$OGG_HOME	☐
GoldenGate (GG) Version		☐
GG Manager Port		☐
GG Administrator	ogg_admin	☐
TNS DB Connection	ogg_admin/oracle@EDBCN2.oracle.com	☐
GGSCI DB Connection		☐
OSM_REPAI DB Connection	osm_repapi/oracle@ EDBCN2.oracle.com	☐
Number of log file		☐
Size of log file (MB)		☐
Database Archive Mode		☐
Database Supplemental log		☐

Workshop 3: Configuring EDBC Redo-Transport

EDBC empowers of Oracle GoldenGate 12c downstream systems. Hence the configuration of Redo-Transport from the primary node and the expended nodes to the associated downstream systems is a requirement and must be completed successfully, that the database transactions logs are shipped and consumed by the Integrated Extract (IE). Refer to Appendix E for further details.

Table 7 lists Workshop 3 tasks.

Task No	Task Description
1	Configure Redo-Transport from the primary to downstream C<n>DS1A
2	Configure Redo-Transport from extended node 1 to downstream C<n>DS1B
3	Configure Redo-Transport from extended node 2 to downstream C<n>DS2A

Table 7: Workshop 3 Tasks

There are three Redo-Transport configurations as listed above and shown in the diagram below. Figure 7 illustrate the Redo-Transport for primary node and extended nodes. The Redo-Transport pre-requisites have been versified in Workshop 2, which include:-

- ✓ Database operates in archive log mode
- ✓ Database Force Logging enable

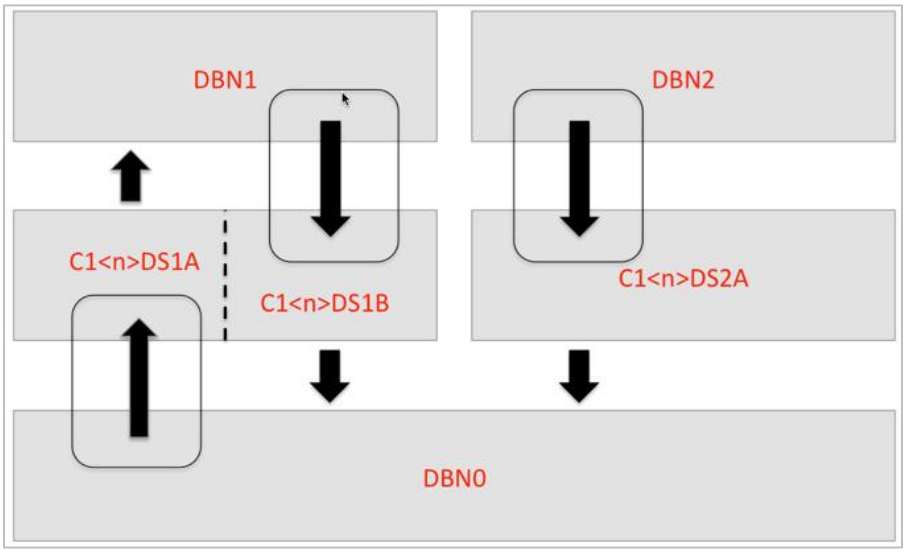


Figure 7: Redo-Transport Across the Cluster

Task 1: Configure Redo-Transport from the Primary Node to Downstream System C<n>DS1A

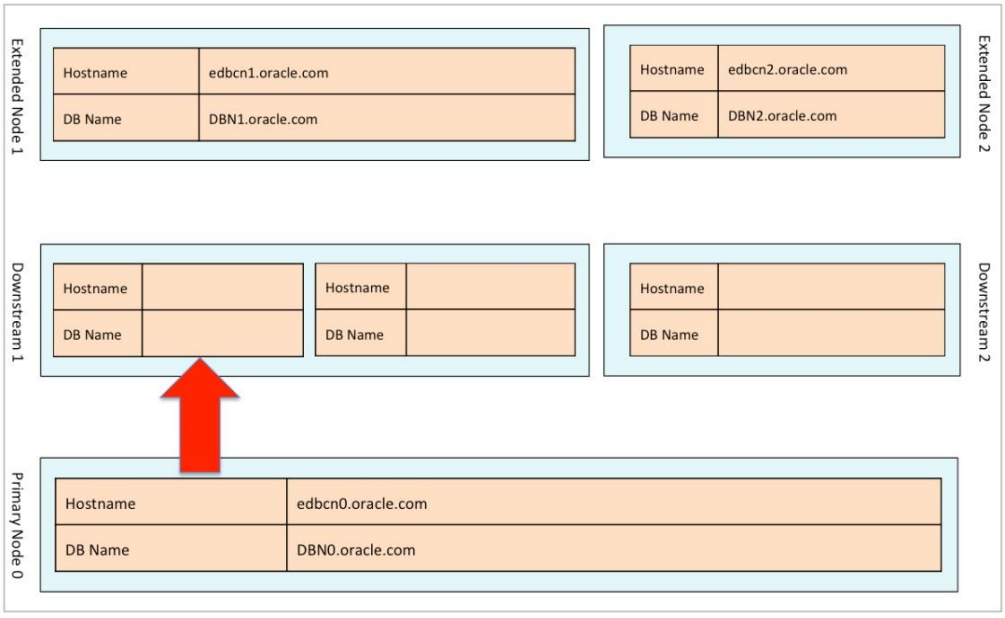
Complete the details of the primary node:

Hostname	
IP Address	
Global Database Name	
Database Unique Name	
Connect string	

Complete the details of the downstream system (C<n>DS1A) for the primary node:

Hostname	
IP Address	
Global Database Name	
Database Unique Name	
Connect string	

Complete the diagram to verify the Redo-Transport for:



- Re-create the password file on the primary node and downstream system – C<n>DS1A. The purpose of re-creating the password file is to ensure both systems have identical SYS password, otherwise, the Redo-transport will fail. The steps for re-creating the password file are online, specifying IGNORECASE requires restarting the database instance.

    1. Login to the primary node and downstream system
    2. Remove the current password file
    3. Re-create the password file on the primary node using orapwd utility
    4. Remote copy the password file from the primary node to the downstream system
    5. Rename and change permission on the downstream system
    6. Restart the database on the primary node and downstream system

Open two PuTTY terminals, execute the following steps to re-create the password successfully.

Server	Hostname	Global Database Name
Primary node	edbcn0.oracle.com	DBN0.oracle.com
Downstream system	c<n>ds1	C<n>DS1A.oracle.com

- 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>.

On the primary node

```
[oracle@edbcn0 ~]$ cd $ORACLE_HOME/dbs
[oracle@edbcn0 dbs]$ ls -al *pw*
-rw-r----- 1 oracle oinstall 7680 Oct 21 11:54 orapwDBN0
[oracle@edbcn0 dbs]$ rm orapwDBN0
[oracle@edbcn0 dbs]$
```

On the downstream system

```
[oracle@c<n>ds1 ~]$ cd $ORACLE_HOME/dbs
[oracle@c<n>ds1 dbs]$ ls -al *pw*
-rw-r----- 1 oracle oinstall 7680 Oct 15 09:25 orapwC<n>DS1A
[oracle@c<n>ds1 dbs]$ rm orapwC<n>DS1A
[oracle@c<n>ds1 dbs]$
```

- 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 generated
ENTRIES: The maximum number of SYSDBA entries to be granted
FORECE: Override if password file exist.

On the primary node

```
[oracle@edbcn0 dbs]$ orapwd file=orapwDBN0 password=oracle ignorecase=Y
entries=30 force=Y
[oracle@edbcn0 dbs]$ pwd
/u01/app/oracle/product/12.1.0/dbhome_1/dbs
```

- Remote copy the password file from primary node to downstream system. Use the extended node hostname, avoid using the IP address.

  The Password for the downstream system: oracle_c<n>

On the primary node

```
[oracle@edbcn0 dbs]$ scp orapwDBN0
root@c<n>ds1:/u01/app/oracle/product/12.1.0/dbhome_1/dbs
The authenticity of host 'c<n>ds1 (192.168.1.3<n>)' can't be established.
RSA key fingerprint is 99:6b:8f:33:fe:dc:62:1e:c8:db:55:27:cd:d3:bb:32.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'c<n>ds1,192.168.1.3<n>' (RSA) to the list of known
hosts.
root@c<n>ds1's password:
orapwDBN0                                                     100%   18KB
17.5KB/s   00:00
[oracle@edbcn0 dbs]$
```

- Rename and change ownership on the downstream system. To change the ownership, login as root for executing the command 'chown'.

  On Linux/UNIX platform, the password file is case sensitive. The file name format must be **orapwC<n>DS1A**.

On downstream system C<n>DS1A

```
[oracle@c<n>ds1 dbs]$ ls -al *pw*
-rw-r----- 1 root root 17920 Nov  5 17:43 orapwDBN0
[oracle@c<n>ds1 dbs]$ mv orapwDBN0 orapwC<n>DS1A
[oracle@c<n>ds1 dbs]$ su
Password:
[root@c<n>ds1 dbs]# chown oracle:oinstall orapwC<n>DS1A
[root@c<n>ds1 dbs]# exit
exit
[oracle@c<n>ds1 dbs]$ ls -al *pw*
-rw-r----- 1 oracle oinstall 17920 Nov  5 17:43 orapwC<n>DS1A
[oracle@c<n>ds1 dbs]$
```

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

On the primary node

```
[oracle@edbcn0 dbs]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sat Nov 5 18:24:29 2016

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

SQL> conn / as sysdba
Connected to an idle instance.
SQL> startup force
ORACLE instance started.

Total System Global Area    1660944384 bytes
Fixed Size                     2925072 bytes
Variable Size               1056968176 bytes
Database Buffers             587202560 bytes
Redo Buffers                  13848576 bytes
Database mounted.
Database opened.
SQL>
```

On downstream system C<n>DS1A

```
[oracle@c<n>ds1 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sat Nov 5 18:24:21 2016

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

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

Total System Global Area    2516582400 bytes
Fixed Size                     2927528 bytes
Variable Size                671089752 bytes
Database Buffers            1828716544 bytes
Redo Buffers                  13848576 bytes
Database mounted.
Database opened.
SQL>
```

- Configure the primary node database for log shipping. The database is operating on archive mode, shipping the transactions logs from the primary node to the downstream system.

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

On the primary node

```
[oracle@edbcn0 admin]$ tnsping c<n>ds1a.oracle.com

TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 05-NOV-2016
18:43:03

Copyright (c) 1997, 2014, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = c<n>ds1.oracle.com)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
C<n>DS1A.oracle.com)))
OK (30 msec)
[oracle@edbcn0 admin]$
```

```
[oracle@edbcn0 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sat Nov 5 18:44:15 2016

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

SQL> connect / as sysdba
Connected.
SQL>
```

- Configure LOG_ARCHIVE_DEST_2 to transmit the Redo log files to the downstream system mining database.

```
SQL> alter system set log_archive_dest_2='SERVICE=c<n>ds1a.oracle.com SYNC
NOREGISTER VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) REOPEN=10
DB_UNIQUE_NAME=C<n>DS1A' scope=both;

System altered.

SQL>
```

- LOG_ARCHIVE_DEST_2 must be explicitly enabled.

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

System altered.

SQL>
```

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

```
SQL> ALTER SYSTEM SET
  2  LOG_ARCHIVE_CONFIG='DG_CONFIG=(DBN0, C<n>DS1A)' scope=both;

System altered.

SQL>
```

- 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.

```
SQL> select bytes/1024/1024 MB from v$log;

     MB
----------
     50
     50
     50

SQL> select count(group#) from v$log;

COUNT(GROUP#)
-------------
         3

SQL>
```

- Configure the standby redo log groups. The number of the standby Redo log groups, should be one greater than the primary node, i.e. 4.

On Downstream system C<n>DS1A

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

FILE NAME
--------------------------------------------------------------------------------
/u01/app/oracle/oradata/C<n>DS1A/datafile/o1_mf_users_d03lmmjg_.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/C<n>DS1A/arch

SQL> host mkdir -p /u01/app/oracle/oradata/C<n>DS1A/redo
```

```
SQL> host mkdir -p /u01/app/oracle/oradata/C<n>DS1A/standby

SQL>
```

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

```
SQL> alter system set
  2  log_archive_dest_1='LOCATION=/u01/app/oracle/oradata/C<n>DS1A/arch
  3  VALID_FOR=(ONLINE_LOGFILE,PRIMARY_ROLE)' scope=both;

System altered.

SQL> alter system set
  2  log_archive_dest_2='LOCATION=/u01/app/oracle/oradata/C<n>DS1A/standby
  3  VALID_FOR=(STANDBY_LOGFILE,ALL_ROLES)' scope=both;

System altered.

SQL> ALTER SYSTEM SET
  2  LOG_ARCHIVE_CONFIG='DG_CONFIG=(DBN0, C<n>DS1A)' scope=both;

System altered.

SQL>
```

- Enable both archive log 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>
```

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

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

MEMBER
--------------------------------------------------------------------------------
/u01/app/oracle/oradata/C<n>DS1A/onlinelog/o1 mf 3 d03lo37k .log
/u01/app/oracle/fast_recovery_area/C<n>DS1A/onlinelog/o1_mf_3_d03lo3d4_.log
/u01/app/oracle/oradata/C<n>DS1A/onlinelog/o1 mf 2 d03lnzj6 .log
/u01/app/oracle/fast_recovery_area/C<n>DS1A/onlinelog/o1_mf_2_d03lnzp4_.log
/u01/app/oracle/oradata/C<n>DS1A/onlinelog/o1_mf_1_d03lnw6z_.log
/u01/app/oracle/fast_recovery_area/C<n>DS1A/onlinelog/o1_mf_1_d03lnwsm_.log

6 rows selected.

SQL>
```

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

```
SQL> alter database add standby logfile group 4
  2
('/u01/app/oracle/oradata/C<n>DS1A/onlinelog/slog4a.log','/u01/app/oracle/fast_r
ecovery_area/C<n>DS1A/onlinelog/slog4b.log') size 60M;

Database altered.


SQL> alter database add standby logfile group 5
  2
('/u01/app/oracle/oradata/C<n>DS1A/onlinelog/slog5a.log','/u01/app/oracle/fast_r
ecovery_area/C<n>DS1A/onlinelog/slog5b.log') size 60M;

Database altered.

SQL> alter database add standby logfile group 6
  2
('/u01/app/oracle/oradata/C<n>DS1A/onlinelog/slog6a.log','/u01/app/oracle/fast_r
ecovery_area/C<n>DS1A/onlinelog/slog6b.log') size 60M;

Database altered.

SQL> alter database add standby logfile group 7
  2
('/u01/app/oracle/oradata/C<n>DS1A/onlinelog/slog7a.log','/u01/app/oracle/fast_r
ecovery_area/C<n>DS1A/onlinelog/slog7b.log') size 60M;

Database altered.

SQL>
```

- Verify the groups successfully created

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

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

SQL>
```

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

- From the primary node, open two terminals windows:

On the first terminal window navigate to the database instance logging directory /u01/app/oracle/diag/rdbms/dbn0/DBN0/trace.

```
[oracle@edbcn0 ~]$ cd $ORACLE_DIAG
[oracle@edbcn0 trace]$ pwd
/u01/app/oracle/diag/rdbms/dbn0/DBN0/trace
[oracle@edbcn0 trace]$
```

On the second terminal open SQL*Plus and login as SYSDBA. The purpose is to show the below message which indicates the archival destination is receiving redo-transport successfully from the primary node.

```
*************************************************************
LGWR: Setting 'active' archival for destination LOG_ARCHIVE_DEST_2
*************************************************************
```

On the primary node - Terminal 1

- Execute the 'tail –f' command on the database alert file.

```
[oracle@edbcn0 trace]$ tail -f /u01/app/oracle/diag/rdbms/dbn1/DBN1/trace/*log*
Sat Nov 05 18:47:20 2016
Thread 1 advanced to log sequence 15 (LGWR switch)
  Current log# 3 seq# 15 mem# 0:
/u01/app/oracle/oradata/DBN0/onlinelog/o1_mf_3_d0moqnb9_.log
  Current log# 3 seq# 15 mem# 1:
/u01/app/oracle/fast_recovery_area/DBN0/onlinelog/o1_mf_3_d0moqnd3_.log
Sat Nov 05 18:47:20 2016
Archived Log entry 2 added for thread 1 sequence 14 ID 0x502a3e1a dest 1:
Sat Nov 05 18:47:49 2016
ALTER SYSTEM SET log_archive_dest_state_2='ENABLE' SCOPE=BOTH;
Sat Nov 05 18:50:07 2016
ALTER SYSTEM SET log_archive_config='DG_CONFIG=(DBN0, C<n>DS1A)' SCOPE=BOTH;
```

On the primary node - Terminal 2

- Login to SQL*Plus as SYSDBA

```
[oracle@edbcn0 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sat Nov 5 19:08:36 2016

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

SQL> conn / as sysdba
Connected.
SQL> alter system switch logfile;

System altered.
```

```
SQL>
```

On the primary node - Terminal 1

- View the alert file update showing the message which indicates successful Redo-Transport to destination LOG_ARCHIVE_DEST_2.

```
Sat Nov 05 19:08:49 2016
**********************************************************************
LGWR: Setting 'active' archival for destination LOG_ARCHIVE_DEST_2
**********************************************************************
Sat Nov 05 19:08:49 2016
Thread 1 advanced to log sequence 16 (LGWR switch)
  Current log# 1 seq# 16 mem# 0:
/u01/app/oracle/oradata/DBN0/onlinelog/o1 mf 1 d0moqfdg .log
  Current log# 1 seq# 16 mem# 1:
/u01/app/oracle/fast recovery area/DBN0/onlinelog/o1 mf 1 d0moqfgx .log
Sat Nov 05 19:08:49 2016
Archived Log entry 5 added for thread 1 sequence 15 ID 0x502a3e1a dest 1:
```

Close PuTTY terminals for:

EDBCN0
&
C<n>DS1A

Task 2: Configure and Verify Redo-Transport from Extended Node 1 to Downstream System C<n>DS1B

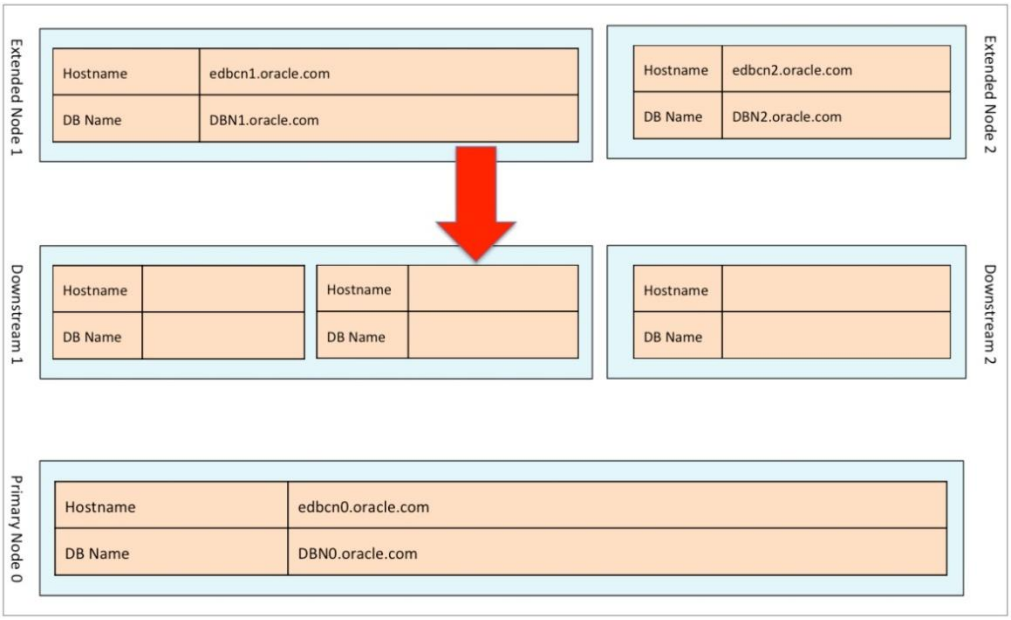
Complete extended node 1 details:

Hostname	
IP Address	
Global Database Name	
Database Unique Name	
Connect string	

Complete the downstream system for extended node 1 details:

Hostname	
IP Address	
Global Database Name	
Database Unique Name	
Connect string	

Complete the diagram below:



- Re-create the password file on the primary node and downstream system – C<n>DS1B. The purpose of re-creating the password file is to ensure both systems have identical passwords, otherwise, the Redo-transport will fail. The steps for re-creating the password file is online, specifying IGNORECASE requires restarting the database instance.

    7. Login to the extended node 1 and downstream system
    8. Remove the current password file
    9. Re-create the password file on the extended node 1 using orapwd utility
    10. Remote copy the password file from the extended node 1 to the downstream system
    11. Rename and change permission on the downstream system
    12. Restart the database on the extended node 1 and downstream system

Open two PuTTY terminals, execute the following steps to re-create the password successfully.

Server	Hostname	Global Database Name
Extended node 1	edbcn1.oracle.com	DBN1.oracle.com
Downstream system	c<n>ds1	C<n>DS1B.oracle.com

- 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>.

On extended node 1

```
[oracle@edbcn1 ~]$ cd $ORACLE_HOME/dbs
[oracle@edbcn1 dbs]$ ls -al *pw*
-rw-r----- 1 oracle oinstall 7680 Oct 21 12:20 orapwDBN1
[oracle@edbcn1 dbs]$ rm orapwDBN1
[oracle@edbcn1 dbs]$
```

On downstream system C<n>DS1B

```
[oracle@c<n>ds1 ~]$ . oraenv
ORACLE_SID = [C<n>DS1A] ? C<n>DS1B
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@c<n>ds1 ~]$
```

```
[oracle@c<n>ds1 ~]$ cd $ORACLE_HOME
[oracle@c<n>ds1 dbhome_2]$ pwd
/u01/app/oracle/product/12.1.0/dbhome_2
[oracle@c<n>ds1 dbhome_2]$ cd $ORACLE_HOME/dbs
[oracle@c<n>ds1 dbs]$ ls -al *pw*
```

```
-rw-r----- 1 oracle oinstall 7680 Oct 15 09:29 orapwC<n>DS1B
[oracle@c<n>ds1 dbs]$ rm orapwC<n>DS1B
[oracle@c<n>ds1 dbs]$
```

- 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
  FORECE: Override if password file exist.

On extended node 1

```
[oracle@edbcn1 dbs]$ orapwd file=orapwDBN1 password=oracle ignorecase=Y
entries=30 force=Y
[oracle@edbcn1 dbs]$ pwd
/u01/app/oracle/product/12.1.0/dbhome_1/dbs
[oracle@edbcn1 dbs]$ ls -al *pw*
-rw-r----- 1 oracle oinstall 17920 Nov  7 05:33 orapwDBN1
[oracle@edbcn1 dbs]$
```

- Remote copy the password file from primary node to downstream system. Use the extended node hostname, avoid using the IP address.

  Ensure the current oracle home is:
  `/u01/app/oracle/product/12.1.0/dbhome_2/dbs`

On extended node 1

```
[oracle@edbcn1 dbs]$ scp orapwDBN1
root@c<n>ds1:/u01/app/oracle/product/12.1.0/dbhome_2/dbs
root@c<n>ds1's password:
orapwDBN1                                                 100%   18KB
17.5KB/s   00:00
[oracle@edbcn1 dbs]$
```

- Rename and change ownership on the downstream system. To change the ownership, login as root for executing the command 'chown'.

On downstream system C<n>DS1B

```
[oracle@c<n>ds1 dbs]$ cat /etc/oratab | grep home
# The first and second fields are the system identifier and home
C<n>DS1A:/u01/app/oracle/product/12.1.0/dbhome_1:N
C<n>DS1B:/u01/app/oracle/product/12.1.0/dbhome_2:N
[oracle@c<n>ds1 dbs]$ mv orapwDBN1 orapwC<n>DS1B
[oracle@c<n>ds1 dbs]$ su
```

```
Password:
[root@c<n>ds1 dbs]# chown oracle:oinstall orapwC<n>DS1B
[root@c<n>ds1 dbs]# exit
exit
[oracle@c<n>ds1 dbs]$ ls -al *pw*
-rw-r----- 1 oracle oinstall 17920 Nov  7 05:40 orapwC<n>DS1B
[oracle@c<n>ds1 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.

On extended node 1

```
[oracle@edbcn1 dbs]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Mon Nov 7 05:44:14 2016

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

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

Total System Global Area     1660944384 bytes
Fixed Size                      2925072 bytes
Variable Size                1056968176 bytes
Database Buffers              587202560 bytes
Redo Buffers                   13848576 bytes
Database mounted.
Database opened.
SQL>
```

On downstream system C<n>DS1B

```
[oracle@c<n>ds1 dbs]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Mon Nov 7 05:44:23 2016

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

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

Total System Global Area    2516582400 bytes
Fixed Size                     2927528 bytes
Variable Size                671089752 bytes
Database Buffers            1828716544 bytes
Redo Buffers                  13848576 bytes
Database mounted.
Database opened.
```

```
SQL>
```

- Configure the primary node database for log shipping. The database is operating on archive mode, shipping the transactions logs from the primary node to the downstream system.

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

On extended node 1

```
[oracle@edbcn1 admin]$ tnsping c<n>ds1b.oracle.com

TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 07-NOV-2016
05:48:55

Copyright (c) 1997, 2014, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = c<n>ds1.oracle.com)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
C<n>DS1B.oracle.com)))
OK (10 msec)
[oracle@edbcn1 admin]$
```

```
[oracle@edbcn1 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Mon Nov 7 05:49:30 2016

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

SQL> conn / as sysdba
Connected.
SQL>
```

- Configure LOG_ARCHIVE_DEST_2 to transmit the Redo log files to the downstream system mining database.

```
SQL> alter system set log_archive_dest_2='SERVICE=c<n>ds1b.oracle.com SYNC
NOREGISTER VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) REOPEN=10
DB_UNIQUE_NAME=C<n>DS1B' scope=both;

System altered.

SQL>
```

- LOG_ARCHIVE_DEST_2 must be explicitly enabled

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

System altered.

SQL>
```

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

```
SQL> ALTER SYSTEM SET
  2  LOG_ARCHIVE_CONFIG='DG_CONFIG=(DBN1, C<n>DS1B)' scope=both;

System altered.

SQL>
```

- 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.

```
SQL> select bytes/1024/1024 MB from v$log;

     MB
----------
     50
     50
     50

SQL> select count(group#) from v$log;

COUNT(GROUP#)
-------------
         3

SQL>
```

- Configure the standby redo log groups. The number of the standby Redo log groups, should be one greater than the primary node, i.e. 4.

On downstream system C<n>DS1B

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

FILE NAME
--------------------------------------------------------------------------------
/u01/app/oracle/oradata/C<n>DS1B/datafile/o1 mf users d03lnfjg .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/C<n>DS1B/arch

SQL> host mkdir -p /u01/app/oracle/oradata/C<n>DS1B/redo

SQL> host mkdir -p /u01/app/oracle/oradata/C<n>DS1B/standby

SQL>
```

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

```
SQL> alter system set
  2  log_archive_dest_1='LOCATION=/u01/app/oracle/oradata/C<n>DS1B/arch
  3  VALID_FOR=(ONLINE_LOGFILE,PRIMARY_ROLE)' scope=both;

System altered.

SQL> alter system set
  2  log_archive_dest_2='LOCATION=/u01/app/oracle/oradata/C<n>DS1B/standby
  3  VALID_FOR=(STANDBY_LOGFILE,ALL_ROLES)' scope=both;

System altered.

SQL> ALTER SYSTEM SET
  2  LOG_ARCHIVE_CONFIG='DG_CONFIG=(DBN1, C<n>DS1B)' 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>
```

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

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

MEMBER
--------------------------------------------------------------------------------
```

```
/u01/app/oracle/oradata/C<n>DS1B/onlinelog/o1_mf_3_d03lwggr_.log
/u01/app/oracle/fast_recovery_area/C<n>DS1B/onlinelog/o1_mf_3_d03lwgn1_.log
/u01/app/oracle/oradata/C<n>DS1B/onlinelog/o1 mf 2 d03lwcws .log
/u01/app/oracle/fast_recovery_area/C<n>DS1B/onlinelog/o1_mf_2_d03lwd28_.log
/u01/app/oracle/oradata/C<n>DS1B/onlinelog/o1_mf_1_d03lw94c_.log
/u01/app/oracle/fast_recovery_area/C<n>DS1B/onlinelog/o1_mf_1_d03lw98t_.log

6 rows selected.

SQL>
```

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

```
SQL> alter database add standby logfile group 4
  2
('/u01/app/oracle/oradata/C<n>DS1B/onlinelog/slog4a.log','/u01/app/oracle/fast_r
ecovery_area/C<n>DS1B/onlinelog/slog4b.log') size 60M;

Database altered.

SQL> alter database add standby logfile group 5
  2
('/u01/app/oracle/oradata/C<n>DS1B/onlinelog/slog5a.log','/u01/app/oracle/fast_r
ecovery_area/C<n>DS1B/onlinelog/slog5b.log') size 60M;

Database altered.

SQL> alter database add standby logfile group 6
  2
('/u01/app/oracle/oradata/C<n>DS1B/onlinelog/slog6a.log','/u01/app/oracle/fast_r
ecovery_area/C<n>DS1B/onlinelog/slog6b.log') size 60M;

Database altered.

SQL> alter database add standby logfile group 7
  2
('/u01/app/oracle/oradata/C<n>DS1B/onlinelog/slog7a.log','/u01/app/oracle/fast_r
ecovery_area/C<n>DS1B/onlinelog/slog7b.log') size 60M;

Database altered.

SQL>
```

- Verify the groups successfully created

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

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

```
      7          0          0 YES UNASSIGNED

SQL>
```

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

- From the primary node, open two terminals windows:

On the first terminal window navigate to the database instance logging directory /u01/app/oracle/diag/rdbms/dbn1/DBN1/trace.

On the second terminal open SQL*Plus and login as SYSDBA. The purpose is to show the below message which indicates the archival destination is receiving redo-transport successfully from the primary node.

```
************************************************************
LGWR: Setting 'active' archival for destination LOG_ARCHIVE_DEST_2
************************************************************
```

On extended node 1 - Terminal 1

- Execute the 'tail –f' command on the database alert file.

```
[oracle@edbcn1 ~]$ cd $ORACLE_DIAG
[oracle@edbcn1 trace]$ pwd
/u01/app/oracle/diag/rdbms/dbn1/DBN1/trace
[oracle@edbcn1 trace]$
```

```
[oracle@edbcn1 trace]$ tail -f /u01/app/oracle/diag/rdbms/dbn1/DBN1/trace/*log*
ORA-16055: FAL request rejected
ARCH: FAL archive failed. Archiver continuing
Mon Nov 07 05:51:17 2016
ORACLE Instance DBN1 - Archival Error. Archiver continuing.
Mon Nov 07 05:51:31 2016
Error 12154 received logging on to the standby
Mon Nov 07 05:52:05 2016
USER: DB_UNIQUE_NAME C<n>DS1A is not in the Data Guard configuration
Mon Nov 07 05:52:05 2016
ALTER SYSTEM SET log_archive_config='DG_CONFIG=(DBN1, C<n>DS1B)' SCOPE=BOTH;
```

On extended node 1 - Terminal 2

- Login to SQL*Plus as SYSDBA

```
[oracle@edbcn0 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sat Nov 5 19:08:36 2016

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

```
SQL> conn / as sysdba
Connected.
SQL> alter system switch logfile;

System altered.

SQL>
```

On the extended node - Terminal 1

- View the alert file update showing the message which indicates successful Redo-Transport to destination LOG_ARCHIVE_DEST_2.

```
Mon Nov 07 06:12:11 2016
************************************************************
LGWR: Setting 'active' archival for destination LOG ARCHIVE DEST 2
************************************************************
Starting background process NSS2
Mon Nov 07 06:12:11 2016
NSS2 started with pid=7, OS id=13825
Mon Nov 07 06:12:17 2016
Thread 1 advanced to log sequence 16 (LGWR switch)
  Current log# 1 seq# 16 mem# 0:
/u01/app/oracle/oradata/DBN1/onlinelog/o1_mf_1_d0mq9dbv_.log
  Current log# 1 seq# 16 mem# 1:
/u01/app/oracle/fast recovery area/DBN1/onlinelog/o1 mf 1 d0mq9dg6 .log
Mon Nov 07 06:12:17 2016
Archived Log entry 5 added for thread 1 sequence 15 ID 0x5279e21c dest 1:
```

Close PuTTY terminals for:

EDBCN1
&
C<n>DS1B

Task 3: Configure and Verify Redo-Transport from Extended Node 2 to Downstream System C<n>DS2A

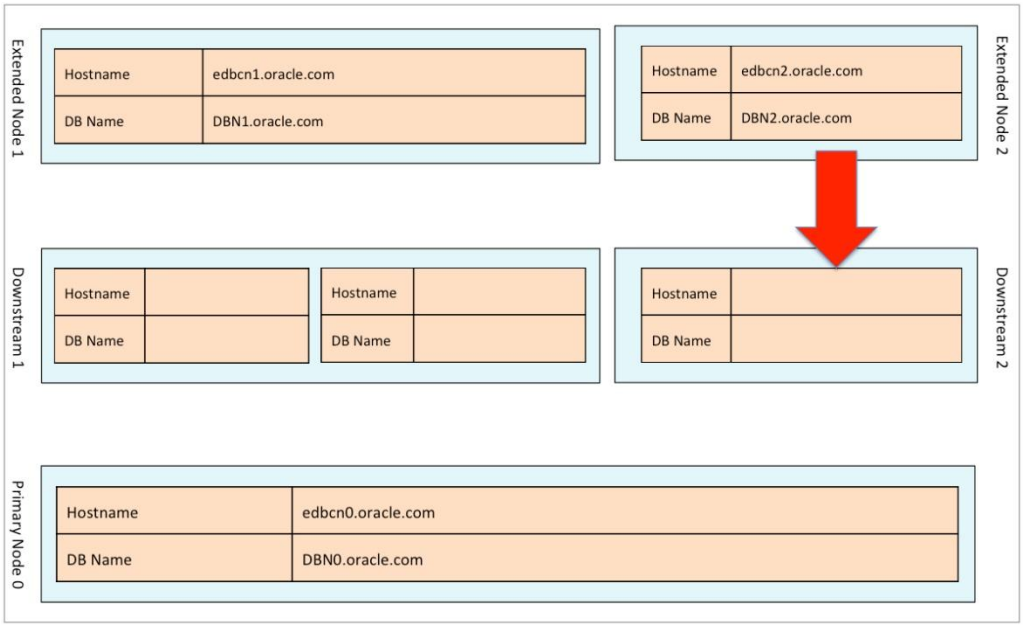
Complete extended node 2 details:

Hostname	
IP Address	
Global Database Name	
Database Unique Name	
Connect string	

Complete the downstream system for extended node 2 details:

Hostname	
IP Address	
Global Database Name	
Database Unique Name	
Connect string	

Complete the diagram below:



- Re-create the password file on the extended node 2 and downstream system – C<n>DS2A. The purpose of re-creating the password file is to ensure both systems have identical passwords, otherwise, the Redo-transport will fail. The steps for re-creating the password file is online, specifying IGNORECASE requires restarting the database instance.

    13. Login to the extended node 2 and downstream system
    14. Remove the current password file
    15. Re-create the password file on the extended node 2 using orapwd utility
    16. Remote copy the password file from the extended node 2 to the downstream system
    17. Rename and change permission on the downstream system
    18. Restart the database on the extended node 2 and downstream system

Open two PuTTY terminals, execute the following steps to re-create the password successfully.

Server	Hostname	Global Database Name
Extended node 2	edbcn2.oracle.com	DBN2.oracle.com
Downstream system	c<n>ds2	C<n>DS2A.oracle.com

- 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>.

On extended node 2

```
[oracle@edbcn2 ~]$ cd $ORACLE_HOME/dbs
[oracle@edbcn2 dbs]$ ls -al *pw*
-rw-r----- 1 oracle oinstall 7680 Oct 22 17:55 orapwDBN2
[oracle@edbcn2 dbs]$ rm orapwDBN2
[oracle@edbcn2 dbs]$
```

On downstream system C<n>DS2A

```
[oracle@c<n>ds2 ~]$ cd $ORACLE_HOME/dbs
[oracle@c<n>ds2 dbs]$ ls -al *pw*
-rw-r----- 1 oracle oinstall 7680 Oct 15 13:37 orapwC<n>DS2A
[oracle@c<n>ds2 dbs]$ rm orapwC<n>DS2A
[oracle@c<n>ds2 dbs]$ pwd
/u01/app/oracle/product/12.1.0/dbhome_1/dbs
[oracle@c<n>ds2 dbs]$
```

- 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
  FORECE: Override if password file exist.

On extended node 2

```
[oracle@edbcn2 dbs]$ orapwd file=orapwDBN2 password=oracle ignorecase=Y
entries=30 force=Y
[oracle@edbcn2 dbs]$ pwd
/u01/app/oracle/product/12.1.0/dbhome 1/dbs
[oracle@edbcn2 dbs]$
```

- Remote copy the password file from primary node to downstream system. Use the extended node hostname, avoid using the IP address.

On extended node 2

```
[oracle@edbcn2 dbs]$ scp orapwDBN2
root@c<n>ds2:/u01/app/oracle/product/12.1.0/dbhome_1/dbs
The authenticity of host 'c<n>ds2 (192.168.1.4<n>)' can't be established.
RSA key fingerprint is f4:4d:6f:48:ac:c8:c1:01:ff:4b:0f:74:da:ef:c1:63.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'c<n>ds2,192.168.1.4<n>' (RSA) to the list of known
hosts.
root@c<n>ds2's password:
orapwDBN2                                                     100%   18KB
17.5KB/s   00:00
[oracle@edbcn2 dbs]$
```

- Rename and change ownership on the downstream system. To change the ownership, login as root for executing the command 'chown'.

On downstream system C<n>DS2A

```
[oracle@c<n>ds2 dbs]$ ls -al *pw*
-rw-r----- 1 root root 17920 Nov  8 06:12 orapwDBN2
[oracle@c<n>ds2 dbs]$ cat /etc/oratab | grep home
# The first and second fields are the system identifier and home
C<n>DS2A:/u01/app/oracle/product/12.1.0/dbhome 1:N
[oracle@c<n>ds2 dbs]$ mv orapwDBN2 orapwC<n>DS2A
[oracle@c<n>ds2 dbs]$ su
Password:
[root@c<n>ds2 dbs]# chown oracle:oinstall orapwC<n>DS2A
[root@c<n>ds2 dbs]# exit
exit
[oracle@c<n>ds2 dbs]$ ls -al *pw*
```

```
-rw-r----- 1 oracle oinstall 17920 Nov  8 06:12 orapwC<n>DS2A
[oracle@c<n>ds2 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.

On extended node 2

```
[oracle@edbcn2 dbs]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Tue Nov 8 06:16:23 2016

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

SQL> conn / as sysdba
Connected to an idle instance.
SQL> startup force
ORACLE instance started.

Total System Global Area    1660944384 bytes
Fixed Size                     2925072 bytes
Variable Size               1040190960 bytes
Database Buffers             603979776 bytes
Redo Buffers                  13848576 bytes
Database mounted.
Database opened.
SQL>
```

On downstream system C<n>DS2A

```
[oracle@c<n>ds2 dbs]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Tue Nov 8 06:16:55 2016

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

SQL> conn / as sysdba
Connected to an idle instance.
SQL> startup force
ORACLE instance started.

Total System Global Area    1660944384 bytes
Fixed Size                     2925072 bytes
Variable Size                989859312 bytes
Database Buffers             654311424 bytes
Redo Buffers                  13848576 bytes
Database mounted.
Database opened.
SQL>
```

- Configure the primary node database for log shipping. The database is operating on archive mode, shipping the transactions logs from the primary node to the downstream system.

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

On extended node 2

```
[oracle@edbcn2 ~]$ tnsping c<n>ds2a.oracle.com

TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 08-NOV-2016
06:19:51

Copyright (c) 1997, 2014, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = c<n>ds2.oracle.com)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
C<n>DS2A.oracle.com)))
OK (10 msec)
[oracle@edbcn2 ~]$
```

```
[oracle@edbcn2 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Tue Nov 8 06:20:16 2016

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

SQL> conn / as sysdba
Connected.
SQL>
```

- Configure LOG_ARCHIVE_DEST_2 to transmit the Redo log files to the downstream system mining database.

```
SQL> alter system set log_archive_dest_2='SERVICE=c<n>ds2a.oracle.com SYNC
NOREGISTER VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) REOPEN=10
DB_UNIQUE_NAME=C<n>DS2A' scope=both;

System altered.

SQL>
```

- LOG_ARCHIVE_DEST_2 must be explicitly enabled

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

```
System altered.

SQL>
```

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

```
SQL> ALTER SYSTEM SET
  2  LOG_ARCHIVE_CONFIG='DG_CONFIG=(DBN2, C<n>DS2A)' scope=both;

System altered.

SQL>
```

- 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.

```
SQL> select bytes/1024/1024 MB from v$log;

     MB
----------
     50
     50
     50

SQL> select count(group#) from v$log;

COUNT(GROUP#)
-------------
         3

SQL>
```

- Configure the standby redo log groups. The number of the standby Redo log groups, should be one greater than the primary node, i.e. 4.

On downstream system C<n>DS2A

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

FILE_NAME
--------------------------------------------------------------------------------
/u01/app/oracle/oradata/C<n>DS2A/datafile/o1_mf_users_d041jht6_.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.

On downstream system C<n>DS2A

```
SQL> host mkdir -p /u01/app/oracle/oradata/C<n>DS2A/arch

SQL> host mkdir -p /u01/app/oracle/oradata/C<n>DS2A/redo

SQL> host mkdir -p /u01/app/oracle/oradata/C<n>DS2A/standby

SQL>
```

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

On downstream system C<n>DS2A

```
SQL> alter system set
  2  log_archive_dest_1='LOCATION=/u01/app/oracle/oradata/C<n>DS2A/arch
  3  VALID_FOR=(ONLINE_LOGFILE,PRIMARY_ROLE)' scope=both;

System altered.

SQL> alter system set
  2  log_archive_dest_2='LOCATION=/u01/app/oracle/oradata/C<n>DS2A/standby
  3  VALID_FOR=(STANDBY_LOGFILE,ALL_ROLES)' scope=both;

System altered.

SQL> ALTER SYSTEM SET
  2  LOG_ARCHIVE_CONFIG='DG_CONFIG=(DBN2, C<n>DS2A)' scope=both;

System altered.

SQL>
```

- Enable both destinations.

On downstream system C<n>DS2A

```
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>
```

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

On downstream system C<n>DS2A

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

MEMBER
--------------------------------------------------------------------------------
/u01/app/oracle/oradata/C<n>DS2A/onlinelog/o1_mf_3_d041kqlq_.log
/u01/app/oracle/fast recovery area/C<n>DS2A/onlinelog/o1 mf 3 d041kqp1 .log
/u01/app/oracle/oradata/C<n>DS2A/onlinelog/o1_mf_2_d041knmk_.log
/u01/app/oracle/fast_recovery_area/C<n>DS2A/onlinelog/o1_mf_2_d041knsk_.log
/u01/app/oracle/oradata/C<n>DS2A/onlinelog/o1_mf_1_d041kk9j_.log
/u01/app/oracle/fast recovery area/C<n>DS2A/onlinelog/o1 mf 1 d041kkgh .log

6 rows selected.

SQL>
```

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

On downstream system C<n>DS2A

```
SQL> alter database add standby logfile group 4
  2
('/u01/app/oracle/oradata/C<n>DS2A/onlinelog/slog4a.log','/u01/app/oracle/fast_r
ecovery_area/C<n>DS2A/onlinelog/slog4b.log') size 60M;

Database altered.

SQL> alter database add standby logfile group 5
  2
('/u01/app/oracle/oradata/C<n>DS2A/onlinelog/slog5a.log','/u01/app/oracle/fast_r
ecovery_area/C<n>DS2A/onlinelog/slog5b.log') size 60M;

Database altered.

SQL> alter database add standby logfile group 6
  2
('/u01/app/oracle/oradata/C<n>DS2A/onlinelog/slog6a.log','/u01/app/oracle/fast_r
ecovery_area/C<n>DS2A/onlinelog/slog6b.log') size 60M;

Database altered.

SQL> alter database add standby logfile group 7
  2
('/u01/app/oracle/oradata/C<n>DS2A/onlinelog/slog7a.log','/u01/app/oracle/fast_r
ecovery_area/C<n>DS2A/onlinelog/slog7b.log') size 60M;

Database altered.

SQL>
```

- Verify the groups successfully created

On downstream system C<n>DS2A

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

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

SQL>
```

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

- From the primary node, open two terminals windows:

On the first terminal window navigate to the database instance logging directory /u01/app/oracle/diag/rdbms/dbn2/DBN2/trace.

```
[oracle@edbcn2 ~]$ cd $ORACLE_DIAG
[oracle@edbcn2 trace]$ pwd
/u01/app/oracle/diag/rdbms/dbn2/DBN2/trace
[oracle@edbcn2 trace]$
```

On the second terminal open SQL*Plus and login as SYSDBA. The purpose is to show the below message which indicates the archival destination is receiving redo-transport successfully from the primary node.

```
************************************************************************
LGWR: Setting 'active' archival for destination LOG_ARCHIVE_DEST_2
************************************************************************
```

On extended node 2 - Terminal 1

- Execute the 'tail –f' command on the database alert file.

```
[oracle@edbcn2 trace]$ tail -f *log*
************************************************************************
Starting background process NSS2
Tue Nov 08 06:48:32 2016
NSS2 started with pid=7, OS id=8462
Tue Nov 08 06:48:38 2016
Thread 1 advanced to log sequence 16 (LGWR switch)
  Current log# 1 seq# 16 mem# 0:
/u01/app/oracle/oradata/DBN2/onlinelog/o1_mf_1_d0pz9qcc_.log
  Current log# 1 seq# 16 mem# 1:
/u01/app/oracle/fast_recovery_area/DBN2/onlinelog/o1_mf_1_d0pz9qfx_.log
Tue Nov 08 06:48:38 2016
```

```
Archived Log entry 4 added for thread 1 sequence 15 ID 0x5f5718e6 dest 1:
```

On extended node 2 - Terminal 2

- Login to SQL*Plus as SYSDBA

```
[oracle@edbcn2 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Tue Nov 8 06:35:02 2016

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

SQL> conn / as sysdba
Connected.
SQL> alter system switch logfile;

System altered.

SQL>
```

On extended node 2 - Terminal 1

- View the alert file update showing the message which indicates successful Redo-Transport to destination LOG_ARCHIVE_DEST_2.

```
Tue Nov 08 06:48:32 2016
************************************************************
LGWR: Setting 'active' archival for destination LOG ARCHIVE DEST 2
************************************************************
Starting background process NSS2
Tue Nov 08 06:48:32 2016
NSS2 started with pid=7, OS id=8462
Tue Nov 08 06:48:38 2016
Thread 1 advanced to log sequence 16 (LGWR switch)
  Current log# 1 seq# 16 mem# 0:
/u01/app/oracle/oradata/DBN2/onlinelog/o1_mf_1_d0pz9qcc_.log
  Current log# 1 seq# 16 mem# 1:
/u01/app/oracle/fast recovery area/DBN2/onlinelog/o1 mf 1 d0pz9qfx .log
Tue Nov 08 06:48:38 2016
Archived Log entry 4 added for thread 1 sequence 15 ID 0x5f5718e6 dest 1:
```

Close PuTTY terminals for:

EDBCN2
&
C<n>DS2A

Workshop 4: Configure EDBC Integrated Extract (IE)

This workshop prepares and configure three unidirectional data replication to deploy a compute capacity of 3-node EDBC, which are:

- From the primary node to extended node 1
- From extended node 1 to the primary node
- From extended node 2 to the primary node

Table 8 lists Workshop 4 tasks.

Task No	Task Description
1	Prepare the primary node, the extended nodes and the downstream systems
2	Configure integrated capture for EDBC primary node
3	Configure integrated capture for EDBC extended node 1
4	Configure integrated capture for EDBC extended node 2
5	Review integrated capture for EDBC
6	Perform EDBC transactions tests

Table 8: Workshop 4 Tasks

Task 1: Prepare the Primary Node, Extended Nodes, and the Downstream Systems Database Parameters Settings

The downstream mining database demands adequate resources to perform mining operations to prepare the LCRn for change data capture (CDC) by the integrated extract (EI). Size the database parameter STREAMS_POOL_SIZE to the minimum recommended value of 750MB on the downstream system. The optimum size depends on the volume of Redo-Transport generated by the primary and extended nodes. The best case scenario is to processes the transaction logs immediately to avoid a rewind and scan operation that is resource intensive.

Note: It's recommended to operate the downstream node on auto-managed memory allocation. Unless, there is specific reason to set SGA components explicitly.

Refer to Appendix E for further details.

Prepare the Primary Node and Downstream System C<n>DS1A

- Set the database parameters ENABLE_GOLDENGATE_REPLICATION to TRUE. The parameter controls services provided by the RDBMS for Oracle

GoldenGate (both capture and apply services). Set this to true to enable RDBMS services used by Oracle GoldenGate.

On the primary node

```
[oracle@c<n>ds1 ~]$ echo $ORACLE_SID
C<n>DS1A
[oracle@c<n>ds1 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sun Dec 25 04:40:48 2016

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

SQL>
```

```
SQL> conn / as sysdba
Connected.
SQL> show parameters goldengate

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------
enable_goldengate_replication        boolean     FALSE

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

System altered.

SQL>
```

- Set the database parameters ENABLE_GOLDENGATE_REPLICATION to TRUE and STREAMS_POOL_SIZE to 1G.

On downstream system C<n>DS1A

```
[oracle@c<n>ds1 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Tue Nov 8 06:55:36 2016

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

SQL> conn / as sysdba
Connected.
SQL>
```

```
SQL> show parameters streams

NAME                                 TYPE        VALUE
------------------------------------ ----------- --------------------------
streams_pool_size                    big integer 0

SQL> alter system set streams_pool_size=1G scope=both;

System altered.
```

```
SQL> show parameters streams

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------
streams_pool_size                    big integer 1G

SQL> show parameters goldengate

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------
enable_goldengate_replication           boolean    FALSE

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

System altered.

SQL> show parameters goldengate

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------
enable_goldengate_replication           boolean    TRUE

SQL>
```

- Verify database connection to the primary node and extended node 1 using TNS.

On downstream system C<n>DS1A

```
SQL> conn ogg_admin/oracle@c<n>ds1a.oracle.com
Connected.
SQL>
```

On the primary node

```
SQL> conn ogg_admin/oracle@dbn0.oracle.com
Connected.
SQL>
```

Prepare Extended Node 1 and Downstream System C<n>DS1B

- Set the database parameters ENABLE_GOLDENGATE_REPLICATION to TRUE.

On extended node 1

```
SQL> conn / as sysdba
Connected.
SQL> show parameters goldengate

NAME                                 TYPE        VALUE
```

```
---------------------------------- ----------- ------------------------
enable_goldengate_replication           boolean    FALSE

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

System altered.

SQL>
```

- Switch to Oracle Home 2 (non-default)

On downstream system C<n>DS1B

```
[oracle@c<n>ds1 ~]$ cat /etc/oratab | grep home
# The first and second fields are the system identifier and home
C<n>DS1A:/u01/app/oracle/product/12.1.0/dbhome_1:N
C<n>DS1B:/u01/app/oracle/product/12.1.0/dbhome_2:N
[oracle@c<n>ds1 ~]$ . oraenv
ORACLE_SID = [C<n>DS1A] ? C<n>DS1B
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@c<n>ds1 ~]$
```

- Set the database parameters ENABLE_GOLDENGATE_REPLICATION to TRUE and STREAMS_POOL_SIZE to 1G.

```
[oracle@c<n>ds1 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Tue Nov 8 06:55:36 2016

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

SQL> conn / as sysdba
Connected.
SQL>
```

```
SQL> show parameters streams

NAME                                 TYPE       VALUE
------------------------------------ ----------- ---------------------------
streams_pool_size                    big integer 0

SQL> alter system set streams_pool_size=1G scope=both;

System altered.

SQL> show parameters streams

NAME                                 TYPE       VALUE
------------------------------------ ----------- -------------------------
streams pool size                    big integer 1G

SQL> show parameters goldengate

NAME                                 TYPE       VALUE
```

```
---------------------------------- ----------- -------------------------
enable_goldengate_replication           boolean    FALSE

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

System altered.

SQL> show parameters goldengate

NAME                                 TYPE        VALUE
---------------------------------- ----------- -------------------------
enable_goldengate_replication           boolean    TRUE

SQL>
```

- Verify connection to the primary node and extended node 1 database using TNS.

On downstream system C<n>DS1B

```
SQL> conn ogg_admin/oracle@c<n>ds1b.oracle.com
Connected.
SQL>
```

On extended node 1

```
SQL> conn ogg_admin/oracle@dbn1.oracle.com
Connected.
SQL>
```

Prepare Extended Node 2 and Downstream System C<n>DS2A

On extended node 2

- Set the database parameters ENABLE_GOLDENGATE_REPLICATION to TRUE.

```
SQL> conn / as sysdba
Connected.
SQL> show parameters goldengate

NAME                                 TYPE        VALUE
---------------------------------- ----------- -------------------------
enable_goldengate_replication           boolean    FALSE

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

System altered.

SQL>
```

On downstream system C<n>DS1B

- Set the database parameters ENABLE_GOLDENGATE_REPLICATION to TRUE and STREAMS_POOL_SIZE to 1G.

```
[oracle@c<n>ds2 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Tue Nov 8 12:57:01 2016

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

SQL> conn / as sysdba
Connected.
SQL>
```

```
SQL> show parameters streams

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
streams_pool_size                    big integer 0

SQL> alter system set streams_pool_size=1G scope=both;

System altered.

SQL> show parameters streams

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
streams pool size                    big integer 1G

SQL> show parameters goldengate

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
enable goldengate replication           boolean    FALSE

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

System altered.

SQL> show parameters goldengate

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
enable goldengate replication           boolean    TRUE

SQL>
```

- Verify connection to the downstream and extended node 2 database using TNS.

On downstream system C<n>DS1B

```
SQL> conn ogg_admin/oracle@c<n>ds2a.oracle.com
```

```
Connected.
SQL>
```

On extended node 2

```
SQL> conn ogg_admin/oracle@dbn2.oracle.com
Connected.
SQL>
```

Task 2: Configure Integrated Extract for the Primary Node

Primary Node Hostname	edbcn0.oracle.com
Global database name	DBN0.oracle.com
Downstream System Hostname	c<n>ds1.oracle.com
Global database name	C<n>DS1A.oracl.e.com

Begin by configuring the unidirectional data replication from the primary node to the extended node 1. The choice of the primary node is critical and must be selected very wisely. Choosing the central database is likely to be the best option. Figure 8 shows the data shipping from the primary to downstream system c<n>ds1.orale.com.

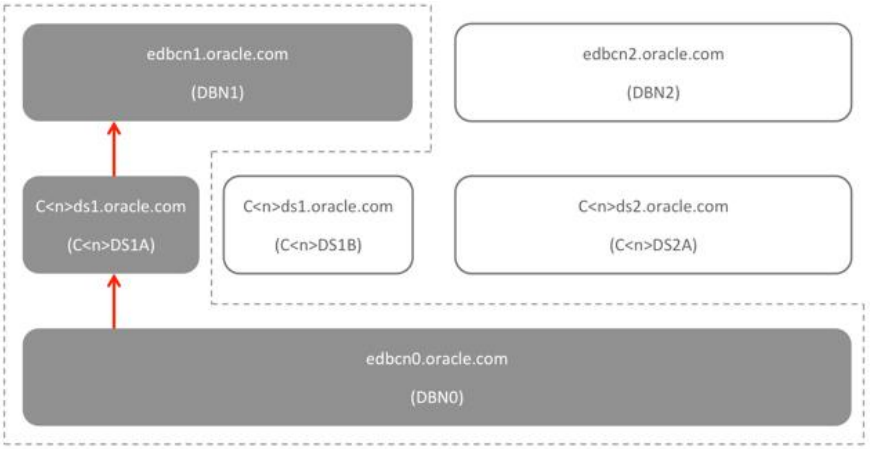


Figure 8: Primary node ▸ Downstream C<n>DS1A ▸ Extended node 1

The primary node Oracle GoldenGate 12c instance operated using the following processes and trails.

GoldenGate Process Name	Process Description
EDBN001	Integrated Extract for DBN0

./dirdat/sp	Local trail for the primary node
PDBN001	Data pump Extract
./dirdat/tp	Remote Trail for the primary node
RDBN001	The Replicat on extended node 1

The steps to configure the integrated capture:

- Login to the downstream system and the primary node
- Register the integrated extract
- Prepare and define the integrated extract and the local trail (sp)
- Prepare and define the data pump and the remote trail (tp)
- Prepare and Define the Replicat on extended node 1
- Add table-level supplemental logging

The downstream system has two GoldenGate instances. For unidirectional from the primary to the extended node 1, use $OGG_HOME1, and from the extended node 1 to the primary, use $OGG_HOME2.

```
[oracle@c<n>ds1 ~]$ cd $OGG_HOME1
[oracle@c<n>ds1 gghome_1]$ pwd
/u01/app/oracle/product/12.2.0/gghome_1
[oracle@c<n>ds1 gghome_1]$

[oracle@c<n>ds1 ~]$ cd $OGG_HOME2
[oracle@c<n>ds1 gghome_2]$ pwd
/u01/app/oracle/product/12.2.0/gghome_2
[oracle@c<n>ds1 gghome_2]$
```

- Use GGSCI, login to the downstream system and the primary node GoldenGate instances, in preparation to register the integrated capture - EDBN001.

On downstream system C<n>DS1A

```
[oracle@c<n>ds1 ~]$ cd $OGG_HOME1
[oracle@c<n>ds1 gghome_1]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.


GGSCI (c<n>ds1.oracle.com) 1> dblogin USERID ogg_admin@DBN0.oracle.com, PASSWORD
oracle
Successfully logged into database.
```

```
GGSCI (c<n>ds1.oracle.com) 2> miningdblogin userid
ogg_admin@C<n>DS1A.oracle.com, password oracle
Successfully logged into mining database.


GGSCI (c<n>ds1.oracle.com) 3>
```

- Execute the REGISTER command, it may take a short while to finish.

```
GGSCI (c<n>ds1.oracle.com) 3> register extract EDBN001, database
Extract EDBN001 successfully registered with database at SCN 2480085.


GGSCI (c<n>ds1.oracle.com) 4>
```

- Query DBA_CAPTURE to verify the REGISTER command successfully registered the extract on the primary database.

On downstream system C<n>DS1A

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

OGG$CAP_EDBN001
     2647448


SQL>
```

- Make sure the downstream OGG instance manager is running.

```
GGSCI (c<n>ds1.oracle.com) 5> start mgr
Manager started.


GGSCI (c<n>ds1.oracle.com) 6> info mgr

Manager is running (IP port c<n>ds1.oracle.com.7809, Process ID 4993).


GGSCI (c<n>ds1.oracle.com) 7>
```

- Before proceeding further, verify the extended node 1 OGG instance is running.

On extended node 1

```
[oracle@edbcn1 ~]$ cd $OGG_HOME
[oracle@edbcn1 gghome_1]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (edbcn1.oracle.com) 1> start mgr
Manager started.


GGSCI (edbcn1.oracle.com) 2> info mgr

Manager is running (IP port edbcn1.oracle.com.7809, Process ID 6062).


GGSCI (edbcn1.oracle.com) 3>
```

- Prepare the parameter file for the integrated extract EDBN001.

On downstream system C<n>DS1A

```
GGSCI (c<n>ds1.oracle.com) 12> edit params EDBN001
```

```
EXTRACT EDBN001
SETENV (ORACLE SID = C<n>DS1A)
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@DBN0.oracle.com, PASSWORD oracle
tranlogoptions mininguser ogg_admin@C<n>DS1A.oracle.com, miningpassword oracle
tranlogoptions integratedparams (max_sga_size 128, downstream_real_time_mine Y)
TRANLOGOPTIONS EXCLUDEUSER ogg admin
EXTTRAIL ./dirdat/sp
TABLE osm_repapi.*;
```

- View the parameter file for reference when defining the extract and trail. Specifying the environment variable TNS_ADMIN is necessary as there two oracle homes.

```
GGSCI (c<n>ds1.oracle.com) 2> view params EDBN001

EXTRACT EDBN001
SETENV (ORACLE SID = C<n>DS1A)
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@DBN0.oracle.com, PASSWORD oracle
tranlogoptions mininguser ogg admin@C<n>DS1A.oracle.com, miningpassword oracle
tranlogoptions integratedparams (max sga size 128, downstream real time mine Y)
TRANLOGOPTIONS EXCLUDEUSER ogg_admin
EXTTRAIL ./dirdat/sp
TABLE osm_repapi.*;
```

```
GGSCI (c<n>ds1.oracle.com) 3>
```

- Define the integrated extract and the local trail of the primary node, ./dirdat/sp.

```
GGSCI (c<n>ds1.oracle.com) 15> add extract EDBN001, integrated tranlog, begin
now
EXTRACT added.


GGSCI (c<n>ds1.oracle.com) 16> add exttrail ./dirdat/sp, extract EDBN001,
megabytes 100
EXTTRAIL added.


GGSCI (c<n>ds1.oracle.com) 17>
```

- Prepare the parameter file for the data pump extract PDBN001. The data pump process send data from the downstream system to extended node 1.

```
GGSCI (c<n>ds1.oracle.com) 17> edit params PDBN001
```

```
EXTRACT PDBN001
SETENV (ORACLE SID = C<n>DS1A)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn0.oracle.com, PASSWORD oracle
RMTHOST edbcn1.oracle.com, MGRPORT 7809
RMTTRAIL ./dirdat/tp
TABLE osm_repapi.*;
```

```
GGSCI (c<n>ds1.oracle.com) 18> view params PDBN001

EXTRACT PDBN001
SETENV (ORACLE_SID = C<n>DS1A)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome 1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg admin@dbn0.oracle.com, PASSWORD oracle
RMTHOST edbcn1.oracle.com, MGRPORT 7809
RMTTRAIL ./dirdat/tp
TABLE osm_repapi.*;



GGSCI (c<n>ds1.oracle.com) 19>
```

- Define the data pump extract and the remote trail of extended node 1, ./dirdat/tp. The remote trail size optional, the default size is 500M.

```
GGSCI (c<n>ds1.oracle.com) 19> add extract PDBN001, exttrailsource ./dirdat/sp
```

```
EXTRACT added.


GGSCI (c<n>ds1.oracle.com) 20> add rmttrail ./dirdat/tp, extract PDBN001,
megabytes 100
RMTTRAIL added.


GGSCI (c<n>ds1.oracle.com) 21>
```

- Show current status, do not start the extracts now.

```
GGSCI (c<n>ds1.oracle.com) 21> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     STOPPED     EDBN001     00:00:00      00:07:08
EXTRACT     STOPPED     PDBN001     00:00:00      00:00:52


GGSCI (c<n>ds1.oracle.com) 22>
```

- Because the default Replicat checkpoint is a database table, before defining the Replicat, the checkpoint table must be created as follow:

    1. Navigate to Oracle GoldenGate instance root directory
    2. Edit the file GLOBALS by adding the parameters: CHECKPOINTTABLE ogg_admin.chkpttab
    3. Start GGSCI
    4. Login to the database
    5. Create the checkpoint table by invoking the command: ADD CHECKPOINTTABLE

On extended node 1

```
[oracle@edbcn1 gghome_1]$ vi GLOBALS

CHECKPOINTTABLE ogg_admin.chkpttab
```

```
[oracle@edbcn1 gghome 1]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.
```

```
GGSCI (edbcn1.oracle.com) 1> dblogin userid ogg_admin@dbn1.oracle.com, password
oracle
Successfully logged into database.

GGSCI (edbcn1.oracle.com) 2> add checkpointtable

No checkpoint table specified. Using GLOBALS specification
(ogg admin.chkpttab)...

Successfully created checkpoint table ogg admin.chkpttab.

GGSCI (edbcn1.oracle.com) 3>
```

- Prepare the parameter file for the Replicat RDBN001.

On extended node 1

```
GGSCI (edbcn1.oracle.com) 4> edit params RDBN001
```

```
replicat RDBN001
SETENV (ORACLE_SID = DBN1)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn1.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn001.dsc, PURGE
MAP osm_repapi.*, TARGET osm_repapi.*;
```

- View the parameter file, define the Replicat using ADD REPLICAT command.

```
GGSCI (edbcn1.oracle.com) 2> view params RDBN001

replicat RDBN001
SETENV (ORACLE_SID = DBN1)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn1.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn001.dsc, PURGE
MAP osm_repapi.*, TARGET osm_repapi.*;



GGSCI (edbcn1.oracle.com) 3>
```

```
GGSCI (edbcn1.oracle.com) 3> add replicat RDBN001, exttrail ./dirdat/tp
REPLICAT added.


GGSCI (edbcn1.oracle.com) 4>
```

- Check current status

```
GGSCI (edbcn1.oracle.com) 4> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    STOPPED     RDBN001     00:00:00      00:01:11


GGSCI (edbcn1.oracle.com) 5>
```

On downstream system C<n>DS1A

```
GGSCI (c<n>ds1.oracle.com) 26> start *

Sending START request to MANAGER ...
EXTRACT EDBN001 starting

Sending START request to MANAGER ...
EXTRACT PDBN001 starting


GGSCI (c<n>ds1.oracle.com) 27> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN001     00:03:15      00:00:20
EXTRACT     RUNNING     PDBN001     00:00:00      00:00:20


GGSCI (c<n>ds1.oracle.com) 28>
```

On extended node 1

```
GGSCI (edbcn1.oracle.com) 2> start *

Sending START request to MANAGER ...
REPLICAT RDBN001 starting


GGSCI (edbcn1.oracle.com) 3> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN001     00:00:00      00:00:05


GGSCI (edbcn1.oracle.com) 4>
```

- Create the table-level supplemental log

On the primary node

```
[oracle@edbcn0 gghome_1]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (edbcn0.oracle.com) 1> dblogin userid ogg_admin@dbn0.oracle.com, password
oracle
Successfully logged into database.

GGSCI (edbcn0.oracle.com) 2> miningdblogin userid ogg_admin@c<n>ds1a.oracle.com,
password oracle
Successfully logged into mining database.

GGSCI (edbcn0.oracle.com) 3> add trandata osm_repapi.*

Logging of supplemental redo data enabled for table OSM REPAPI.CUSTOMERS.
TRANDATA for scheduling columns has been added on table 'OSM_REPAPI.CUSTOMERS'.
Logging of supplemental redo data enabled for table OSM_REPAPI.DISCOUNT_TYPES.
TRANDATA for scheduling columns has been added on table
'OSM REPAPI.DISCOUNT TYPES'.
2016-11-08 11:09:37  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.

Logging of supplemental redo data enabled for table OSM_REPAPI.OSM_DOCUMENTS.
TRANDATA for scheduling columns has been added on table
'OSM_REPAPI.OSM_DOCUMENTS'.
Logging of supplemental redo data enabled for table OSM REPAPI.POLICIES.
TRANDATA for scheduling columns has been added on table 'OSM_REPAPI.POLICIES'.
Logging of supplemental redo data enabled for table OSM_REPAPI.POLICY_TYPES.
TRANDATA for scheduling columns has been added on table
'OSM REPAPI.POLICY TYPES'.
Logging of supplemental redo data enabled for table OSM_REPAPI.RANDOM_DOMAIN.
TRANDATA for scheduling columns has been added on table
'OSM_REPAPI.RANDOM_DOMAIN'.
Logging of supplemental redo data enabled for table OSM_REPAPI.RANDOM_FIRST.
TRANDATA for scheduling columns has been added on table
'OSM_REPAPI.RANDOM_FIRST'.
Logging of supplemental redo data enabled for table OSM REPAPI.RANDOM LAST.
TRANDATA for scheduling columns has been added on table
'OSM_REPAPI.RANDOM_LAST'.
Logging of supplemental redo data enabled for table OSM_REPAPI.RANDOM_MIDDLE.
TRANDATA for scheduling columns has been added on table
'OSM_REPAPI.RANDOM_MIDDLE'.
GGSCI (edbcn0.oracle.com) 4>
```

Task 3: Configure Integrated Extract for Extended Node 1

Second, configure the unidirectional data replication from extended node 1 to the primary node. Figure 9 shows the data flow from extended node 1 to the primary node.

Extended Node 1 Hostname	edbcn1.oracle.com
Global database name	DBN1.oracle.com
Downstream System Hostname	c<n>ds1.oracle.com
Global database name	C<n>DS1B.oracl.e.com

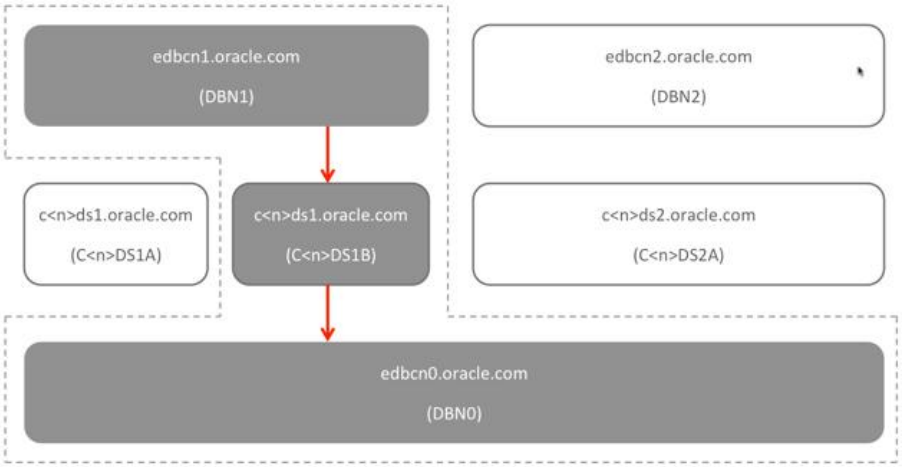


Figure 9: Extended node 1 ▸ Downstream C<n>DS1B ▸ Primary node

The Oracle GoldenGate 12c instance extended node 1 operated using the following set of processes and trails.

GoldenGate Process	Process Description
EDBN101	Integrated Extract for DBN1
./dirdat/s1	Local trail for the extended node 1
PDBN101	Data pump Extract
./dirdat/t1	Remote Trail for the extended node 1
RDBN101	The Replicat on the primary node

The steps to configure the integrated capture:

- Login to the downstream system and the extended node 1
- Register the integrated extract - EDBN101

- Prepare and define the integrated extract and the local trail (s1)
- Prepare and define the data pump and the remote trail (t1)
- Prepare and Define the Replicat on the primary node
- Add table-level supplemental logging

The downstream system has two GoldenGate instances. For unidirectional from the primary to the extended node 1, use $OGG_HOME1, and from the extended node 1 to the primary, use $OGG_HOME2.

```
[oracle@c<n>s1 ~]$ cd $OGG_HOME1
[oracle@c<n>ds1 gghome_1]$ pwd
/u01/app/oracle/product/12.2.0/gghome_1
[oracle@c<n>ds1 gghome_1]$

[oracle@c<n>ds1 ~]$ cd $OGG_HOME2
[oracle@c<n>ds1 gghome_2]$ pwd
/u01/app/oracle/product/12.2.0/gghome_2
[oracle@c<n>ds1 gghome_2]$
```

- Start GGSCI, login to the downstream system and the extended node 1 GoldenGate instances, in preparation to register the integrated capture – EDBN101.

On downstream system C<n>DS1B

```
[oracle@c1ds1 ~]$ . oraenv
ORACLE_SID = [C1DS1A] ? C1DS1B
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@c1ds1 ~]$ echo $ORACLE_HOME
/u01/app/oracle/product/12.1.0/dbhome_2
[oracle@c1ds1 ~]$ cd $OGG_HOME2
```

```
[oracle@c<n>ds1 admin]$ cd $OGG_HOME2
[oracle@c<n>ds1 gghome_1]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (c<n>ds1.oracle.com) 1> dblogin userid ogg_admin@dbn1.oracle.com, password
oracle
Successfully logged into database.

GGSCI (c<n>ds1.oracle.com) 2> miningdblogin userid
ogg_admin@c<n>ds1b.oracle.com, password oracle
Successfully logged into mining database.

GGSCI (c<n>ds1.oracle.com) 3>
```

- Execute the REGISTER command, it may take a short while to finish.

```
GGSCI (c<n>ds1.oracle.com) 3> register extract EDBN101, database
Extract EDBN101 successfully registered with database at SCN 2124885.


GGSCI (c<n>ds1.oracle.com) 4>
```

- Make sure the primary node 1 OGG instance manager is running. If not, start it.

On the primary system

```
[oracle@edbcn1 ~]$ cd $OGG_HOME
[oracle@edbcn1 gghome_1]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (edbcn1.oracle.com) 1> start mgr
Manager started.


GGSCI (edbcn1.oracle.com) 2> info mgr

Manager is running (IP port edbcn1.oracle.com.7809, Process ID 6717).


GGSCI (edbcn1.oracle.com) 3>
```

- Make sure the downstream (C<n>DS1B) OGG instance manager is running. If not, start it.

On downstream system C<n>DS1B

```
GGSCI (c1ds1.oracle.com) 4> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     STOPPED


GGSCI (c1ds1.oracle.com) 5> start mgr
Manager started.


GGSCI (c1ds1.oracle.com) 6> info mgr
```

```
Manager is running (IP port c1ds1.oracle.com.7810, Process ID 20588).


GGSCI (c1ds1.oracle.com) 7>
```

- Prepare the parameter file for the integrated extract EDBN001.

On downstream system C<n>DS1B

```
GGSCI (edbcn1.oracle.com) 2> edit params EDBN101
```

```
EXTRACT EDBN101
SETENV (ORACLE_SID = C<n>DS1B)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_2")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
SETENV (TNS_ADMIN = "/u01/app/oracle/product/12.1.0/dbhome_1/network/admin")
USERID ogg_admin@DBN1.oracle.com, PASSWORD oracle
tranlogoptions mininguser ogg_admin@C<n>DS1B.oracle.com, miningpassword oracle
tranlogoptions integratedparams (max_sga_size 128, downstream_real_time_mine Y)
TRANLOGOPTIONS EXCLUDEUSER ogg admin
EXTTRAIL ./dirdat/s1
TABLE osm_repapi.*;
```

- View the parameter file for reference when defining the extract and trail. Specifying the environment variable TNS_ADMIN is necessary as there two oracle homes.

```
GGSCI (c<n>ds1.oracle.com) 3> view params EDBN101

EXTRACT EDBN101
SETENV (ORACLE_SID = C<n>DS1B)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_2")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
SETENV (TNS_ADMIN = "/u01/app/oracle/product/12.1.0/dbhome_1/network/admin")
USERID ogg_admin@DBN1.oracle.com, PASSWORD oracle
tranlogoptions mininguser ogg admin@C<n>DS1B.oracle.com, miningpassword oracle
tranlogoptions integratedparams (max_sga_size 128, downstream_real_time_mine Y)
TRANLOGOPTIONS EXCLUDEUSER ogg_admin
EXTTRAIL ./dirdat/s1
TABLE osm_repapi.*;



GGSCI (c<n>ds1.oracle.com) 4>
```

- Define the integrated extract and the local trail of the extended node 1, ./dirdat/s1.

```
GGSCI (c<n>ds1.oracle.com) 4> add extract EDBN101, integrated tranlog, begin now
EXTRACT added.


GGSCI (c<n>ds1.oracle.com) 5> add exttrail ./dirdat/s1, extract EDBN101,
megabytes 100
```

```
EXTTRAIL added.


GGSCI (c<n>ds1.oracle.com) 6>
```

- Prepare the parameter file for the data pump extract PDBN101.

```
GGSCI (c<n>ds1.oracle.com) 6> edit params PDBN101
```

```
EXTRACT PDBN101
SETENV (ORACLE_SID = C<n>DS1B)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_2")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn1.oracle.com, PASSWORD oracle
RMTHOST edbcn0.oracle.com, MGRPORT 7809
RMTTRAIL ./dirdat/t1
TABLE osm_repapi.*;
```

```
GGSCI (c<n>ds1.oracle.com) 7> view params PDBN101

EXTRACT PDBN101
SETENV (ORACLE_SID = C<n>DS1B)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_2")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn1.oracle.com, PASSWORD oracle
RMTHOST edbcn0.oracle.com, MGRPORT 7809
RMTTRAIL ./dirdat/t1
TABLE osm_repapi.*;



GGSCI (c<n>ds1.oracle.com) 8>
```

- Define the data pump extract and the remote trail of the extended node 1, ./dirdat/t1. The remote trail size optional, the default size is 500M.

```
GGSCI (c<n>ds1.oracle.com) 9> add extract PDBN101, exttrailsource ./dirdat/s1
EXTRACT added.

GGSCI (c<n>ds1.oracle.com) 11> add rmttrail ./dirdat/t1, extract PDBN101,
megabytes 100
RMTTRAIL added.


GGSCI (c<n>ds1.oracle.com) 12>
```

- Show current status, do not start the extracts now.

```
GGSCI (c<n>ds1.oracle.com) 12> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
```

```
EXTRACT     STOPPED     EDBN101     00:00:00      00:05:13
EXTRACT     STOPPED     PDBN101     00:00:00      00:01:15


GGSCI (c<n>ds1.oracle.com) 13>
```

- Because the default Replicat checkpoint is a database table, before defining the Replicat, the checkpoint table must be created as follow:

    6. Navigate to Oracle GoldenGate instance root directory
    7. Edit the file GLOBALS by adding the parameters: CHECKPOINTTABLE ogg_admin.chkpttab
    8. Start GGSCI
    9. Login to the database
    10. Create the checkpoint table by invoking the command: ADD CHECKPOINTTABLE

On the primary node

```
[oracle@edbcn0 gghome_1]$ vi GLOBALS

CHECKPOINTTABLE ogg_admin.chkpttab
```

```
[oracle@edbcn0 gghome_1]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (edbcn0.oracle.com) 1> dblogin userid ogg_admin@dbn0.oracle.com, password
oracle
Successfully logged into database.

GGSCI (edbcn0.oracle.com) 2> add checkpointtable

No checkpoint table specified. Using GLOBALS specification
(ogg_admin.chkpttab)...

Successfully created checkpoint table ogg_admin.chkpttab.

GGSCI (edbcn0.oracle.com) 3>
```

- Prepare the parameter file for the Replicat RDBN101.

```
GGSCI (edbcn1.oracle.com) 3> edit params RDBN101
```

```
replicat RDBN101
SETENV (ORACLE_SID = DBN0)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn0.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn101.dsc, PURGE
MAP osm_repapi.*, TARGET osm_repapi.*;
```

- View the parameter file, define the Replicat using ADD REPLICAT command.

```
GGSCI (edbcn0.oracle.com) 6> view params RDBN101

replicat RDBN101
SETENV (ORACLE_SID = DBN0)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn0.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn101.dsc, PURGE
MAP osm_repapi.*, TARGET osm_repapi.*;



GGSCI (edbcn0.oracle.com) 7>
```

```
GGSCI (edbcn0.oracle.com) 7> add replicat RDBN101, exttrail ./dirdat/t1
REPLICAT added.


GGSCI (edbcn0.oracle.com) 8>
```

- Check current status

```
GGSCI (edbcn1.oracle.com) 6> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    STOPPED     RDBN101     00:00:00      00:00:22


GGSCI (edbcn1.oracle.com) 7>
```

On the downstream system

```
GGSCI (c<n>ds1.oracle.com) 13> start *

Sending START request to MANAGER ...
EXTRACT EDBN101 starting

Sending START request to MANAGER ...
```

```
EXTRACT PDBN101 starting


GGSCI (c<n>ds1.oracle.com) 14> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     STARTING    EDBN101     00:00:00      00:12:46
EXTRACT     RUNNING     PDBN101     00:00:00      00:08:48


GGSCI (c<n>ds1.oracle.com) 15>
```

On the primary node

```
GGSCI (edbcn1.oracle.com) 14> start *

Sending START request to MANAGER ...
REPLICAT RDBN101 starting


GGSCI (edbcn1.oracle.com) 15> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN101     00:00:00      00:00:00


GGSCI (edbcn1.oracle.com) 16>
```

- Create the table-level supplemental log

On extended node 1

```
GGSCI (edbcn1.oracle.com) 16> dblogin userid ogg_admin@dbn1.oracle.com, password
oracle
Successfully logged into database.

GGSCI (edbcn1.oracle.com) 17> miningdblogin userid
ogg_admin@c<n>ds1b.oracle.com, password oracle
Successfully logged into mining database.

GGSCI (edbcn1.oracle.com) 18> add trandata osm_repapi.*

Logging of supplemental redo data enabled for table OSM REPAPI.CUSTOMERS.
TRANDATA for scheduling columns has been added on table 'OSM_REPAPI.CUSTOMERS'.
Logging of supplemental redo data enabled for table OSM_REPAPI.DISCOUNT_TYPES.
TRANDATA for scheduling columns has been added on table
'OSM REPAPI.DISCOUNT TYPES'.
2016-11-08 17:10:41  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.
```

```
Logging of supplemental redo data enabled for table OSM_REPAPI.OSM_DOCUMENTS.
TRANDATA for scheduling columns has been added on table
'OSM REPAPI.OSM DOCUMENTS'.
Logging of supplemental redo data enabled for table OSM_REPAPI.POLICIES.
TRANDATA for scheduling columns has been added on table 'OSM_REPAPI.POLICIES'.
Logging of supplemental redo data enabled for table OSM_REPAPI.POLICY_TYPES.
TRANDATA for scheduling columns has been added on table
'OSM_REPAPI.POLICY_TYPES'.
Logging of supplemental redo data enabled for table OSM REPAPI.RANDOM DOMAIN.
TRANDATA for scheduling columns has been added on table
'OSM REPAPI.RANDOM DOMAIN'.
Logging of supplemental redo data enabled for table OSM_REPAPI.RANDOM_FIRST.
TRANDATA for scheduling columns has been added on table
'OSM_REPAPI.RANDOM_FIRST'.
Logging of supplemental redo data enabled for table OSM REPAPI.RANDOM LAST.
TRANDATA for scheduling columns has been added on table
'OSM_REPAPI.RANDOM_LAST'.
Logging of supplemental redo data enabled for table OSM_REPAPI.RANDOM_MIDDLE.
TRANDATA for scheduling columns has been added on table
'OSM_REPAPI.RANDOM_MIDDLE'.
GGSCI (edbcn1.oracle.com) 19>
```

Task 4: Configure Integrated Extract for Extended Node 2

Extended Node 2 Hostname	edbcn2.oracle.com
Global database name (11g)	DBN2.oracle.com
Downstream System Hostname	c<n>ds2.oracle.com
Global database name	C<n>DS2A.oracl.e.com

Last, configure the unidirectional data replication from extended node 2 to the primary node as illustrated on Figure 10.

Remarks related to extende node 2:

- Extended node 2 is running Oracle Database 11g, its associated downstream system is running on Oracle Database 12c. This is a valid configuration and there should not be a compatibility issue for operating on integerated capture mode.

- When configuring the Recovery Manager (RMAN) for backup, we will use Oracl e Database 11g client, until it extnded node 2 upgraded and integerated with Oracl e Database 12c.

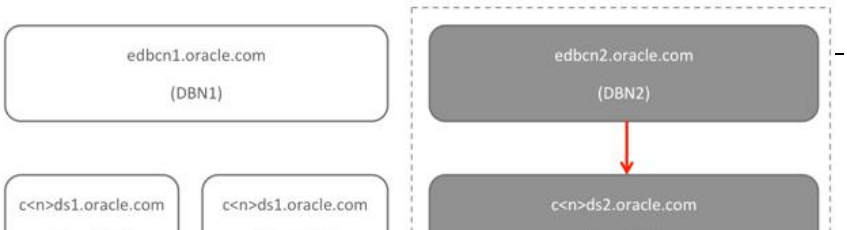


Figure 10: Extended node 2 ◗ Downstream C<n>DS2A ◗ Primary node

Oracle GoldenGate 12c instance extended node 2 operated using the following set of processes and trails.

GoldenGate Process	Process Description
EDBN201	Integrated Extract for DBN2
./dirdat/s2	Local trail for the extended node 2
PDBN201	Data pump Extract
./dirdat/t2	Remote Trail for the extended node 2
RDBN201	The Replicat on the primary node

The steps to configure the integrated capture:

- Login to the downstream system and the extended node 2
- Register the integrated extract – EDBN201
- Prepare and define the integrated extract and the local trail (s2)
- Prepare and define the data pump and the remote trail (t2)
- Prepare and Define the Replicat on the primary node
- Add table-level supplemental logging

The downstream system has one GoldenGate instances. For unidirectional from the extended 2 to the primary node, use $OGG_HOME1.

```
[oracle@c<n>ds1 ~]$ cd $OGG_HOME1
[oracle@c<n>ds1 gghome_1]$ pwd
/u01/app/oracle/product/12.2.0/gghome_1
[oracle@c<n>ds1 gghome_1]$
```

- Start GGSCI, login to the downstream system and the extended node 2 GoldenGate instances, in preparation to register the integrated capture – EDBN101.

On downstream system C<n>DS2A

```
[oracle@c<n>ds2 ~]$ cd $OGG_HOME1
[oracle@c<n>ds2 gghome_1]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (c<n>ds2.oracle.com) 1> dblogin userid ogg_admin@dbn2.oracle.com, password
oracle
Successfully logged into database.

GGSCI (c<n>ds2.oracle.com) 2> miningdblogin userid
ogg_admin@c<n>ds2a.oracle.com, password oracle
Successfully logged into mining database.

GGSCI (c<n>ds2.oracle.com) 3>
```

- Execute the REGISTER command, it may take a short while to finish. Query the database dictionary view DBA_CAPTURE to display more details.

```
GGSCI (c<n>ds2.oracle.com) 3> register extract EDBN201, database
Extract EDBN201 successfully registered with database at SCN 1995391.


GGSCI (c<n>ds2.oracle.com) 4>
```

- Make sure the downstream OGG instance manager is running.

```
GGSCI (c<n>ds2.oracle.com) 4> start mgr
Manager started.


GGSCI (c<n>ds2.oracle.com) 5> info mgr

Manager is running (IP port c<n>ds2.oracle.com.7809, Process ID 18032).


GGSCI (c<n>ds2.oracle.com) 6>
```

- Prepare the parameter file for the integrated extract EDBN201.

On downstream system C<n>DS2A

```
GGSCI (c<n>ds2.oracle.com) 6> edit params EDBN201
```

```
EXTRACT EDBN201
SETENV (ORACLE_SID = C<n>DS2A)
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@DBN2.oracle.com, PASSWORD oracle
tranlogoptions mininguser ogg_admin@C<n>DS2A.oracle.com, miningpassword oracle
tranlogoptions integratedparams (max sga size 128, downstream real time mine Y)
TRANLOGOPTIONS EXCLUDEUSER ogg_admin
EXTTRAIL ./dirdat/s2
TABLE osm_repapi.*;
```

- View the parameter file for reference when defining the extract and trail. Specifying the environment variable TNS_ADMIN is necessary as there two oracle homes.

```
GGSCI (c<n>ds2.oracle.com) 7> view params EDBN201

EXTRACT EDBN201
SETENV (ORACLE_SID = C<n>DS2A)
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@DBN2.oracle.com, PASSWORD oracle
tranlogoptions mininguser ogg_admin@C<n>DS2A.oracle.com, miningpassword oracle
tranlogoptions integratedparams (max_sga_size 128, downstream_real_time_mine Y)
TRANLOGOPTIONS EXCLUDEUSER ogg_admin
EXTTRAIL ./dirdat/s2
TABLE osm_repapi.*;



GGSCI (c<n>ds2.oracle.com) 8>
```

- Define the integrated extract and the local trail of the extended node 2, ./dirdat/s2.

```
GGSCI (c<n>ds2.oracle.com) 8> add extract EDBN201, integrated tranlog, begin now
EXTRACT added.


GGSCI (c<n>ds2.oracle.com) 9> add exttrail ./dirdat/s2, extract EDBN201,
megabytes 100
EXTTRAIL added.


GGSCI (c<n>ds2.oracle.com) 10>
```

- Prepare the parameter file for the data pump extract PDBN201.

```
GGSCI (c<n>ds2.oracle.com) 10> edit params PDBN201
```

```
EXTRACT PDBN201
SETENV (ORACLE_SID = C<n>DS2A)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn2.oracle.com, PASSWORD oracle
RMTHOST edbcn0.oracle.com, MGRPORT 7809
RMTTRAIL ./dirdat/t2
TABLE osm_repapi.*;
```

```
GGSCI (c<n>ds2.oracle.com) 11> view params PDBN201

EXTRACT PDBN201
SETENV (ORACLE_SID = C<n>DS2A)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn2.oracle.com, PASSWORD oracle
RMTHOST edbcn0.oracle.com, MGRPORT 7809
RMTTRAIL ./dirdat/t2
TABLE osm_repapi.*;



GGSCI (c<n>ds2.oracle.com) 12>
```

- Define the data pump extract and the remote trail of the extended node 2, ./dirdat/t2. The remote trail size optional, the default size is 500M.

```
GGSCI (c<n>ds2.oracle.com) 12> add extract PDBN201, exttrailsource ./dirdat/s2
EXTRACT added.


GGSCI (c<n>ds2.oracle.com) 13> add rmttrail ./dirdat/t2, extract PDBN201,
megabytes 100
RMTTRAIL added.


GGSCI (c<n>ds2.oracle.com) 14>
```

- Show current status, do not start the extracts now.

```
GGSCI (c<n>ds2.oracle.com) 14> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     STOPPED     EDBN201     00:00:00      00:03:45
EXTRACT     STOPPED     PDBN201     00:00:00      00:00:46


GGSCI (c<n>ds2.oracle.com) 15>
```

- Prepare the parameter file for the Replicat RDBN201.

On the primary node

```
[oracle@edbcn0 ~]$ cd $OGG_HOME
[oracle@edbcn0 gghome_1]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (edbcn0.oracle.com) 1>
```

```
GGSCI (edbcn0.oracle.com) 1> edit params RDBN201
```

```
replicat RDBN201
SETENV (ORACLE_SID = DBN0)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg admin@dbn0.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn201.dsc, PURGE
MAP osm_repapi.*, TARGET osm_repapi.*;
```

- View the parameter file, define the Replicat using ADD REPLICAT command.

```
GGSCI (edbcn0.oracle.com) 2> view params RDBN201

replicat RDBN201
SETENV (ORACLE SID = DBN0)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS LANG = AMERICAN AMERICA.AL32UTF8)
USERID ogg_admin@dbn0.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn201.dsc, PURGE
MAP osm repapi.*, TARGET osm repapi.*;



GGSCI (edbcn0.oracle.com) 3>
```

```
GGSCI (edbcn0.oracle.com) 3> add replicat RDBN201, exttrail ./dirdat/t2
REPLICAT added.


GGSCI (edbcn0.oracle.com) 4>
```

- Check current status

On downstream system C<n>DS2A

```
GGSCI (c<n>ds2.oracle.com) 15> start *

Sending START request to MANAGER ...
EXTRACT EDBN201 starting

Sending START request to MANAGER ...
EXTRACT PDBN201 starting


GGSCI (c<n>ds2.oracle.com) 16> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN201     00:00:00      00:12:05
EXTRACT     RUNNING     PDBN201     00:00:00      00:09:06


GGSCI (c<n>ds2.oracle.com) 17>
```

On the primary node

```
GGSCI (edbcn0.oracle.com) 8> start *

Sending START request to MANAGER ...
REPLICAT RDBN001 starting

Sending START request to MANAGER ...
REPLICAT RDBN201 starting


GGSCI (edbcn0.oracle.com) 9> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN001     00:00:00      00:00:00
REPLICAT    RUNNING     RDBN201     00:00:00      00:00:08


GGSCI (edbcn0.oracle.com) 10>
```

- Create the table-level supplemental log

On extended node 2

```
[oracle@edbcn2 ~]$ cd $OGG_HOME
[oracle@edbcn2 gghome_1]$ ./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 (edbcn2.oracle.com) 1>
```

```
GGSCI (edbcn2.oracle.com) 12> dblogin userid ogg_admin@dbn2.oracle.com, password
oracle
Successfully logged into database.

GGSCI (edbcn2.oracle.com) 13> miningdblogin userid
ogg_admin@c<n>ds2a.oracle.com, password oracle
Successfully logged into mining database.

GGSCI (edbcn2.oracle.com) 14> add trandata osm_repapi.*

Logging of supplemental redo data enabled for table OSM_REPAPI.CUSTOMERS.
TRANDATA for scheduling columns has been added on table 'OSM_REPAPI.CUSTOMERS'.
Logging of supplemental redo data enabled for table OSM REPAPI.DISCOUNT TYPES.
TRANDATA for scheduling columns has been added on table
'OSM REPAPI.DISCOUNT TYPES'.
2016-11-08 17:46:33  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.

Logging of supplemental redo data enabled for table OSM_REPAPI.OSM_DOCUMENTS.
TRANDATA for scheduling columns has been added on table
'OSM_REPAPI.OSM_DOCUMENTS'.
Logging of supplemental redo data enabled for table OSM_REPAPI.POLICIES.
TRANDATA for scheduling columns has been added on table 'OSM_REPAPI.POLICIES'.
Logging of supplemental redo data enabled for table OSM REPAPI.POLICY TYPES.
TRANDATA for scheduling columns has been added on table
'OSM REPAPI.POLICY TYPES'.
Logging of supplemental redo data enabled for table OSM_REPAPI.RANDOM_DOMAIN.
TRANDATA for scheduling columns has been added on table
'OSM_REPAPI.RANDOM_DOMAIN'.
Logging of supplemental redo data enabled for table OSM REPAPI.RANDOM FIRST.
TRANDATA for scheduling columns has been added on table
'OSM REPAPI.RANDOM FIRST'.
Logging of supplemental redo data enabled for table OSM_REPAPI.RANDOM_LAST.
TRANDATA for scheduling columns has been added on table
'OSM REPAPI.RANDOM LAST'.
Logging of supplemental redo data enabled for table OSM_REPAPI.RANDOM_MIDDLE.
TRANDATA for scheduling columns has been added on table
'OSM_REPAPI.RANDOM_MIDDLE'.
GGSCI (edbcn2.oracle.com) 15>
```

Task 5: Perform EDBC Oracle GoldenGate 12c instance Review

Review Oracle GoldenGate 12c instance status. At this stage there are three unidirectional data replication.

1. Primary node to extended node 1
2. Extended node 1 to primary node
3. Extended node 2 to primary Node

If any of the Oracle GoldenGate 12c process is not on RUNNING status, investigate, correct and check status.

Primary Node

```
[oracle@edbcn0 gghome_1]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (edbcn0.oracle.com) 1> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN101     00:00:00      00:00:07
REPLICAT    RUNNING     RDBN201     00:00:00      00:00:09


GGSCI (edbcn0.oracle.com) 2>
```

Process Name	Process Description	Running Status
Manager	Manager	☐
RDBN101	Replicat for Node 1	☐
RDBN201	Replicat for Node 2	☐

Extended Node 1

```
[oracle@edbcn1 ~]$ cd $OGG_HOME
[oracle@edbcn1 gghome_1]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.
```

```
GGSCI (edbcn1.oracle.com) 1> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN001     00:00:00      00:00:05


GGSCI (edbcn1.oracle.com) 2>
```

Process Name	Process Description	Running Status
Manager	Manager	☐
RDBN001	Replicat for primary node	☐

Extended Node 2

```
[oracle@edbcn2 ~]$ cd $OGG_HOME
[oracle@edbcn2 gghome_1]$ ./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 (edbcn2.oracle.com) 1> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING


GGSCI (edbcn2.oracle.com) 2>
```

Process Name	Process Description	Running Status
Manager	Manager	☐

Downstream System 1A (C<n>DS1A)

```
[oracle@c1ds1 ~]$ cd $OGG_HOME1
[oracle@c1ds1 gghome_1]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.
```

```
Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (c1ds1.oracle.com) 1> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN001     00:00:25      00:00:03
EXTRACT     RUNNING     PDBN001     00:00:00      00:00:03


GGSCI (c1ds1.oracle.com) 2>
```

Process Name	Process Description	Running Status
Manager	Manager	☐
EDBN001	Integrated Extract for the primary node	☐
PDBN001	Data Pump for the primary node	☐

Downstream System 1B (C<n>DS1B)

```
[oracle@c1ds1 gghome_1]$ . oraenv
ORACLE_SID = [C1DS1A] ? C1DS1B
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@c1ds1 gghome_1]$ cd $OGG_HOME2
[oracle@c1ds1 gghome_2]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (c1ds1.oracle.com) 1> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN101     00:00:30      00:00:06
EXTRACT     RUNNING     PDBN101     00:00:00      00:00:05


GGSCI (c1ds1.oracle.com) 2>
```

Process Name	Process Description	Running Status
Manager	Manager	☐

EDBN101	Integrated Extract for extended 1 node	☐
PDBN101	Data Pump for extended node 1	☐

Downstream System 2A (C<n>DS2A)

```
[oracle@c1ds2 gghome_1]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (c1ds2.oracle.com) 1> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN201     00:00:25      00:00:00
EXTRACT     RUNNING     PDBN201     00:00:00      00:00:03


GGSCI (c1ds2.oracle.com) 2>
```

Process Name	Process Description	Running Status
Manager	Manager	☐
EDBN201	Integrated Extract for extended 2 node	☐
PDBN201	Data Pump for extended node 2	☐

Task 6: Perform EDBC Transactions Testing

There will be three basic SQL INSERT statements performed to test the integrated capture, data pump and replicat are working as expected.

Test 1: Insert unidirectional from the primary to extended node 1
Test 2: Insert unidirectional from extended node 1 to the primary node
Test 3: Insert unidirectional from extended node 2 to the primary node

Test 1: Insert unidirectional from the primary to extended node 1

Login to the primary node, perform a simple INSERT DML statement.

On the primary node

```
SQL> conn osm_repapi/oracle@dbn0.oracle.com
Connected.
SQL> insert into POLICY_TYPES
  2  values ('FJ', 'Figher Jet');

1 row created.

SQL> commit;

Commit complete.

SQL>
```

Login to the extended node 1, verify data replication is correct.

On the extended node 1

```
[oracle@edbcn1 gghome 1]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Mon Dec 26 04:43:55 2016

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

SQL> conn osm_repapi/oracle@dbn1.oracle.com
Connected.
SQL> select * from POLICY_TYPES
  2  where pt_code='FJ';

PT CODE
--------
PT_DESC
--------------------------------------------------------------------------------
FJ
Figher Jet


SQL>
```

Test 2: Insert unidirectional from extended node 1 to the primary node

Login to extended node 1, perform a simple INSERT DML statement.

On extended node 1

```
SQL> conn osm_repapi/oracle@dbn1.oracle.com
Connected.
SQL> insert into POLICY_TYPES
  2  values ('PJ', 'Private Jet');

1 row created.
```

```
SQL> commit;

Commit complete.

SQL>
```

On the Primary node

```
SQL> select * from POLICY_TYPES
  2  where PT_CODE='PJ';

PT_CODE
--------
PT DESC
--------------------------------------------------------------------------------
PJ
Private Jet


SQL>
```

Test 3: Insert unidirectional from extended node 2 to the primary node

The test insert 1 row on extended node 2 and verify the row exist on the primary node.

On extended node 2

```
[oracle@edbcn2 gghome_1]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sat Nov 12 09:08:15 2016

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

SQL> conn osm_repapi/oracle@dbn2.oracle.com
Connected.
SQL>
```

```
SQL> desc POLICY_TYPES
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PT_CODE                                   NOT NULL VARCHAR2(8)
 PT_DESC                                   NOT NULL VARCHAR2(128)

SQL> insert into POLICY_TYPES
  2  values ('CA', 'Commercial Airplane');

1 row created.

SQL> commit;

Commit complete.
```

```
SQL>
```

On the primary node

```
[oracle@edbcn0 gghome_1]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sat Nov 12 09:11:17 2016

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

SQL> conn osm_repapi/oracle@dbn0.oracle.com;
Connected.
SQL> select * from POLICY_TYPES
  2  where PT_CODE='CA';

PT_CODE
--------
PT_DESC
--------------------------------------------------------------------------------
CA
Commercial Airplane


SQL>
```

Workshop 5: Configuring EDBC Data Streaming

Data Streaming perform real-time data replication from the trail generated at the downstream systems to extended node 1 and extended node 2.

Tale 9 lists Workshop 5 tasks.

Task No	Task Description
1	Configure Data Streaming of the primary node
2	Configure Data Streaming of extended node 1
3	Configure Data Streaming of extended node 2
4	Perform basic Data Streaming tests

Table 9: Workshop 5 Tasks

Task 1: Configure the Primary Node Data Streaming

The data streaming from the downstream system C<n>DS1A to extended node 2. The configuration defines the following:

- A data pump (PDBN002) on the downstream (C<n>DS1A) for data streaming to extended node 2
- A remote trail, ./dirdat/tp

On downstream system C<n>DS1A

```
[oracle@c1ds1 ~]$ cd $OGG_HOME1
[oracle@c1ds1 gghome_1]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (c1ds1.oracle.com) 1>
```

```
GGSCI (c<n>ds1.oracle.com) 1> edit params PDBN002
```

```
EXTRACT PDBN002
SETENV (ORACLE_SID = C<n>DS1A)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
```

```
USERID ogg_admin@dbn0.oracle.com, PASSWORD oracle
RMTHOST edbcn2.oracle.com, MGRPORT 7809
RMTTRAIL ./dirdat/tp
TABLE osm_repapi.*;
```

```
GGSCI (c<n>ds1.oracle.com) 3> view params PDBN002

EXTRACT PDBN002
SETENV (ORACLE_SID = C<n>DS1A)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn0.oracle.com, PASSWORD oracle
RMTHOST edbcn2.oracle.com, MGRPORT 7809
RMTTRAIL ./dirdat/tp
TABLE osm_repapi.*;



GGSCI (c<n>ds1.oracle.com) 4>
```

Define the data streaming process and its associated remote trail.

```
GGSCI (c<n>ds1.oracle.com) 4> add extract PDBN002, exttrailsource ./dirdat/sp
EXTRACT added.


GGSCI (c<n>ds1.oracle.com) 5> add rmttrail ./dirdat/tp, extract PDBN002,
megabytes 100
RMTTRAIL added.


GGSCI (c<n>ds1.oracle.com) 6>
```

On downstream system C<n>DS1A

```
GGSCI (c<n>ds1.oracle.com) 2> start PDBN002

Sending START request to MANAGER ...
EXTRACT PDBN002 starting


GGSCI (c<n>ds1.oracle.com) 3> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN001     00:00:08      00:00:03
EXTRACT     ABENDED     PDBN001     00:00:00      00:37:48
EXTRACT     RUNNING     PDBN002     00:00:00      00:10:09


GGSCI (c<n>ds1.oracle.com) 4>
```

Task 2: Configure Extended Node 1 Data Streaming

The data streaming from the downstream system C<n>DS1B to the extended node 2. The configuration define the following:

- A data pump (PDBN002) on the downstream (C<n>DS1B) for data streaming to extended node 2
- A remote trail ./dirdat/t1

On downstream system C<n>DS1B

```
[oracle@c1ds1 gghome 1]$ . oraenv
ORACLE_SID = [C1DS1A] ? C1DS1B
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@c1ds1 gghome_1]$ cd $OGG_HOME2
[oracle@c1ds1 gghome_2]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (c1ds1.oracle.com) 1>
```

```
GGSCI (c<n>ds1.oracle.com) 1> edit params pdbn102
```

```
EXTRACT PDBN102
SETENV (ORACLE_SID = C<n>DS1B)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_2")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn1.oracle.com, PASSWORD oracle
RMTHOST edbcn2.oracle.com, MGRPORT 7809
RMTTRAIL ./dirdat/t1
TABLE osm_repapi.*;
```

```
GGSCI (c<n>ds1.oracle.com) 2> view params pdbn102

EXTRACT PDBN102
SETENV (ORACLE_SID = C<n>DS1B)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_2")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn1.oracle.com, PASSWORD oracle
RMTHOST edbcn2.oracle.com, MGRPORT 7809
RMTTRAIL ./dirdat/t1
TABLE osm_repapi.*;
```

```
GGSCI (c<n>ds1.oracle.com) 3>
```

Define the data streaming process and its associated remote trail.

```
GGSCI (c<n>ds1.oracle.com) 3> add extract PDBN102, exttrailsource ./dirdat/s1
EXTRACT added.


GGSCI (c<n>ds1.oracle.com) 4> add rmttrail ./dirdat/t1, extract PDBN102,
megabytes 100
RMTTRAIL added.


GGSCI (c<n>ds1.oracle.com) 5>
```

On downstream system C<n>DS1B

```
GGSCI (c<n>ds1.oracle.com) 11> start PDBN102

Sending START request to MANAGER ...
EXTRACT PDBN102 starting


GGSCI (c<n>ds1.oracle.com) 12> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN101     00:00:13      00:00:05
EXTRACT     RUNNING     PDBN101     00:00:00      00:03:41
EXTRACT     RUNNING     PDBN102     00:00:00      00:12:52


GGSCI (c<n>ds1.oracle.com) 13>
```

Task 3: Configure Extended Node 2 Data Streaming

The data streaming from the downstream system C<n>DS2A to the extended node 2. The configuration define the following:

- A data pump (PDBN202) on the downstream (C<n>DS2A) for data streaming to extended node 1
- A remote trail ./dirdat/t2

On downstream system C<n>DS2A

```
GGSCI (c<n>ds2.oracle.com) 2> edit params PDBN202
```

```
EXTRACT PDBN202
SETENV (ORACLE_SID = C<n>DS2A)
```

```
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg admin@dbn2.oracle.com, PASSWORD oracle
RMTHOST edbcn1.oracle.com, MGRPORT 7809
RMTTRAIL ./dirdat/t2
TABLE osm_repapi.*;
```

```
GGSCI (c<n>ds2.oracle.com) 1> view params pdbn202

EXTRACT PDBN202
SETENV (ORACLE SID = C<n>DS2A)
SETENV (ORACLE HOME = "/u01/app/oracle/product/12.1.0/dbhome 1")
SETENV (NLS LANG = AMERICAN AMERICA.AL32UTF8)
USERID ogg_admin@dbn2.oracle.com, PASSWORD oracle
RMTHOST edbcn1.oracle.com, MGRPORT 7809
RMTTRAIL ./dirdat/t2
TABLE osm repapi.*;



GGSCI (c<n>ds2.oracle.com) 2>
```

Define the data streaming processes and its associated remote trail.

```
GGSCI (c<n>ds2.oracle.com) 2> add extract PDBN202, exttrailsource ./dirdat/s2
EXTRACT added.

GGSCI (c<n>ds2.oracle.com) 4> add rmttrail ./dirdat/t2, extract PDBN202,
megabytes 100
RMTTRAIL added.


GGSCI (c<n>ds2.oracle.com) 5>
```

```
GGSCI (c<n>ds2.oracle.com) 6> start PDBN202

Sending START request to MANAGER ...
EXTRACT PDBN202 starting


GGSCI (c<n>ds2.oracle.com) 7> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN201     00:00:07      00:00:09
EXTRACT     RUNNING     PDBN201     00:00:00      01:56:33
EXTRACT     RUNNING     PDBN202     00:00:00      00:07:16


GGSCI (c<n>ds2.oracle.com) 8>
```

Task 4: Perform Data Streaming Tests

The tests is to perform data streaming across the cluster by executing:

Test 1: Run workload on primary node and verify tails is incrementing on extended node 1
Test 2: Run workload on extended node 1 and verify tails is incrementing on primary node
Test 3: Run workload on extended node 2 and verify tails is incrementing on primary node

Test 1: Run workload on the primary node and verify tails is incrementing on extended node 1

Run the workload simulator from the primary node, and verify the trail is accumulating at extended node 1.

On the primary node

```
[oracle@edbcn0 gghome_1]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Mon Dec 26 05:27:36 2016

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

SQL> conn osm_repapi/oracle@dbn0.oracle.com
Connected.
SQL> exec call_trans_load;
```

Verify remote trail is created on the primary node

On extended node 1

```
[oracle@edbcn1 ~]$ cd $OGG_HOME
[oracle@edbcn1 gghome_1]$ cd dirdat
[oracle@edbcn1 dirdat]$ ls -ltr
total 57472
-rw-r----- 1 oracle oinstall 58762037 Dec 26 05:21 t2000000
-rw-r----- 1 oracle oinstall    22337 Dec 26 05:28 tp000000
[oracle@edbcn1 dirdat]$ ls -ltr
total 57476
-rw-r----- 1 oracle oinstall 58762037 Dec 26 05:21 t2000000
-rw-r----- 1 oracle oinstall    26433 Dec 26 05:28 tp000000
[oracle@edbcn1 dirdat]$ ls -ltr
total 57488
-rw-r----- 1 oracle oinstall 58762037 Dec 26 05:21 t2000000
-rw-r----- 1 oracle oinstall    37681 Dec 26 05:28 tp000000
[oracle@edbcn1 dirdat]$
```

Terminate primary node SQL*Plus session now.

Test 2: Run workload on extended node 1 and verify tails is incrementing on the primary node

Run the workload simulator from extended node 1, and verify the trail is accumulating at the primary.

On extended node 1

```
[oracle@edbcn1 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Mon Dec 26 05:33:55 2016

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

SQL> conn osm_repapi/oracle@dbn1.oracle.com
Connected.
SQL> exec call_trans_load;
```

On the primary node

```
[oracle@edbcn0 gghome_1]$ cd dirdat
[oracle@edbcn0 dirdat]$ ls -ltr
total 57480
-rw-r----- 1 oracle oinstall 58762037 Dec 26 04:47 t2000000
-rw-r----- 1 oracle oinstall    32608 Dec 26 05:34 t1000000
[oracle@edbcn0 dirdat]$ ls -ltr
total 57480
-rw-r----- 1 oracle oinstall 58762037 Dec 26 04:47 t2000000
-rw-r----- 1 oracle oinstall    32608 Dec 26 05:34 t1000000
[oracle@edbcn0 dirdat]$ ls -ltr
total 57480
-rw-r----- 1 oracle oinstall 58762037 Dec 26 04:47 t2000000
-rw-r----- 1 oracle oinstall    32608 Dec 26 05:34 t1000000
[oracle@edbcn0 dirdat]$ ls -ltr
total 57488
-rw-r----- 1 oracle oinstall 58762037 Dec 26 04:47 t2000000
-rw-r----- 1 oracle oinstall    37726 Dec 26 05:35 t1000000
[oracle@edbcn0 dirdat]$ ls -ltr
total 57492
-rw-r----- 1 oracle oinstall 58762037 Dec 26 04:47 t2000000
-rw-r----- 1 oracle oinstall    41822 Dec 26 05:35 t1000000
[oracle@edbcn0 dirdat]$
```

Terminate extended node 1 SQL*Plus session now.

Test 3: Run workload on extended node 2 and verify tails is incrementing on the primary node

Run the workload simulator from extended node 2, and verify the trail is accumulating at the primary.

```
[oracle@edbcn2 gghome_1]$ sqlplus /nolog

SQL*Plus: Release 11.2.0.4.0 Production on Mon Dec 26 05:36:08 2016

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> conn osm_repapi/oracle@dbn2.oracle.com
Connected.
SQL> exec call_trans_load
```

```
[oracle@edbcn0 gghome_1]$ cd dirdat
[oracle@edbcn0 dirdat]$ ls -ltr
total 57480
-rw-r----- 1 oracle oinstall 58762037 Dec 26 04:47 t2000000
-rw-r----- 1 oracle oinstall    32608 Dec 26 05:34 t1000000
[oracle@edbcn0 dirdat]$ ls -ltr
total 57480
-rw-r----- 1 oracle oinstall 58762037 Dec 26 04:47 t2000000
-rw-r----- 1 oracle oinstall    32608 Dec 26 05:34 t1000000
[oracle@edbcn0 dirdat]$ ls -ltr
total 57480
-rw-r----- 1 oracle oinstall 58762037 Dec 26 04:47 t2000000
-rw-r----- 1 oracle oinstall    32608 Dec 26 05:34 t1000000
[oracle@edbcn0 dirdat]$ ls -ltr
total 57488
-rw-r----- 1 oracle oinstall 58762037 Dec 26 04:47 t2000000
-rw-r----- 1 oracle oinstall    37726 Dec 26 05:35 t1000000
[oracle@edbcn0 dirdat]$ ls -ltr
total 57492
-rw-r----- 1 oracle oinstall 58762037 Dec 26 04:47 t2000000
-rw-r----- 1 oracle oinstall    41822 Dec 26 05:35 t1000000
[oracle@edbcn0 dirdat]$ ls -ltr
total 57592
-rw-r----- 1 oracle oinstall    93747 Dec 26 05:35 t1000000
-rw-r----- 1 oracle oinstall 58807182 Dec 26 05:37 t2000000
[oracle@edbcn0 dirdat]$ ls -ltr
total 57600
-rw-r----- 1 oracle oinstall    93747 Dec 26 05:35 t1000000
-rw-r----- 1 oracle oinstall 58817384 Dec 26 05:37 t2000000
[oracle@edbcn0 dirdat]$ ls -ltr
total 57600
-rw-r----- 1 oracle oinstall    93747 Dec 26 05:35 t1000000
-rw-r----- 1 oracle oinstall 58818407 Dec 26 05:37 t2000000
[oracle@edbcn0 dirdat]$
```

Terminate extended node2 SQL*Plus session now.

Workshop 6: Configuring EDBC Nodes Synchronization

This configure set of data synchronization processes - Replicats. Data is applied to the participated nodes on real-time using the target database SQL native calls.

Table 10 lists Workshop 6 tasks.

Task No	Task Description
1	Configure Synchronization of the Primary node
2	Configure Synchronization of extended node 1
3	Configure Synchronization of extended node 2
4	Review and verify Oracle GoldenGate 12c instances processes status
5	Perform basic Synchronization test

Table 10: Workshop 6 Tasks

Task 1: Configure Synchronization of the Primary node

Apply data streaming of the primary node to extended node 2. The Replicat process RDBN002 applies ./dirdat/tp to extended node 2.

On extended node 2

Create the checkpoint table for extended node 2 Replicats.

```
[oracle@edbcn2 gghome_1]$ vi GOLBALS

checkpointtable ogg_admin.chkpttab
```

```
GGSCI (edbcn2.oracle.com) 1>$./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 (edbcn2.oracle.com) 1> dblogin userid ogg_admin@dbn2.oracle.com, password
oracle
Successfully logged into database.

GGSCI (edbcn2.oracle.com) 2> add checkpointtable
```

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

Successfully created checkpoint table ogg_admin.chkpttab.

GGSCI (edbcn2.oracle.com) 3>
```

```
GGSCI (edbcn2.oracle.com) 1> edit params RDBN002
```

```
replicat RDBN002
SETENV (ORACLE_SID = DBN2)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/11.2.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn2.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn002.dsc, PURGE
MAP osm_repapi.*, TARGET osm_repapi.*;
```

```
GGSCI (edbcn2.oracle.com) 4> view params RDBN002


replicat RDBN002
SETENV (ORACLE_SID = DBN2)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/11.2.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn2.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn002.dsc, PURGE
MAP osm_repapi.*, TARGET osm_repapi.*;



GGSCI (edbcn2.oracle.com) 4>
```

Define the Replicat

```
GGSCI (edbcn2.oracle.com) 4> add replicat RDBN002, exttrail ./dirdat/tp
REPLICAT added.


GGSCI (edbcn2.oracle.com) 5>
```

On extended node 2

```
GGSCI (edbcn2.oracle.com) 6> start *

Sending START request to MANAGER ...
REPLICAT RDBN002 starting


GGSCI (edbcn2.oracle.com) 8> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt
```

```
MANAGER     RUNNING
REPLICAT    RUNNING     RDBN002     00:00:00      00:03:58


GGSCI (edbcn2.oracle.com) 9>
```

Task 2: Configure Synchronization for Extended Node 1

Configure apply on extended node 2. It reads trails received from downstream C1DS1B.

On extended node 2

```
GGSCI (edbcn2.oracle.com) 1> edit params rdbn102
```

```
replicat RDBN102
SETENV (ORACLE_SID = DBN2)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/11.2.0/dbhome 1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn2.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn102.dsc, PURGE
MAP osm_repapi.*, TARGET osm_repapi.*;
```

```
GGSCI (edbcn2.oracle.com) 2> view params rdbn102

replicat RDBN102
SETENV (ORACLE_SID = DBN2)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS LANG = AMERICAN AMERICA.AL32UTF8)
USERID ogg_admin@dbn2.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn102.dsc, PURGE
MAP osm_repapi.*, TARGET osm_repapi.*;



GGSCI (edbcn2.oracle.com) 3>
```

Define the Replicat

```
GGSCI (edbcn2.oracle.com) 3> add replicat rdbn102, exttrail ./dirdat/t1
REPLICAT added.


GGSCI (edbcn2.oracle.com) 4>
```

On the extended node 2

```
GGSCI (edbcn2.oracle.com) 5> start RDBN102

Sending START request to MANAGER ...
```

```
REPLICAT RDBN102 starting


GGSCI (edbcn2.oracle.com) 6> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN002     00:00:00      00:00:08
REPLICAT    RUNNING     RDBN102     01:13:16      00:00:03


GGSCI (edbcn2.oracle.com) 7>
```

Task 3: Configure Synchronization for Extended Node 2

Configure apply on extended node 1.

On extended node 2

```
GGSCI (edbcn1.oracle.com) 1> edit params rdbn202
```

```
replicat RDBN202
SETENV (ORACLE_SID = DBN1)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn1.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn202.dsc, PURGE
MAP osm_repapi.*, TARGET osm_repapi.*;
```

```
GGSCI (edbcn1.oracle.com) 2> view params rdbn202

replicat RDBN202
SETENV (ORACLE_SID = DBN1)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn1.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn202.dsc, PURGE
MAP osm_repapi.*, TARGET osm_repapi.*;



GGSCI (edbcn1.oracle.com) 3>
```

Define the Replicat

```
GGSCI (edbcn1.oracle.com) 3> add replicat RDBN202, exttrail ./dirdat/t2
REPLICAT added.
```

```
GGSCI (edbcn1.oracle.com) 4>
```

```
GGSCI (edbcn1.oracle.com) 27> start RDBN202

Sending START request to MANAGER ...
REPLICAT RDBN202 starting


GGSCI (edbcn1.oracle.com) 28> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN001     00:00:00      00:00:07
REPLICAT    RUNNING     RDBN202     00:00:00      00:00:04


GGSCI (edbcn1.oracle.com) 29>
```

Task 4: Review and Verify EDBC Processes Status

The following are the list of EDBC processes running on:

- The primary node
- Extended node 1
- Extended node 2
- Downstream system C<n>DS1A
- Downstream system C<n>DS1B
- Downstream system C<n>DS2A

Primary node

```
[oracle@edbcn0 gghome_1]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (edbcn0.oracle.com) 1> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN101     00:00:00      00:00:04
REPLICAT    RUNNING     RDBN201     00:00:00      00:00:07
```

```
GGSCI (edbcn0.oracle.com) 2>
```

Process Name	Process Description	Running Status
Manager	Manager	☐
EDBN101	Integrated Extract for extended 1 node	☐
PDBN101	Data Pump for extended node 1	☐

Extended node 1

```
GGSCI (edbcn1.oracle.com) 29> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN001     00:00:00      00:00:06
REPLICAT    RUNNING     RDBN202     00:00:00      00:00:02


GGSCI (edbcn1.oracle.com) 30>
```

Process Name	Process Description	Running Status
Manager	Manager	☐
RDBN001	Replicat for primary node	☐
RDBN202	Replicat for extended node 2	☐

Extended node 2

```
GGSCI (edbcn2.oracle.com) 11> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN002     00:00:00      00:00:09
REPLICAT    RUNNING     RDBN102     00:00:00      00:00:08


GGSCI (edbcn2.oracle.com) 12>
```

Process Name	Process Description	Running Status
Manager	Manager	☐
RDBN002	Replicat for primary node	☐
RDBN102	Replicat for extended node 1	☐

Downstream System 1A (C<n>DS1A)

```
[oracle@c1ds1 ~]$ cd $OGG_HOME1
[oracle@c1ds1 gghome_1]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (c1ds1.oracle.com) 1> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN001     00:00:23      00:00:08
EXTRACT     RUNNING     PDBN001     00:00:00      00:00:04
EXTRACT     RUNNING     PDBN002     00:00:00      00:00:01


GGSCI (c1ds1.oracle.com) 2>
```

Process Name	Process Description	Running Status
Manager	Manager	☐
EDBN001	Integrated Extract for the primary node	☐
PDBN001	Data Pump for the primary node	☐
PDBN002	Data Pump for extended node 2	☐

Downstream System 1B (C<n>DS1B)

```
GGSCI (c1ds1.oracle.com) 2> exit
[oracle@c1ds1 gghome_1]$ . oraenv
ORACLE_SID = [C1DS1A] ? C1DS1B
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@c1ds1 gghome_1]$ cd $OGG_HOME2
[oracle@c1ds1 gghome_2]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.


GGSCI (c1ds1.oracle.com) 1> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt
```

```
MANAGER     RUNNING
EXTRACT     RUNNING     EDBN101     00:00:22      00:00:09
EXTRACT     RUNNING     PDBN101     00:00:00      00:00:09
EXTRACT     RUNNING     PDBN102     00:00:00      00:00:05

GGSCI (c1ds1.oracle.com) 2>
```

Process Name	Process Description	Running Status
Manager	Manager	☐
EDBN101	Integrated Extract for extended 1 node	☐
PDBN101	Data Pump for extended node 1	☐
PDBN102	Data Pump for extended node 2	☐

Downstream System 2A (C<n>DS2A)

```
[oracle@c1ds2 gghome_1]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.


GGSCI (c1ds2.oracle.com) 1> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN201     00:00:28      00:00:07
EXTRACT     RUNNING     PDBN201     00:00:00      00:00:05
EXTRACT     RUNNING     PDBN202     00:00:00      00:00:05


GGSCI (c1ds2.oracle.com) 2>
```

Process Name	Process Description	Running Status
Manager	Manager	☐
EDBN201	Integrated Extract for extended 2 node	☐
PDBN201	Data Pump for extended node 2	☐
PDBN202	Data Pump for extended node 1	☐

All EDBC process must be on RUNNING status, otherwise troubleshooting is necessary.

Close all PuTTY terminals for:

edbcn0.oracle.com, edbcn1.oracle.com, edbcn2.oracle.com

c<n>ds1.oracle.com. c<n>ds2.oracle.com

Workshop 7: EDBC Workload Simulation

This workshop empowers EDBC PL/SQL API workload simulator. The workload simulator delivers workload for DML statements INSERT, UPDATE and DELETE. This workshop demonstrated using INSERT workload only, UPDATE and DELETE statements are demonstrated on Workshop 8.

Table 11 lists Workshop 7 tasks.

Task No	Task Description
1	Run workload on the primary node
2	Run workload on the extended node 1
3	Run workload on the extended node 2
4	Run workload on the cluster

Table 11: Workshop 7 Tasks

The workload simulator has default values suitable for basic workload sessions as described next.

```
SQL> desc trans_load
PROCEDURE trans_load
 Argument Name                  Type                    In/Out Default?
 ------------------------------ ----------------------- ------ --------
 PLOAD_LEVEL                    NUMBER                  IN     DEFAULT
 PSTARTFROM                     NUMBER                  IN     DEFAULT
 PDB_NAME                       VARCHAR2                IN
 PDML                           VARCHAR2                IN
 PBUCKET                        NUMBER                  IN
 PCDR                           VARCHAR2                IN

SQL> desc call trans load
PROCEDURE call_trans_load
 Argument Name                  Type                    In/Out Default?
 ------------------------------ ----------------------- ------ --------
 N                              NUMBER                  IN     DEFAULT
 S                              NUMBER                  IN     DEFAULT
 PDB_NAME                       VARCHAR2                IN     DEFAULT
 PDML                           VARCHAR2                IN     DEFAULT
 PBUCKET                        NUMBER                  IN     DEFAULT
 PCDR                           VARCHAR2                IN     DEFAULT

SQL>
```

When overriding the default values for the call_trans_load PL/SQL procedure, consider the system resource available, this workshop developed for Oracle VirtualBox Linux server running on laptop, use the default values as increasing the workload may severely slowdown the processing resources.

The following is the PL/SQL APIs headers and description.

```
trans_load (pLoad_Level in number default 10,
            pStartFrom  in number default 1000000,
            pDB NAME    in varchar2,
            pDML        in varchar2,
            pBucket     in number,
            pCDR in varchar2)
```

Parameter Name	Parameter Description
pLoad_Level	Multiplier of the workload runtime, default is 10.
pStartFrom	For initial run, it's default to 1000000, otherwise it start from last value
pDB_Name	Target database name
pDML	Type of run-time DML statement
pBucket	Row set size
pCDR	Is it from Conflict Detection and Resolution Workload

```
call_trans_load(n        in number   default 500,
                s        in number   default 5,
                pDB NAME in varchar2 default null,
                pDML     in varchar2 default 'I',
                pBucket  in number   default 50,
                pCDR     in varchar2 default 'N')
```

Parameter Name	Parameter Description
n	Number of calls to TRANS_LOAD, default is 500 calls
s	Elapsed time between transactions, default is 5 seconds
pDB_Name	Determines the target database name, default is null
pDML	Determines the type of run-time DML statement, default (I)nsert
pBucket	Pass the row set size, default is 50 rows
pCDR	Is it from Conflict Detection and Resolution Workload, default none

Task 1: Run Workload on the Primary Node

The workload performs insert DML on the primary, the data streaming to extended node 1 and extended node 2 are implicitly executed.

On the primary node

```
SQL> conn osm_repapi/oracle@dbn0.oracle.com
Connected.
SQL>
```

```
SQL> exec call_trans_load;
```

On extended node 1

```
SQL> conn osm_repapi/oracle@dbn1.oracle.com
Connected.
SQL> select count(*) from customers;

  COUNT(*)
----------
    123323

SQL> /

  COUNT(*)
----------
    123333

SQL>
```

On extended node 2

```
[oracle@edbcn2 gghome_1]$ sqlplus /nolog

SQL*Plus: Release 11.2.0.4.0 Production on Mon Dec 26 13:03:30 2016

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> conn osm_repapi/oracle@dbn2.oracle.com
Connected.
SQL> select count(*) from customers;

  COUNT(*)
----------
    125093

SQL> /

  COUNT(*)
----------
    125103

SQL>
```

Task 2: Run Workload on Extended Node 1

```
SQL> conn osm_repapi/oracle@dbn1.oracle.com
Connected.
SQL>
```

```
SQL> exec call_trans_load;
```

On the primary node

```
SQL> conn osm_repapi/oracle@dbn0.oracle.com
Connected.
SQL> select count(*) from customers;

  COUNT(*)
----------
    123963

SQL> /

  COUNT(*)
----------
    123973

SQL>
```

On extended node 2

```
SQL> conn osm_repapi/oracle@dbn2.oracle.com
Connected.
SQL> select count(*) from customers;

  COUNT(*)
----------
    125473

SQL> /

  COUNT(*)
----------
    125494

SQL>
```

Task 3: Run Workload on Extended Node 2

```
SQL> conn osm_repapi/oracle@dbn2.oracle.com
Connected.
SQL>
```

```
SQL> exec call_trans_load;
```

On the primary node

```
SQL> conn osm_repapi/oracle@dbn0.oracle.com
Connected.
SQL> select count(*) from customers;

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

```
    124973

SQL> /


  COUNT(*)
----------
    124983

SQL>
```

On extended node 1

```
SQL> conn osm_repapi/oracle@dbn1.oracle.com
Connected.
SQL> select count(*) from customers;

  COUNT(*)
----------
    125223

SQL> /

  COUNT(*)
----------
    125282

SQL>
```

Task 4: Run Workload Across EDBC

Run a simultaneous workload sessions.

Connect and run the workload on the primary node

On the primary node

```
[oracle@edbcn0 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Tue Dec 27 02:50:21 2016

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

SQL> conn osm_repapi/oracle@dbn0.oracle.com
Connected.
SQL> exec call_trans_load;
```

Connect and run the workload on extended node 1

On extended node 1

```
[oracle@edbcn1 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Tue Dec 27 02:51:35 2016

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

SQL> conn osm_repapi/oracle@dbn1.oracle.com
Connected.
SQL> exec call_trans_load;
```

Connect and run the workload on extended node 2

On extended node 2

```
[oracle@edbcn2 ~]$ sqlplus /nolog

SQL*Plus: Release 11.2.0.4.0 Production on Tue Dec 27 02:52:28 2016

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> conn osm_repapi/oracle@dbn2.oracle.com
Connected.
SQL> exec call_trans_load;
```

Query the workload on the primary node, extended node 1 and extended node 2. The session uses database links to access data from extended node 1 and extended node 2.

On the primary node

```
SQL> select count(*) from customers@dbn1.oracle.com;

  COUNT(*)
----------
    127382

SQL> select count(*) from customers@dbn1.oracle.com;

  COUNT(*)
----------
    127412

SQL> select count(*) from customers@dbn2.oracle.com;

  COUNT(*)
----------
    129162

SQL> select count(*) from customers@dbn2.oracle.com;

  COUNT(*)
----------
    129253

SQL>
```

Query the database DBN0, DBN1 and DBN2 using the next query. It's originated at DBN0 and query DBN1 and DBN2 through database links.

On the primary node

```
SQL> select * from (select 'DBN0' DB, count(*) ROWS# from customers) union
  2  select * from (select 'DBN1' DBN1, count(*) ROWS# from
customers@dbn1.oracle.com) union
  3  select * from (select 'DBN2' DBN2, count(*) ROWS# from
customers@dbn2.oracle.com)
  4  /

DB     ROWS#
---- ----------
DBN0  133002
DBN1  132212
DBN2  133172

SQL> /

DB     ROWS#
---- ----------
DBN0  133032
DBN1  132234
DBN2  133197

SQL>
```

Workshop 8: EDBC Conflict Detection and Resolution

An Introduction

A major enablement is the automatic data conflict detection and resolution (CDR) across the application and Oracle GoldenGate. Understanding the levels of handling CDR is crucial and must be part of the application development process. The application must show an interoperability with Oracle GoldenGate change data capture happening at the data level. Figure 11 shows Oracle best practice for handling data conflict detection and resolution.

For further details, refer to Appendix B.

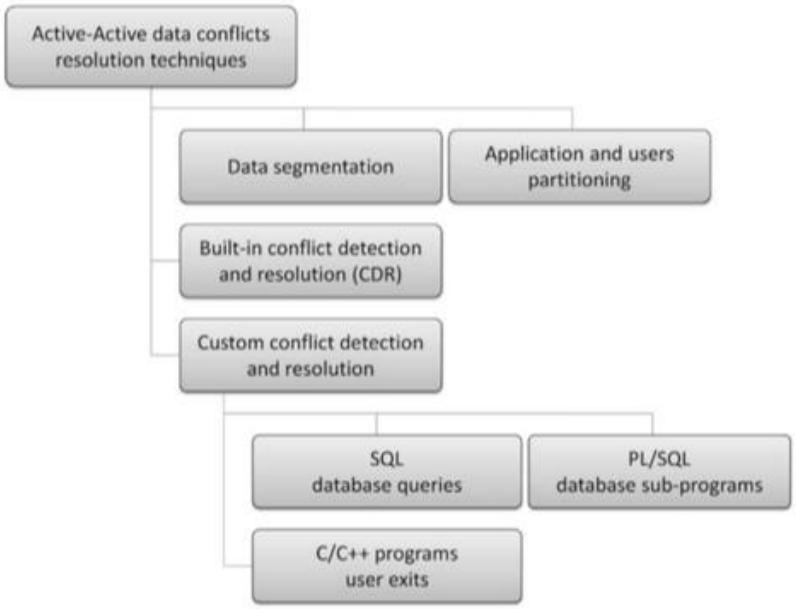


Figure 11: Conflict Detection and Resolution (CDR) Hierarchy

Table 12 lists Workshop8 tasks to deploy Oracle GoldenGate 12c Conflict Detection and Resolution (CDR).

Task No	Description
1	Perform Conflict Detection and Resolution prerequisites
2	Set up built-in Conflict Detection and Resolution
3	Testing INSERT Conflict Detection and Resolution

4	Testing UPDATE Conflict Detection and Resolution
5	Testing DELETE Conflict Detection and Resolution
6	Deploy Virtual Private Database

Table 12: EDBC Conflict Detection and Resolution

There will be several tests conducted to verify data conflict detection and resolution as an integrated feature of EDBC. The following test will be performed:

- Unique constraint violation
- Update row exist
- Update row does not exist
- Delete row miss-matched
- Delete row does not exist

Task 1: Perform Conflict Detection and Resolution Prerequisites

The tables at DBN0, DBN1 and DBN2 defined as shown below. CDR is protected using the data segregation column (BAS_X) and the timestamp column (DML_TS).

```
SQL> 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_ADDRESS                                        VARCHAR2(128)
 BAS_X                                     NOT NULL VARCHAR2(32)
 DML_TS                                             TIMESTAMP(6)

SQL>
```

```
SQL> desc policies
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 POL_NO                                    NOT NULL NUMBER
 POL_FROM                                  NOT NULL DATE
 POL_TO                                    NOT NULL DATE
 POL_VALUE                                 NOT NULL NUMBER(8,2)
 POL_SUB_TOTAL                             NOT NULL NUMBER(8,2)
 PT_CODE                                   NOT NULL VARCHAR2(8)
 DT_CODE                                            VARCHAR2(8)
 POL_TOTAL                                 NOT NULL NUMBER(8,2)
 CUST_NO                                   NOT NULL NUMBER
 BAS_X                                     NOT NULL VARCHAR2(32)
 DML_TS                                             TIMESTAMP(6)

SQL>
```

In order to capture the before image of a row, the table-level supplemental logging requires explicit column names. Drop-recreate the table-level supplemental logging using the DELETE TRANDATA and ADD TRANDATA with the COLS option to list the columns names.

Primary node

Login to the database

```
GGSCI (edbcn0.oracle.com) 2> miningdblogin userid ogg_admin@c1ds1a.oracle.com,
password oracle
Successfully logged into mining database.

GGSCI (edbcn0.oracle.com) 3> dblogin userid ogg_admin@dbn0.oracle.com, password
oracle
Successfully logged into database.

GGSCI (edbcn0.oracle.com) 4>
```

Delete existing table-level supplemental logging

```
GGSCI (edbcn0.oracle.com) 7> delete trandata OSM_REPAPI.POLICIES

Logging of supplemental redo log data disabled for table OSM REPAPI.POLICIES.
TRANDATA for scheduling columns has been disabled on table
'OSM REPAPI.POLICIES'.
GGSCI (edbcn0.oracle.com) 8> delete trandata OSM_REPAPI.customers

Logging of supplemental redo log data disabled for table OSM REPAPI.CUSTOMERS.
TRANDATA for scheduling columns has been disabled on table
'OSM_REPAPI.CUSTOMERS'.
GGSCI (edbcn0.oracle.com) 9>
```

Add supplemental logging columns and verify

```
GGSCI (c1ds1.oracle.com) 44> add trandata osm repapi.customers, cols(CUST NAME,
CUST_EMAIL, CUST_MOBILE, CUST_ADDRESS, DML_TS)

Logging of supplemental redo data enabled for table OSM_REPAPI.CUSTOMERS.
TRANDATA for scheduling columns has been added on table 'OSM_REPAPI.CUSTOMERS'.
GGSCI (c1ds1.oracle.com) 45> info trandata osm_repapi.customers

Logging of supplemental redo log data is enabled for table OSM_REPAPI.CUSTOMERS.

Columns supplementally logged for table OSM_REPAPI.CUSTOMERS: ALL.

GGSCI (c1ds1.oracle.com) 46> add trandata osm_repapi.policies, cols(POL_FROM,
POL TO, POL VALUE, POL SUB TOTAL, PT CODE, DT CODE, POL TOTAL, CUST NO, DML TS)

Logging of supplemental redo data enabled for table OSM_REPAPI.POLICIES.
TRANDATA for scheduling columns has been added on table 'OSM_REPAPI.POLICIES'.
```

```
GGSCI (c1ds1.oracle.com) 47> info trandata osm_repapi.policies

Logging of supplemental redo log data is enabled for table OSM REPAPI.POLICIES.

Columns supplementally logged for table OSM_REPAPI.POLICIES: BAS_X, POL_NO,
POL_FROM, POL_SUB_TOTAL, PT_CODE, DT_CODE, CUST_NO, POL_TO, POL_VALUE,
POL_TOTAL, DML_TS.

GGSCI (c1ds1.oracle.com) 48>
```

Extended node 1

Login to the database

```
GGSCI (c1ds1.oracle.com) 2> miningdblogin userid ogg_admin@c1ds1b.oracle.com,
password oracle
Successfully logged into mining database.

GGSCI (c1ds1.oracle.com) 4> dblogin userid ogg_admin@dbn1.oracle.com, password
oracle
Successfully logged into database.

GGSCI (c1ds1.oracle.com) 5>
```

Delete existing table-level supplemental logging

```
 GGSCI (c1ds1.oracle.com) 5> delete trandata OSM_REPAPI.POLICIES

Logging of supplemental redo log data disabled for table OSM_REPAPI.POLICIES.
TRANDATA for scheduling columns has been disabled on table
'OSM_REPAPI.POLICIES'.
GGSCI (c1ds1.oracle.com) 6> delete trandata OSM_REPAPI.customers

Logging of supplemental redo log data disabled for table OSM_REPAPI.CUSTOMERS.
TRANDATA for scheduling columns has been disabled on table
'OSM_REPAPI.CUSTOMERS'.
GGSCI (c1ds1.oracle.com) 7>
```

Add supplemental logging columns and verify

```
GGSCI (c1ds1.oracle.com) 18> add trandata osm_repapi.customers, cols(CUST_NAME,
CUST EMAIL, CUST MOBILE, CUST ADDRESS, DML TS)

Logging of supplemental redo data enabled for table OSM_REPAPI.CUSTOMERS.
TRANDATA for scheduling columns has been added on table 'OSM_REPAPI.CUSTOMERS'.
GGSCI (c1ds1.oracle.com) 19> info trandata osm repapi.customers

Logging of supplemental redo log data is enabled for table OSM REPAPI.CUSTOMERS.

Columns supplementally logged for table OSM_REPAPI.CUSTOMERS: ALL.

GGSCI (c1ds1.oracle.com) 20> add trandata osm repapi.policies, cols(POL FROM,
POL_TO, POL_VALUE, POL_SUB_TOTAL, PT_CODE, DT_CODE, POL_TOTAL, CUST_NO, DML_TS)
```

```
Logging of supplemental redo data enabled for table OSM_REPAPI.POLICIES.
TRANDATA for scheduling columns has been added on table 'OSM_REPAPI.POLICIES'.
GGSCI (c1ds1.oracle.com) 21> info trandata osm repapi.policies

Logging of supplemental redo log data is enabled for table OSM_REPAPI.POLICIES.

Columns supplementally logged for table OSM_REPAPI.POLICIES: ALL.

GGSCI (c1ds1.oracle.com) 22>
```

Extended node 2

Login to the database

```
GGSCI (c1ds2.oracle.com) 2> miningdblogin userid ogg_admin@c1ds2a.oracle.com,
password oracle
Successfully logged into mining database.

GGSCI (c1ds2.oracle.com) 4> dblogin userid ogg_admin@dbn2.oracle.com, password
oracle
Successfully logged into database.

GGSCI (c1ds2.oracle.com) 5>
```

Delete existing table-level supplemental logging

```
GGSCI (c1ds2.oracle.com) 5> delete trandata OSM_REPAPI.POLICIES

Logging of supplemental redo log data disabled for table OSM_REPAPI.POLICIES.
TRANDATA for scheduling columns has been disabled on table
'OSM_REPAPI.POLICIES'.
GGSCI (c1ds2.oracle.com) 6> delete trandata OSM_REPAPI.customers

Logging of supplemental redo log data disabled for table OSM_REPAPI.CUSTOMERS.
TRANDATA for scheduling columns has been disabled on table
'OSM_REPAPI.CUSTOMERS'.
GGSCI (c1ds2.oracle.com) 7>
```

Add supplemental logging columns and verify

```
GGSCI (c1ds2.oracle.com) 16> add trandata osm_repapi.customers, cols(CUST_NAME,
CUST EMAIL, CUST MOBILE, CUST ADDRESS, DML TS)

Logging of supplemental redo data enabled for table OSM_REPAPI.CUSTOMERS.
TRANDATA for scheduling columns has been added on table 'OSM_REPAPI.CUSTOMERS'.
GGSCI (c1ds2.oracle.com) 17> info trandata osm repapi.customers

Logging of supplemental redo log data is enabled for table OSM REPAPI.CUSTOMERS.

Columns supplementally logged for table OSM_REPAPI.CUSTOMERS: ALL.

GGSCI (c1ds2.oracle.com) 18> add trandata osm repapi.policies, cols(POL FROM,
POL_TO, POL_VALUE, POL_SUB_TOTAL, PT_CODE, DT_CODE, POL_TOTAL, CUST_NO, DML_TS)
```

```
Logging of supplemental redo data enabled for table OSM_REPAPI.POLICIES.
TRANDATA for scheduling columns has been added on table 'OSM_REPAPI.POLICIES'.
GGSCI (c1ds2.oracle.com) 19> info trandata osm repapi.policies

Logging of supplemental redo log data is enabled for table OSM_REPAPI.POLICIES.

Columns supplementally logged for table OSM_REPAPI.POLICIES: ALL.

GGSCI (c1ds2.oracle.com) 20>
```

Task 2: Setup for Built-in Conflict Detection and Resolution

There will be two conflict Detection and Resolution (CDR) configuration. For EDBC with compute capacity of 3-node, there will be several CDR enabled as follows:

- CDR of primary node (IE), extended node 1 (R) and extended node 2 (DSR)
- CDR of extended node 1 (IE), primary node (R) and extended node 2 (DSR)
- CDR of extended node 2 (IE), Primary node (R) and extended node 1 (DSR)

Legends:

IE : Integrated Extract
R: Replicat
DSR: Data Streaming Replicat

CDR is implemented using parameters setting for the integrated extract and Replicat. Table 11 lists the parameters for CDR.

CDR Parameter	Parameter Desertion
EXCLUDEUSER	Apply this parameter to the extract process of the primary and extended nodes.
LOGALLSUPCOLS	Writes all supplemental logged columns to the trail, including those required for conflict detection and resolution and the scheduling columns required to support integrated Replicat.
GETUPDATEBEFORES	GETUPDATEBEFORES is required when using the Conflict Detection and Resolution (CDR) feature for a Oracle database on any of the platforms that are supported by Oracle GoldenGate.
GETBEFORECOLS	Use GETBEFORECOLS to specify columns for which you want before image to be captured and written to the trail upon an update or delete

	operation. Use GETBEFORECOLS when using the Oracle GoldenGate Conflict Detection and Resolution (CDR) feature in a bi-directional or multi-master configuration.
IGNOREGETUPDATEBEFORES	Disable GETUPDATEBEFORES.

Table 11: Active-Active Basic CDR parameters

On the Integrated Extract, the following Conflict Detection and Resolution parameters added.

```
TRANLOGOPTIONS EXCLUDEUSER ogg_admin
LOGALLSUPCOLS
-- CDR Parameters
GETUPDATEBEFORES
TABLE osm_repapi.customers, &
GETBEFORECOLS(ON UPDATE KEYINCLUDING (CUST_MOBILE), ON DELETE ALL);
IGNOREUPDATEBEFORES
GETUPDATEBEFORES
TABLE osm_repapi.policies, &
GETBEFORECOLS(ON UPDATE ALL, ON DELETE ALL);
IGNOREUPDATEBEFORES
```

On the Replicat, the following Conflict Detection and Resolution parameters added.

```
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD))
```

CDR of primary node and extended node 1

Modify the primary node Integrated Extract (EDBN001) to enable CDR for only the transactional type of tables - CUSTOMERS and POLICIES tables.

```
EXTRACT EDBN001
SETENV (ORACLE_SID = C1DS1A)
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@DBN0.oracle.com, PASSWORD oracle
tranlogoptions mininguser ogg_admin@C1DS1A.oracle.com, miningpassword oracle
tranlogoptions integratedparams (max_sga_size 128, downstream_real_time_mine Y)
EXTTRAIL ./dirdat/sp
--TABLE osm_repapi.*;
TRANLOGOPTIONS EXCLUDEUSER ogg admin
-- LOGALLSUPCOLS
-- CDR Parameters
GETUPDATEBEFORES
TABLE osm_repapi.customers, &
GETBEFORECOLS(ON UPDATE KEYINCLUDING (CUST_MOBILE), ON DELETE ALL);
```

```
IGNOREUPDATEBEFORES
GETUPDATEBEFORES
TABLE osm repapi.policies, &
GETBEFORECOLS(ON UPDATE ALL, ON DELETE ALL);
IGNOREUPDATEBEFORES
```

Modify the extended node 1 Replicat (RDBN001) to handle CDR

```
replicat RDBN001
SETENV (ORACLE_SID = DBN1)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn1.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn001.dsc, PURGE, megabytes 2000
--MAP osm_repapi.*, TARGET osm_repapi.*;
MAP osm_repapi.customers, TARGET osm_repapi.customers, &
-- Implement Conflict Detection and Resolution - CDR
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
--
MAP osm_repapi.policies, TARGET osm_repapi.policies, &
-- Implement Conflict Detection and Resolution - CDR
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
```

Modify the extended node 2 Replicat (RDBN002) to handle CDR

```
replicat RDBN002
SETENV (ORACLE_SID = DBN2)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/11.2.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn2.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn002.dsc, PURGE, megabytes 2000
-- MAP osm_repapi.*, TARGET osm_repapi.*;
MAP osm_repapi.customers, TARGET osm_repapi.customers, &
-- Implement Conflict Detection and Resolution - CDR
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
--
MAP osm_repapi.policies, TARGET osm_repapi.policies, &
-- Implement Conflict Detection and Resolution - CDR
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
```

```
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
```

Stop, restart and verify the primary node, extended node 1 and extended node 2 Oracle GoldenGate 12c instances.

```
GGSCI (c1ds1.oracle.com) 21> stop *

Sending STOP request to EXTRACT EDBN101 ...
Request processed.

Sending STOP request to EXTRACT PDBN101 ...
Request processed.

Sending STOP request to EXTRACT PDBN102 ...
Request processed.


GGSCI (c1ds1.oracle.com) 22> start *

Sending START request to MANAGER ...
EXTRACT EDBN101 starting

Sending START request to MANAGER ...
EXTRACT PDBN101 starting

Sending START request to MANAGER ...
EXTRACT PDBN102 starting


GGSCI (c1ds1.oracle.com) 23> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN101     00:00:21      00:00:00
EXTRACT     RUNNING     PDBN101     00:00:00      00:00:09
EXTRACT     RUNNING     PDBN102     00:00:00      00:00:08


GGSCI (c1ds1.oracle.com) 24>
```

```
GGSCI (edbcn1.oracle.com) 8> stop *

Sending STOP request to REPLICAT RDBN001 ...
Request processed.

Sending STOP request to REPLICAT RDBN202 ...
Request processed.


GGSCI (edbcn1.oracle.com) 9> start *

Sending START request to MANAGER ...
REPLICAT RDBN001 starting
```

```
Sending START request to MANAGER ...
REPLICAT RDBN202 starting


GGSCI (edbcn1.oracle.com) 10> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN001     00:00:00      00:00:07
REPLICAT    RUNNING     RDBN202     00:00:00      00:00:06


GGSCI (edbcn1.oracle.com) 11>
```

```
GGSCI (edbcn2.oracle.com) 7> stop *

Sending STOP request to REPLICAT RDBN002 ...
Request processed.

Sending STOP request to REPLICAT RDBN102 ...
Request processed.


GGSCI (edbcn2.oracle.com) 8> start *

Sending START request to MANAGER ...
REPLICAT RDBN002 starting

Sending START request to MANAGER ...
REPLICAT RDBN102 starting


GGSCI (edbcn2.oracle.com) 9> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN002     00:00:00      00:00:05
REPLICAT    RUNNING     RDBN102     00:00:00      00:00:04


GGSCI (edbcn2.oracle.com) 10>
```

CDR of extended node 1 and the primary node

Modify extended node Integrated Extract (EDBN101) to enable CDR.

```
EXTRACT EDBN101
SETENV (ORACLE_SID = C1DS1B)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_2")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
SETENV (TNS_ADMIN = "/u01/app/oracle/product/12.1.0/dbhome_1/network/admin")
USERID ogg_admin@DBN1.oracle.com, PASSWORD oracle
tranlogoptions mininguser ogg_admin@C1DS1B.oracle.com, miningpassword oracle
```

```
tranlogoptions integratedparams (max_sga_size 128, downstream_real_time_mine Y)
EXTTRAIL ./dirdat/s1
--TABLE osm repapi.*;
TRANLOGOPTIONS EXCLUDEUSER ogg_admin
-- LOGALLSUPCOLS
-- CDR Parameters
GETUPDATEBEFORES
TABLE osm_repapi.customers, &
-- GETBEFORECOLS(ON UPDATE KEYINCLUDING (CUST NAME, CUST EMAIL, CUST MOBILE, CUS
T_ADDRESS, DML_TS), ON DELETE ALL);
GETBEFORECOLS(ON UPDATE KEYINCLUDING (CUST MOBILE), ON DELETE ALL);
IGNOREUPDATEBEFORES
GETUPDATEBEFORES
TABLE osm_repapi.policies, &
-- GETBEFORECOLS(ON UPDATE KEYINCLUDING (POL FROM, POL TO, POL VALUE, POL SUB TO
TAL, PT_CODE, DT_CODE, POL_TOTAL, CUST_NO, DML_TS), ON DELETE ALL);
GETBEFORECOLS(ON UPDATE ALL, ON DELETE ALL);
IGNOREUPDATEBEFORES
```

Modify the primary node Replicat (RDBN101) to handle CDR

```
replicat RDBN101
SETENV (ORACLE SID = DBN0)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS LANG = AMERICAN AMERICA.AL32UTF8)
USERID ogg_admin@dbn0.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn101.dsc, PURGE, megabytes 2000
-- MAP osm repapi.*, TARGET osm repapi.*;
MAP osm_repapi.customers, TARGET osm_repapi.customers, &
-- Implement Conflict Detection and Resolution - CDR
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
--
MAP osm_repapi.policies, TARGET osm_repapi.policies, &
-- Implement Conflict Detection and Resolution - CDR
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
```

Modify the extended node 2 Replicat (RDBN102) to handle CDR

```
replicat RDBN102
SETENV (ORACLE SID = DBN2)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/11.2.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn2.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn102.dsc, PURGE, megabytes 2000
-- MAP osm_repapi.*, TARGET osm_repapi.*;
```

```
MAP osm_repapi.customers, TARGET osm_repapi.customers, &
-- Implement Conflict Detection and Resolution - CDR
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
--
MAP osm_repapi.policies, TARGET osm_repapi.policies, &
-- Implement Conflict Detection and Resolution - CDR
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
```

Stop, restart and verify extended node 1, primary node and extended node 2 Oracle GoldenGate 12c instances.

```
GGSCI (c1ds1.oracle.com) 39> stop *

Sending STOP request to EXTRACT EDBN101 ...
Request processed.

Sending STOP request to EXTRACT PDBN101 ...
Request processed.

Sending STOP request to EXTRACT PDBN102 ...
Request processed.


GGSCI (c1ds1.oracle.com) 40> start *
EXTRACT EDBN101 is already running.

Sending START request to MANAGER ...
EXTRACT PDBN101 starting

Sending START request to MANAGER ...
EXTRACT PDBN102 starting


GGSCI (c1ds1.oracle.com) 41> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN101     00:00:56      00:00:06
EXTRACT     RUNNING     PDBN101     00:00:00      00:00:06
EXTRACT     RUNNING     PDBN102     00:00:00      00:00:06


GGSCI (c1ds1.oracle.com) 42>
```

```
GGSCI (edbcn0.oracle.com) 19> stop *
```

```
Sending STOP request to REPLICAT RDBN101 ...
Request processed.

Sending STOP request to REPLICAT RDBN201 ...
Request processed.


GGSCI (edbcn0.oracle.com) 20> start *

Sending START request to MANAGER ...
REPLICAT RDBN101 starting

Sending START request to MANAGER ...
REPLICAT RDBN201 starting


GGSCI (edbcn0.oracle.com) 21> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN101     00:00:00      00:00:07
REPLICAT    RUNNING     RDBN201     00:00:00      00:00:06


GGSCI (edbcn0.oracle.com) 22>
```

```
GGSCI (edbcn2.oracle.com) 14> stop *

Sending STOP request to REPLICAT RDBN002 ...
Request processed.

Sending STOP request to REPLICAT RDBN102 ...
Request processed.


GGSCI (edbcn2.oracle.com) 15> start *

Sending START request to MANAGER ...
REPLICAT RDBN002 starting

Sending START request to MANAGER ...
REPLICAT RDBN102 starting


GGSCI (edbcn2.oracle.com) 16> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN002     00:00:00      00:00:04
REPLICAT    RUNNING     RDBN102     00:00:00      00:00:03


GGSCI (edbcn2.oracle.com) 17>
```

CDR of extended node 2 and primary node

Modify extended node 2 Integrated Extract (EDBN201) to enable CDR.

```
EXTRACT EDBN201
SETENV (ORACLE_SID = C1DS2A)
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@DBN2.oracle.com, PASSWORD oracle
tranlogoptions mininguser ogg_admin@C1DS2A.oracle.com, miningpassword oracle
tranlogoptions integratedparams (max_sga_size 128, downstream_real_time_mine Y)
EXTTRAIL ./dirdat/s2
--TABLE osm_repapi.*;
TRANLOGOPTIONS EXCLUDEUSER ogg_admin
-- LOGALLSUPCOLS
-- CDR Parameters
GETUPDATEBEFORES
TABLE osm_repapi.customers, &
-- GETBEFORECOLS(ON UPDATE KEYINCLUDING (CUST_NAME, CUST_EMAIL, CUST_MOBILE, CUS
T_ADDRESS, DML_TS), ON DELETE ALL);
GETBEFORECOLS(ON UPDATE KEYINCLUDING (CUST_MOBILE), ON DELETE ALL);
IGNOREUPDATEBEFORES
GETUPDATEBEFORES
TABLE osm_repapi.policies, &
-- GETBEFORECOLS(ON UPDATE KEYINCLUDING (POL_FROM, POL_TO, POL_VALUE, POL_SUB_TO
TAL, PT_CODE, DT_CODE, POL_TOTAL, CUST_NO, DML_TS), ON DELETE ALL);
GETBEFORECOLS(ON UPDATE ALL, ON DELETE ALL);
IGNOREUPDATEBEFORES
```

Modify the primary node Replicat (RDBN201) to handle CDR

```
replicat RDBN201
SETENV (ORACLE_SID = DBN0)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn0.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn201.dsc, PURGE, megabytes 2000
-- MAP osm_repapi.*, TARGET osm_repapi.*;
MAP osm_repapi.customers, TARGET osm_repapi.customers, &
-- Implement Conflict Detection and Resolution - CDR
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
--
MAP osm_repapi.policies, TARGET osm_repapi.policies, &
-- Implement Conflict Detection and Resolution - CDR
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
```

```
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
```

Modify the extended node 1 Replicat (RDBN202) to handle CDR

```
replicat RDBN202
SETENV (ORACLE SID = DBN1)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS LANG = AMERICAN AMERICA.AL32UTF8)
USERID ogg_admin@dbn1.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn202.dsc, PURGE, megabytes 2000
--MAP osm repapi.*, TARGET osm repapi.*;
MAP osm repapi.customers, TARGET osm repapi.customers, &
-- Implement Conflict Detection and Resolution - CDR
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
--
MAP osm_repapi.policies, TARGET osm_repapi.policies, &
-- Implement Conflict Detection and Resolution - CDR
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
```

Stop, restart and verify extended node 2, primary node and extended node 1 Oracle GoldenGate 12c instances.

```
GGSCI (c1ds2.oracle.com) 9> stop *
EXTRACT EDBN201 is already stopped.

Sending STOP request to EXTRACT PDBN201 ...
Request processed.

Sending STOP request to EXTRACT PDBN202 ...
Request processed.

GGSCI (c1ds2.oracle.com) 10> start *

Sending START request to MANAGER ...
EXTRACT EDBN201 starting

Sending START request to MANAGER ...
EXTRACT PDBN201 starting

Sending START request to MANAGER ...
EXTRACT PDBN202 starting


GGSCI (c1ds2.oracle.com) 11> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt
```

```
MANAGER     RUNNING
EXTRACT     STARTING    EDBN201     00:35:00      00:00:32
EXTRACT     RUNNING     PDBN201     00:00:00      00:00:08
EXTRACT     RUNNING     PDBN202     00:00:00      00:00:08


GGSCI (c1ds2.oracle.com) 12> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     STARTING    EDBN201     00:35:00      00:00:39
EXTRACT     RUNNING     PDBN201     00:00:00      00:00:04
EXTRACT     RUNNING     PDBN202     00:00:00      00:00:04


GGSCI (c1ds2.oracle.com) 13> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN201     00:35:42      00:00:04
EXTRACT     RUNNING     PDBN201     00:00:00      00:00:01
EXTRACT     RUNNING     PDBN202     00:00:00      00:00:01


GGSCI (c1ds2.oracle.com) 14>
```

```
GGSCI (edbcn0.oracle.com) 27> stop *

Sending STOP request to REPLICAT RDBN101 ...
Request processed.

Sending STOP request to REPLICAT RDBN201 ...
Request processed.


GGSCI (edbcn0.oracle.com) 28> start *

Sending START request to MANAGER ...
REPLICAT RDBN101 starting

Sending START request to MANAGER ...
REPLICAT RDBN201 starting


GGSCI (edbcn0.oracle.com) 29> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN101     00:00:00      00:00:07
REPLICAT    RUNNING     RDBN201     00:00:00      00:00:07


GGSCI (edbcn0.oracle.com) 30>
```

```
GGSCI (edbcn1.oracle.com) 15> stop *

Sending STOP request to REPLICAT RDBN001 ...
Request processed.

Sending STOP request to REPLICAT RDBN202 ...
Request processed.

GGSCI (edbcn1.oracle.com) 16> start *

Sending START request to MANAGER ...
REPLICAT RDBN001 starting

Sending START request to MANAGER ...
REPLICAT RDBN202 starting


GGSCI (edbcn1.oracle.com) 17> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN001     00:00:00      00:00:04
REPLICAT    RUNNING     RDBN202     00:00:00      00:00:04


GGSCI (edbcn1.oracle.com) 18>
```

Task 3: Testing for INSERT Conflict Detection and Resolution

Applications inter-operability with Oracle GoldenGate 12c must be developed and applied. Data conflict may occurs due to network disconnect, application logic and database design. Figure 12 depicts the level of data conflict handling techniques following a network failure.

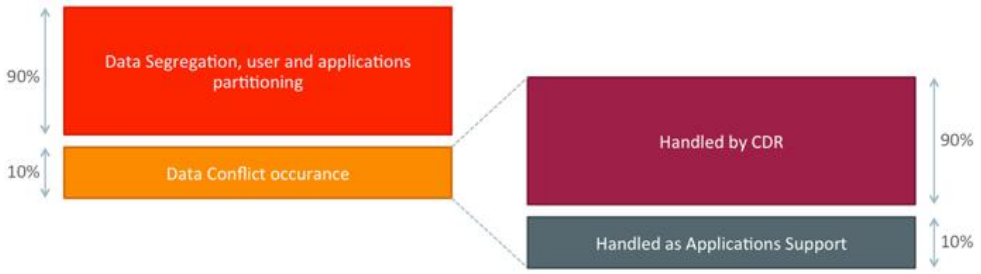


Figure 12: General Data Conflict Handlings

The following steps demonstrate Conflict Detection and Resolution (CDR) following a network disconnect.

Disconnecting Network (Simulation)

Stop the data pumps running at the downstream systems C1DS1A, C1DS1B and C1DS2A.

On downstream system C<n>DS1A

```
GGSCI (c1ds1.oracle.com) 4> stop PDBN001

Sending STOP request to EXTRACT PDBN001 ...
Request processed.


GGSCI (c1ds1.oracle.com) 5> stop PDBN002

Sending STOP request to EXTRACT PDBN002 ...
Request processed.


GGSCI (c1ds1.oracle.com) 6> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN001     00:00:06      00:00:04
EXTRACT     STOPPED     PDBN001     00:00:00      00:00:10
EXTRACT     STOPPED     PDBN002     00:00:00      00:00:02


GGSCI (c1ds1.oracle.com) 7>
```

On downstream system C<n>DS1AB

```
GGSCI (c1ds1.oracle.com) 4> stop PDBN101

Sending STOP request to EXTRACT PDBN101 ...
Request processed.


GGSCI (c1ds1.oracle.com) 5> stop PDBN102

Sending STOP request to EXTRACT PDBN102 ...
Request processed.


GGSCI (c1ds1.oracle.com) 6> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN101     00:00:07      00:00:01
EXTRACT     STOPPED     PDBN101     00:00:00      00:00:07
EXTRACT     STOPPED     PDBN102     00:00:00      00:00:02
```

```
GGSCI (c1ds1.oracle.com) 7>
```

On downstream system C<n>DS2A

```
GGSCI (c1ds2.oracle.com) 4> stop PDBN201

Sending STOP request to EXTRACT PDBN201 ...
Request processed.


GGSCI (c1ds2.oracle.com) 5> stop PDBN202

Sending STOP request to EXTRACT PDBN202 ...
Request processed.

GGSCI (c1ds2.oracle.com) 6> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN201     01:26:09      00:00:06
EXTRACT     STOPPED     PDBN201     00:00:00      00:00:06
EXTRACT     STOPPED     PDBN202     00:00:00      00:00:01


GGSCI (c1ds2.oracle.com) 7
```

Run the Workload Simulator for INSERT

Let the session execute for approximately 5 minutes, then terminate by pressing CTL-C.

On the primary node

```
SQL> @call_trans_load_2.sql

Procedure created.

SQL> exec call_trans_load(pDML=>'I', pCDR=>'Y');
^CBEGIN call trans load(pDML=>'I', pCDR=>'Y'); END;

*
ERROR at line 1:
ORA-01013: user requested cancel of current operation
ORA-06512: at "SYS.DBMS_LOCK", line 205
ORA-06512: at "OSM REPAPI.CALL TRANS LOAD", line 20
ORA-06512: at line 1



SQL>
```

Start the Data Pumps

Start the data pumps on downstream systems C1DS1A, C1DS1B and C1DS2A.

On downstream system C<n>DS1A

```
GGSCI (c1ds1.oracle.com) 15> start *
EXTRACT EDBN001 is already running.

Sending START request to MANAGER ...
EXTRACT PDBN001 starting

Sending START request to MANAGER ...
EXTRACT PDBN002 starting


GGSCI (c1ds1.oracle.com) 16> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN001     00:00:13      00:00:09
EXTRACT     RUNNING     PDBN001     00:00:00      00:08:00
EXTRACT     RUNNING     PDBN002     00:00:00      00:07:54


GGSCI (c1ds1.oracle.com) 17>
```

On downstream system C<n>DS1B

```
GGSCI (c1ds1.oracle.com) 13> start *
EXTRACT EDBN101 is already running.

Sending START request to MANAGER ...
EXTRACT PDBN101 starting

Sending START request to MANAGER ...
EXTRACT PDBN102 starting


GGSCI (c1ds1.oracle.com) 14> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN101     00:00:05      00:00:05
EXTRACT     RUNNING     PDBN101     00:00:00      00:07:55
EXTRACT     RUNNING     PDBN102     00:00:00      00:07:50


GGSCI (c1ds1.oracle.com) 15>
```

```
GGSCI (c1ds2.oracle.com) 14> start *
EXTRACT EDBN201 is already running.
```

```
Sending START request to MANAGER ...
EXTRACT PDBN201 starting

Sending START request to MANAGER ...
EXTRACT PDBN202 starting


GGSCI (c1ds2.oracle.com) 15> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN201     02:18:09      00:00:04
EXTRACT     RUNNING     PDBN201     00:00:00      00:07:50
EXTRACT     RUNNING     PDBN202     00:00:00      00:07:45


GGSCI (c1ds2.oracle.com) 16>
```

Review Current Conflict Detection and Resolution (CDR) Reports

Insert data conflicts are detected and resolved on extended node 1

```
GGSCI (edbcn1.oracle.com) 18> stats replicat RDBN001, reportcdr

Sending STATS request to REPLICAT RDBN001 ...

Start of Statistics at 2017-01-09 14:27:24.

Replicating from OSM REPAPI.CUSTOMERS to OSM REPAPI.CUSTOMERS:

*** Total statistics since 2017-01-08 18:36:26 ***
    Total inserts                                 484.00
    Total updates                              192847.00
    Total deletes                                3568.00
    Total discards                                  0.00
    Total operations                           196899.00
    Total CDR conflicts                                    178652.00
    CDR resolutions succeeded                              178652.00
    CDR INSERTROWEXISTS conflicts                              62.00
    CDR UPDATEROWEXISTS conflicts                          175229.00
    CDR DELETEROWMISSING conflicts                           3361.00

*** Daily statistics since 2017-01-09 00:00:00 ***
    Total inserts                                 453.00
    Total updates                              161900.00
    Total deletes                                 278.00
    Total discards                                  0.00
    Total operations                           162631.00
    Total CDR conflicts                                    162002.00
    CDR resolutions succeeded                              162002.00
    CDR INSERTROWEXISTS conflicts                              32.00
    CDR UPDATEROWEXISTS conflicts                          161899.00
    CDR DELETEROWMISSING conflicts                             71.00
```

```
*** Hourly statistics since 2017-01-09 14:00:00 ***
    Total inserts                                 31.00
    Total updates                                  0.00
    Total deletes                                  0.00
    Total discards                                 0.00
    Total operations                              31.00

*** Latest statistics since 2017-01-08 18:36:26 ***
    Total inserts                                484.00
    Total updates                             192847.00
    Total deletes                               3568.00
    Total discards                                 0.00
    Total operations                          196899.00
    Total CDR conflicts                                    178652.00
    CDR resolutions succeeded                              178652.00
    CDR INSERTROWEXISTS conflicts                              62.00
    CDR UPDATEROWEXISTS conflicts                          175229.00
    CDR DELETEROWMISSING conflicts                           3361.00

Replicating from OSM_REPAPI.POLICIES to OSM_REPAPI.POLICIES:

*** Total statistics since 2017-01-08 18:36:26 ***
    Total inserts                                 482.00
    Total updates                                   0.00
    Total deletes                                   0.00
    Total discards                                  0.00
    Total operations                              482.00
    Total CDR conflicts                                        62.00
    CDR resolutions succeeded                                  62.00
    CDR INSERTROWEXISTS conflicts                              62.00

*** Daily statistics since 2017-01-09 00:00:00 ***
    Total inserts                                 452.00
    Total updates                                   0.00
    Total deletes                                   0.00
    Total discards                                  0.00
    Total operations                              452.00
    Total CDR conflicts                                        32.00
    CDR resolutions succeeded                                  32.00
    CDR INSERTROWEXISTS conflicts                              32.00

*** Hourly statistics since 2017-01-09 14:00:00 ***
    Total inserts                                 30.00
    Total updates                                  0.00
    Total deletes                                  0.00
    Total discards                                 0.00
    Total operations                              30.00

*** Latest statistics since 2017-01-08 18:36:26 ***
    Total inserts                                 482.00
    Total updates                                   0.00
    Total deletes                                   0.00
    Total discards                                  0.00
    Total operations                              482.00
    Total CDR conflicts                                        62.00
    CDR resolutions succeeded                                  62.00
    CDR INSERTROWEXISTS conflicts                              62.00

End of Statistics.
```

```
GGSCI (edbcn1.oracle.com) 19>
```

Insert data conflict detected and resolved on extended node 2

```
GGSCI (edbcn2.oracle.com) 8> stats replicat RDBN002, reportcdr

Sending STATS request to REPLICAT RDBN002 ...

Start of Statistics at 2017-01-09 14:28:22.

Replicating from OSM REPAPI.CUSTOMERS to OSM REPAPI.CUSTOMERS:

*** Total statistics since 2017-01-08 18:35:37 ***
    Total inserts                                 484.00
    Total updates                              192847.00
    Total deletes                                3568.00
    Total discards                                  0.00
    Total operations                           196899.00
    Total CDR conflicts                                    180148.00
    CDR resolutions succeeded                              180148.00
    CDR INSERTROWEXISTS conflicts                              62.00
    CDR UPDATEROWEXISTS conflicts                          176796.00
    CDR DELETEROWMISSING conflicts                           3290.00

*** Daily statistics since 2017-01-09 00:00:00 ***
    Total inserts                                 453.00
    Total updates                              161900.00
    Total deletes                                 278.00
    Total discards                                  0.00
    Total operations                           162631.00
    Total CDR conflicts                                    161931.00
    CDR resolutions succeeded                              161931.00
    CDR INSERTROWEXISTS conflicts                              32.00
    CDR UPDATEROWEXISTS conflicts                          161899.00

*** Hourly statistics since 2017-01-09 14:00:00 ***
    Total inserts                                  31.00
    Total updates                                   0.00
    Total deletes                                   0.00
    Total discards                                  0.00
    Total operations                               31.00

*** Latest statistics since 2017-01-08 18:35:37 ***
    Total inserts                                 484.00
    Total updates                              192847.00
    Total deletes                                3568.00
    Total discards                                  0.00
    Total operations                           196899.00
    Total CDR conflicts                                    180148.00
    CDR resolutions succeeded                              180148.00
    CDR INSERTROWEXISTS conflicts                              62.00
    CDR UPDATEROWEXISTS conflicts                          176796.00
    CDR DELETEROWMISSING conflicts                           3290.00

Replicating from OSM_REPAPI.POLICIES to OSM_REPAPI.POLICIES:
```

```
*** Total statistics since 2017-01-08 18:35:37 ***
    Total inserts                                482.00
    Total updates                                  0.00
    Total deletes                                  0.00
    Total discards                                 0.00
    Total operations                             482.00
    Total CDR conflicts                                    60.00
    CDR resolutions succeeded                              60.00
    CDR INSERTROWEXISTS conflicts                          60.00

*** Daily statistics since 2017-01-09 00:00:00 ***
    Total inserts                                452.00
    Total updates                                  0.00
    Total deletes                                  0.00
    Total discards                                 0.00
    Total operations                             452.00
    Total CDR conflicts                                    30.00
    CDR resolutions succeeded                              30.00
    CDR INSERTROWEXISTS conflicts                          30.00

*** Hourly statistics since 2017-01-09 14:00:00 ***
    Total inserts                                 30.00
    Total updates                                  0.00
    Total deletes                                  0.00
    Total discards                                 0.00
    Total operations                              30.00

*** Latest statistics since 2017-01-08 18:35:37 ***
    Total inserts                                482.00
    Total updates                                  0.00
    Total deletes                                  0.00
    Total discards                                 0.00
    Total operations                             482.00
    Total CDR conflicts                                    60.00
    CDR resolutions succeeded                              60.00
    CDR INSERTROWEXISTS conflicts                          60.00

End of Statistics.


GGSCI (edbcn2.oracle.com) 9>
```

Task 4: Testing for UPDATE Conflict Detection and Resolution

The following steps demonstrate Conflict Detection and Resolution (CDR) following a network disconnect.

Disconnecting Network (Simulation)

Stop the data pumps running at the downstream systems C1DS1A, C1DS1B and C1DS2A

On downstream system C<n>DS1A

```
GGSCI (c1ds1.oracle.com) 18> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN001     00:00:10      00:00:06
EXTRACT     RUNNING     PDBN001     00:00:00      00:00:07
EXTRACT     RUNNING     PDBN002     00:00:00      00:00:06


GGSCI (c1ds1.oracle.com) 19> stop PDBN001

Sending STOP request to EXTRACT PDBN001 ...
Request processed.


GGSCI (c1ds1.oracle.com) 20> stop PDBN002
Request processed.

GGSCI (c1ds1.oracle.com) 21>
```

On downstream system C<n>DS1B

```
GGSCI (c1ds1.oracle.com) 16> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN101     00:00:03      00:00:07
EXTRACT     RUNNING     PDBN101     00:00:00      00:00:02
EXTRACT     RUNNING     PDBN102     00:00:00      00:00:02


GGSCI (c1ds1.oracle.com) 17> stop PDBN101

Sending STOP request to EXTRACT PDBN101 ...
Request processed.


GGSCI (c1ds1.oracle.com) 18> stop PDBN102

Sending STOP request to EXTRACT PDBN102 ...
Request processed.


GGSCI (c1ds1.oracle.com) 19>
```

On downstream system C<n>DS2A

```
GGSCI (c1ds2.oracle.com) 17> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt
```

```
MANAGER     RUNNING
EXTRACT     RUNNING     EDBN201     03:12:00      00:00:00
EXTRACT     RUNNING     PDBN201     00:00:00      00:00:01
EXTRACT     RUNNING     PDBN202     00:00:00      00:00:01


GGSCI (c1ds2.oracle.com) 18> stop PDBN201

Sending STOP request to EXTRACT PDBN201 ...
Request processed.


GGSCI (c1ds2.oracle.com) 19> stop PDBN202

Sending STOP request to EXTRACT PDBN202 ...
Request processed.


GGSCI (c1ds2.oracle.com) 20>
```

Run the Workload Simulator for UPDATE

The workload simulator passed the parameters pDML for update (U) and to process CDR (Y) pCDR. The default bucket size (row set) is 50 rows.

On the primary node

```
SQL> exec call_trans_load(pDML=>'U', pCDR=>'Y');

PL/SQL procedure successfully completed.

SQL>
```

Re-Connect the network

On downstream system C<n>DS1A

```
GGSCI (c1ds1.oracle.com) 24> start *
EXTRACT EDBN001 is already running.

Sending START request to MANAGER ...
EXTRACT PDBN001 starting

Sending START request to MANAGER ...
EXTRACT PDBN002 starting


GGSCI (c1ds1.oracle.com) 25> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
```

```
EXTRACT     RUNNING     EDBN001     00:00:12      00:00:05
EXTRACT     RUNNING     PDBN001     00:00:00      00:50:59
EXTRACT     RUNNING     PDBN002     00:00:00      00:48:42


GGSCI (c1ds1.oracle.com) 26>
```

On downstream system C<n>DS1B

```
GGSCI (c1ds1.oracle.com) 20> start *
EXTRACT EDBN101 is already running.

Sending START request to MANAGER ...
EXTRACT PDBN101 starting

Sending START request to MANAGER ...
EXTRACT PDBN102 starting


GGSCI (c1ds1.oracle.com) 21> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN101     00:00:02      00:00:00
EXTRACT     RUNNING     PDBN101     00:00:00      00:51:14
EXTRACT     RUNNING     PDBN102     00:00:00      00:51:09


GGSCI (c1ds1.oracle.com) 22>
```

On downstream system C<n>DS2A

```
GGSCI (c1ds2.oracle.com) 21> start *
EXTRACT EDBN201 is already running.

Sending START request to MANAGER ...
EXTRACT PDBN201 starting

Sending START request to MANAGER ...
EXTRACT PDBN202 starting


GGSCI (c1ds2.oracle.com) 22> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN201     04:04:10      00:00:03
EXTRACT     RUNNING     PDBN201     00:00:00      00:51:35
EXTRACT     RUNNING     PDBN202     00:00:00      00:51:29

GGSCI (c1ds2.oracle.com) 23>
```

Review Current Conflict Detection and Resolution (CDR) Reports

On extended node 1

```
GGSCI (edbcn1.oracle.com) 24> stats replicat RDBN001, reportcdr

Sending STATS request to REPLICAT RDBN001 ...

Start of Statistics at 2017-01-09 15:04:43.

Replicating from OSM REPAPI.CUSTOMERS to OSM REPAPI.CUSTOMERS:

*** Total statistics since 2017-01-08 18:36:26 ***
    Total inserts                                593.00
    Total updates                             192945.00
    Total deletes                               3568.00
    Total discards                                 0.00
    Total operations                          197106.00
    Total CDR conflicts                                    178732.00
    CDR resolutions succeeded                              178732.00
    CDR INSERTROWEXISTS conflicts                              62.00
    CDR UPDATEROWEXISTS conflicts                          175309.00
    CDR DELETEROWMISSING conflicts                           3361.00

*** Daily statistics since 2017-01-09 00:00:00 ***
    Total inserts                                562.00
    Total updates                             161998.00
    Total deletes                                278.00
    Total discards                                 0.00
    Total operations                          162838.00
    Total CDR conflicts                                    162082.00
    CDR resolutions succeeded                              162082.00
    CDR INSERTROWEXISTS conflicts                              32.00
    CDR UPDATEROWEXISTS conflicts                          161979.00
    CDR DELETEROWMISSING conflicts                             71.00

*** Hourly statistics since 2017-01-09 15:00:00 ***

    No database operations have been performed.

*** Latest statistics since 2017-01-08 18:36:26 ***
    Total inserts                                593.00
    Total updates                             192945.00
    Total deletes                               3568.00
    Total discards                                 0.00
    Total operations                          197106.00
    Total CDR conflicts                                    178732.00
    CDR resolutions succeeded                              178732.00
    CDR INSERTROWEXISTS conflicts                              62.00
    CDR UPDATEROWEXISTS conflicts                          175309.00
    CDR DELETEROWMISSING conflicts                           3361.00

Replicating from OSM REPAPI.POLICIES to OSM REPAPI.POLICIES:

*** Total statistics since 2017-01-08 18:36:26 ***
    Total inserts                                592.00
    Total updates                                  0.00
```

```
    Total deletes                                0.00
    Total discards                               0.00
    Total operations                           592.00
    Total CDR conflicts                                      62.00
    CDR resolutions succeeded                                62.00
    CDR INSERTROWEXISTS conflicts                            62.00

*** Daily statistics since 2017-01-09 00:00:00 ***
    Total inserts                              562.00
    Total updates                                0.00
    Total deletes                                0.00
    Total discards                               0.00
    Total operations                           562.00
    Total CDR conflicts                                      32.00
    CDR resolutions succeeded                                32.00
    CDR INSERTROWEXISTS conflicts                            32.00

*** Hourly statistics since 2017-01-09 15:00:00 ***

    No database operations have been performed.

*** Latest statistics since 2017-01-08 18:36:26 ***
    Total inserts                              592.00
    Total updates                                0.00
    Total deletes                                0.00
    Total discards                               0.00
    Total operations                           592.00
    Total CDR conflicts                                      62.00
    CDR resolutions succeeded                                62.00
    CDR INSERTROWEXISTS conflicts                            62.00

End of Statistics.

GGSCI (edbcn1.oracle.com) 25>
```

On extended node 2

```
GGSCI (edbcn2.oracle.com) 8> stats replicat RDBN002, reportcdr

Sending STATS request to REPLICAT RDBN002 ...

Start of Statistics at 2017-01-09 14:28:22.

Replicating from OSM REPAPI.CUSTOMERS to OSM REPAPI.CUSTOMERS:

*** Total statistics since 2017-01-08 18:35:37 ***
    Total inserts                                 484.00
    Total updates                              192847.00
    Total deletes                                3568.00
    Total discards                                  0.00
    Total operations                           196899.00
    Total CDR conflicts                                    180148.00
    CDR resolutions succeeded                              180148.00
    CDR INSERTROWEXISTS conflicts                              62.00
    CDR UPDATEROWEXISTS conflicts                          176796.00
    CDR DELETEROWMISSING conflicts                           3290.00
```

```
*** Daily statistics since 2017-01-09 00:00:00 ***
     Total inserts                                 453.00
     Total updates                              161900.00
     Total deletes                                 278.00
     Total discards                                  0.00
     Total operations                           162631.00
     Total CDR conflicts                                    161931.00
     CDR resolutions succeeded                              161931.00
     CDR INSERTROWEXISTS conflicts                              32.00
     CDR UPDATEROWEXISTS conflicts                          161899.00

*** Hourly statistics since 2017-01-09 14:00:00 ***
     Total inserts                                  31.00
     Total updates                                   0.00
     Total deletes                                   0.00
     Total discards                                  0.00
     Total operations                               31.00

*** Latest statistics since 2017-01-08 18:35:37 ***
     Total inserts                                 484.00
     Total updates                              192847.00
     Total deletes                                3568.00
     Total discards                                  0.00
     Total operations                           196899.00
     Total CDR conflicts                                    180148.00
     CDR resolutions succeeded                              180148.00
     CDR INSERTROWEXISTS conflicts                              62.00
     CDR UPDATEROWEXISTS conflicts                          176796.00
     CDR DELETEROWMISSING conflicts                           3290.00

Replicating from OSM_REPAPI.POLICIES to OSM_REPAPI.POLICIES:

*** Total statistics since 2017-01-08 18:35:37 ***
     Total inserts                                 482.00
     Total updates                                   0.00
     Total deletes                                   0.00
     Total discards                                  0.00
     Total operations                              482.00
     Total CDR conflicts                                     60.00
     CDR resolutions succeeded                               60.00
     CDR INSERTROWEXISTS conflicts                           60.00

*** Daily statistics since 2017-01-09 00:00:00 ***
     Total inserts                                 452.00
     Total updates                                   0.00
     Total deletes                                   0.00
     Total discards                                  0.00
     Total operations                              452.00
     Total CDR conflicts                                     30.00
     CDR resolutions succeeded                               30.00
     CDR INSERTROWEXISTS conflicts                           30.00

*** Hourly statistics since 2017-01-09 14:00:00 ***
     Total inserts                                  30.00
     Total updates                                   0.00
     Total deletes                                   0.00
     Total discards                                  0.00
     Total operations                               30.00
```

```
*** Latest statistics since 2017-01-08 18:35:37 ***
    Total inserts                                482.00
    Total updates                                  0.00
    Total deletes                                  0.00
    Total discards                                 0.00
    Total operations                             482.00
    Total CDR conflicts                                    60.00
    CDR resolutions succeeded                              60.00
    CDR INSERTROWEXISTS conflicts                          60.00

End of Statistics.

GGSCI (edbcn2.oracle.com) 9> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN002     00:00:00      00:00:04
REPLICAT    RUNNING     RDBN102     00:00:00      00:00:05


GGSCI (edbcn2.oracle.com) 10> stats replicat RDBN002

Sending STATS request to REPLICAT RDBN002 ...

Start of Statistics at 2017-01-09 15:05:24.

Replicating from OSM REPAPI.CUSTOMERS to OSM REPAPI.CUSTOMERS:

*** Total statistics since 2017-01-08 18:35:37 ***
    Total inserts                                593.00
    Total updates                             192945.00
    Total deletes                               3568.00
    Total discards                                 0.00
    Total operations                          197106.00

*** Daily statistics since 2017-01-09 00:00:00 ***
    Total inserts                                562.00
    Total updates                             161998.00
    Total deletes                                278.00
    Total discards                                 0.00
    Total operations                          162838.00

*** Hourly statistics since 2017-01-09 15:00:00 ***

    No database operations have been performed.

*** Latest statistics since 2017-01-08 18:35:37 ***
    Total inserts                                593.00
    Total updates                             192945.00
    Total deletes                               3568.00
    Total discards                                 0.00
    Total operations                          197106.00

Replicating from OSM REPAPI.POLICIES to OSM REPAPI.POLICIES:

*** Total statistics since 2017-01-08 18:35:37 ***
    Total inserts                                592.00
    Total updates                                  0.00
```

```
    Total deletes                                0.00
    Total discards                               0.00
    Total operations                           592.00

*** Daily statistics since 2017-01-09 00:00:00 ***
    Total inserts                              562.00
    Total updates                                0.00
    Total deletes                                0.00
    Total discards                               0.00
    Total operations                           562.00

*** Hourly statistics since 2017-01-09 15:00:00 ***

    No database operations have been performed.

*** Latest statistics since 2017-01-08 18:35:37 ***
    Total inserts                              592.00
    Total updates                                0.00
    Total deletes                                0.00
    Total discards                               0.00
    Total operations                           592.00

End of Statistics.

GGSCI (edbcn2.oracle.com) 11>
```

Task 5: Testing for DELETE Conflict Detection and Resolution

The following steps demonstrate Conflict Detection and Resolution (CDR) following a network disconnect.

Disconnecting Network (Simulation)

Stop the data pumps running at the downstream systems C1DS1A, C1DS1B and C1DS2A

💻 *On downstream system C<n>DS1A*

```
GGSCI (c1ds1.oracle.com) 54> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN001     00:00:08      00:00:09
EXTRACT     RUNNING     PDBN001     00:00:00      00:00:04
EXTRACT     RUNNING     PDBN002     00:00:00      00:00:04


GGSCI (c1ds1.oracle.com) 55> stop PDBN001

Sending STOP request to EXTRACT PDBN001 ...
Request processed.
```

```
GGSCI (c1ds1.oracle.com) 56> stop PDBN002

Sending STOP request to EXTRACT PDBN002 ...
Request processed.


GGSCI (c1ds1.oracle.com) 57> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN001     00:00:09      00:00:03
EXTRACT     STOPPED     PDBN001     00:00:00      00:00:27
EXTRACT     STOPPED     PDBN002     00:00:00      00:00:22


GGSCI (c1ds1.oracle.com) 58>
```

On downstream system C<n>DS1B

```
GGSCI (c1ds1.oracle.com) 43> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN101     00:00:08      00:00:03
EXTRACT     RUNNING     PDBN101     00:00:00      00:00:06
EXTRACT     RUNNING     PDBN102     00:00:00      00:00:06


GGSCI (c1ds1.oracle.com) 44> stop PDBN101

Sending STOP request to EXTRACT PDBN101 ...
Request processed.


GGSCI (c1ds1.oracle.com) 45> stop PDBN102

Sending STOP request to EXTRACT PDBN102 ...
Request processed.


GGSCI (c1ds1.oracle.com) 46> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN101     00:00:08      00:00:03
EXTRACT     STOPPED     PDBN101     00:00:00      00:00:05
EXTRACT     STOPPED     PDBN102     00:00:00      00:00:01


GGSCI (c1ds1.oracle.com) 47>
```

On downstream system C<n>DS2A

```
GGSCI (c1ds2.oracle.com) 42> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN201     01:39:20      00:00:06
EXTRACT     RUNNING     PDBN201     00:00:00      00:00:04
EXTRACT     RUNNING     PDBN202     00:00:00      00:00:04


GGSCI (c1ds2.oracle.com) 43> stop PDBN201

Sending STOP request to EXTRACT PDBN201 ...
Request processed.


GGSCI (c1ds2.oracle.com) 44> stop PDBN202

Sending STOP request to EXTRACT PDBN202 ...
Request processed.


GGSCI (c1ds2.oracle.com) 45> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN201     01:39:30      00:00:07
EXTRACT     STOPPED     PDBN201     00:00:00      00:00:05
EXTRACT     STOPPED     PDBN202     00:00:00      00:00:01


GGSCI (c1ds2.oracle.com) 46>
```

Run the Workload Simulator for UPDATE

On the primary node

```
SQL> exec call_trans_load(pDML=>'D', pCDR=>'Y');

PL/SQL procedure successfully completed.

SQL>
```

Re-Connect the network

On downstream system C<n>DS1A

```
GGSCI (c1ds1.oracle.com) 58> start *
```

```
EXTRACT EDBN001 is already running.

Sending START request to MANAGER ...
EXTRACT PDBN001 starting

Sending START request to MANAGER ...
EXTRACT PDBN002 starting


GGSCI (c1ds1.oracle.com) 59> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN001     00:00:09      00:00:08
EXTRACT     RUNNING     PDBN001     00:00:00      00:00:03
EXTRACT     RUNNING     PDBN002     00:00:00      00:00:03


GGSCI (c1ds1.oracle.com) 60>
```

On downstream system C<n>DS1B

```
GGSCI (c1ds1.oracle.com) 47> start *
EXTRACT EDBN101 is already running.

Sending START request to MANAGER ...
EXTRACT PDBN101 starting

Sending START request to MANAGER ...
EXTRACT PDBN102 starting


GGSCI (c1ds1.oracle.com) 48> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN101     00:00:06      00:00:08
EXTRACT     RUNNING     PDBN101     00:00:00      00:00:06
EXTRACT     RUNNING     PDBN102     00:00:00      00:00:06


GGSCI (c1ds1.oracle.com) 49>
```

On downstream system C<n>DS2A

```
GGSCI (c1ds2.oracle.com) 46> start *
EXTRACT EDBN201 is already running.

Sending START request to MANAGER ...
EXTRACT PDBN201 starting

Sending START request to MANAGER ...
EXTRACT PDBN202 starting
```

```
GGSCI (c1ds2.oracle.com) 47> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN201     01:41:40      00:00:08
EXTRACT     RUNNING     PDBN201     00:00:00      00:00:09
EXTRACT     RUNNING     PDBN202     00:00:00      00:00:08


GGSCI (c1ds2.oracle.com) 48>
```

Review Current Conflict Detection and Resolution (CDR) Reports

On extended node 1

```
GGSCI (edbcn1.oracle.com) 26> stats replicat RDBN001, reportcdr

Sending STATS request to REPLICAT RDBN001 ...

Start of Statistics at 2017-01-09 15:13:23.

Replicating from OSM REPAPI.CUSTOMERS to OSM REPAPI.CUSTOMERS:

*** Total statistics since 2017-01-08 18:36:26 ***
    Total inserts                                593.00
    Total updates                             192945.00
    Total deletes                               3689.00
    Total discards                                 0.00
    Total operations                          197227.00
    Total CDR conflicts                                    178732.00
    CDR resolutions succeeded                              178732.00
    CDR INSERTROWEXISTS conflicts                              62.00
    CDR UPDATEROWEXISTS conflicts                          175309.00
    CDR DELETEROWMISSING conflicts                           3361.00

*** Daily statistics since 2017-01-09 00:00:00 ***
    Total inserts                                562.00
    Total updates                             161998.00
    Total deletes                                399.00
    Total discards                                 0.00
    Total operations                          162959.00
    Total CDR conflicts                                    162082.00
    CDR resolutions succeeded                              162082.00
    CDR INSERTROWEXISTS conflicts                              32.00
    CDR UPDATEROWEXISTS conflicts                          161979.00
    CDR DELETEROWMISSING conflicts                             71.00

*** Hourly statistics since 2017-01-09 15:00:00 ***
    Total inserts                                  0.00
    Total updates                                  0.00
    Total deletes                                121.00
    Total discards                                 0.00
    Total operations                             121.00
```

```
*** Latest statistics since 2017-01-08 18:36:26 ***
    Total inserts                                593.00
    Total updates                             192945.00
    Total deletes                               3689.00
    Total discards                                 0.00
    Total operations                          197227.00
    Total CDR conflicts                                    178732.00
    CDR resolutions succeeded                              178732.00
    CDR INSERTROWEXISTS conflicts                              62.00
    CDR UPDATEROWEXISTS conflicts                          175309.00
    CDR DELETEROWMISSING conflicts                           3361.00

Replicating from OSM_REPAPI.POLICIES to OSM_REPAPI.POLICIES:

*** Total statistics since 2017-01-08 18:36:26 ***
    Total inserts                                592.00
    Total updates                                  0.00
    Total deletes                                  0.00
    Total discards                                 0.00
    Total operations                             592.00
    Total CDR conflicts                                     62.00
    CDR resolutions succeeded                               62.00
    CDR INSERTROWEXISTS conflicts                           62.00

*** Daily statistics since 2017-01-09 00:00:00 ***
    Total inserts                                562.00
    Total updates                                  0.00
    Total deletes                                  0.00
    Total discards                                 0.00
    Total operations                             562.00
    Total CDR conflicts                                     32.00
    CDR resolutions succeeded                               32.00
    CDR INSERTROWEXISTS conflicts                           32.00

*** Hourly statistics since 2017-01-09 15:00:00 ***

    No database operations have been performed.

*** Latest statistics since 2017-01-08 18:36:26 ***
    Total inserts                                592.00
    Total updates                                  0.00
    Total deletes                                  0.00
    Total discards                                 0.00
    Total operations                             592.00
    Total CDR conflicts                                     62.00
    CDR resolutions succeeded                               62.00
    CDR INSERTROWEXISTS conflicts                           62.00

End of Statistics.


GGSCI (edbcn1.oracle.com) 27>
```

On extended node 2

```
GGSCI (edbcn2.oracle.com) 10> stats replicat RDBN002
```

```
Sending STATS request to REPLICAT RDBN002 ...

Start of Statistics at 2017-01-09 15:05:24.

Replicating from OSM_REPAPI.CUSTOMERS to OSM_REPAPI.CUSTOMERS:

*** Total statistics since 2017-01-08 18:35:37 ***
     Total inserts                                593.00
     Total updates                             192945.00
     Total deletes                               3568.00
     Total discards                                 0.00
     Total operations                          197106.00

*** Daily statistics since 2017-01-09 00:00:00 ***
     Total inserts                                562.00
     Total updates                             161998.00
     Total deletes                                278.00
     Total discards                                 0.00
     Total operations                          162838.00

*** Hourly statistics since 2017-01-09 15:00:00 ***

     No database operations have been performed.

*** Latest statistics since 2017-01-08 18:35:37 ***
     Total inserts                                593.00
     Total updates                             192945.00
     Total deletes                               3568.00
     Total discards                                 0.00
     Total operations                          197106.00

Replicating from OSM REPAPI.POLICIES to OSM REPAPI.POLICIES:

*** Total statistics since 2017-01-08 18:35:37 ***
     Total inserts                                592.00
     Total updates                                  0.00
     Total deletes                                  0.00
     Total discards                                 0.00
     Total operations                             592.00

*** Daily statistics since 2017-01-09 00:00:00 ***
     Total inserts                                562.00
     Total updates                                  0.00
     Total deletes                                  0.00
     Total discards                                 0.00
     Total operations                             562.00

*** Hourly statistics since 2017-01-09 15:00:00 ***

     No database operations have been performed.

*** Latest statistics since 2017-01-08 18:35:37 ***
     Total inserts                                592.00
     Total updates                                  0.00
     Total deletes                                  0.00
     Total discards                                 0.00
     Total operations                             592.00

End of Statistics.
```

```
GGSCI (edbcn2.oracle.com) 11>

GGSCI (edbcn2.oracle.com) 11>

GGSCI (edbcn2.oracle.com) 11> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN002     00:00:00      00:00:00
REPLICAT    RUNNING     RDBN102     00:00:00      00:00:04


GGSCI (edbcn2.oracle.com) 12> stats replicat RDBN002, reportcdr

Sending STATS request to REPLICAT RDBN002 ...

Start of Statistics at 2017-01-09 15:14:14.

Replicating from OSM_REPAPI.CUSTOMERS to OSM_REPAPI.CUSTOMERS:

*** Total statistics since 2017-01-08 18:35:37 ***
    Total inserts                                593.00
    Total updates                             192945.00
    Total deletes                               3689.00
    Total discards                                 0.00
    Total operations                          197227.00
    Total CDR conflicts                          180228.00
    CDR resolutions succeeded                    180228.00
    CDR INSERTROWEXISTS conflicts                    62.00
    CDR UPDATEROWEXISTS conflicts                176876.00
    CDR DELETEROWMISSING conflicts                 3290.00

*** Daily statistics since 2017-01-09 00:00:00 ***
    Total inserts                                562.00
    Total updates                             161998.00
    Total deletes                                399.00
    Total discards                                 0.00
    Total operations                          162959.00
    Total CDR conflicts                          162011.00
    CDR resolutions succeeded                    162011.00
    CDR INSERTROWEXISTS conflicts                    32.00
    CDR UPDATEROWEXISTS conflicts                161979.00

*** Hourly statistics since 2017-01-09 15:00:00 ***
    Total inserts                                  0.00
    Total updates                                  0.00
    Total deletes                                121.00
    Total discards                                 0.00
    Total operations                             121.00

*** Latest statistics since 2017-01-08 18:35:37 ***
    Total inserts                                593.00
    Total updates                             192945.00
    Total deletes                               3689.00
    Total discards                                 0.00
    Total operations                          197227.00
```

```
    Total CDR conflicts                                 180228.00
    CDR resolutions succeeded                           180228.00
    CDR INSERTROWEXISTS conflicts                           62.00
    CDR UPDATEROWEXISTS conflicts                       176876.00
    CDR DELETEROWMISSING conflicts                        3290.00

Replicating from OSM_REPAPI.POLICIES to OSM_REPAPI.POLICIES:

*** Total statistics since 2017-01-08 18:35:37 ***
    Total inserts                                  592.00
    Total updates                                    0.00
    Total deletes                                    0.00
    Total discards                                   0.00
    Total operations                               592.00
    Total CDR conflicts                                     60.00
    CDR resolutions succeeded                               60.00
    CDR INSERTROWEXISTS conflicts                           60.00

*** Daily statistics since 2017-01-09 00:00:00 ***
    Total inserts                                  562.00
    Total updates                                    0.00
    Total deletes                                    0.00
    Total discards                                   0.00
    Total operations                               562.00
    Total CDR conflicts                                     30.00
    CDR resolutions succeeded                               30.00
    CDR INSERTROWEXISTS conflicts                           30.00

*** Hourly statistics since 2017-01-09 15:00:00 ***

    No database operations have been performed.

*** Latest statistics since 2017-01-08 18:35:37 ***
    Total inserts                                  592.00
    Total updates                                    0.00
    Total deletes                                    0.00
    Total discards                                   0.00
    Total operations                               592.00
    Total CDR conflicts                                     60.00
    CDR resolutions succeeded                               60.00
    CDR INSERTROWEXISTS conflicts                           60.00

End of Statistics.


GGSCI (edbcn2.oracle.com) 13>
```

Note: The current version of EDBC workload simulator design to perform DML INSERT conflict only for the POLICIES table.

Task 6: EDBC Virtualization

In order to minimize data conflict and greatly enhance database security, this section deploy Virtual Private Database (VPD) - Virtualization. VPD is an Oracle Database 12c EE security feature, and deployed using a set of supplied PL/SQL API. The motive is to

restrict users' access to their associated database(s). Figure 13 shows the design of VPD access control list (VPD-ACL).

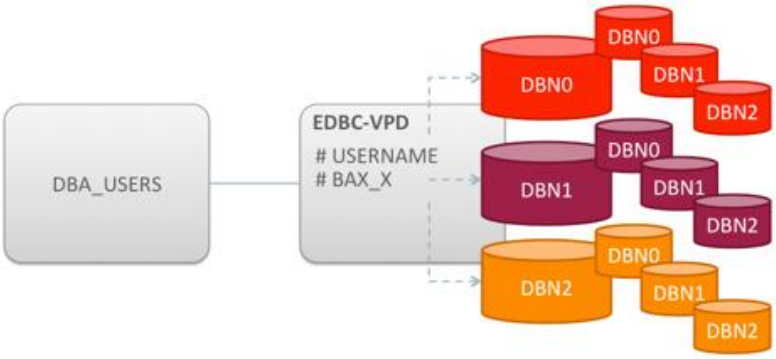


Figure 13: EDBC Virtualization

The steps to create Virtual Private Database (VPD) for the primary and extended nodes:

- Seed VPD metadata and create associated database users
- Grant access privileges
- Create the Security Policy
- Create the Context Management package
- Create the security function
- Create the database application context
- Run the applications

Refer to Appendix B for further details.

Seed VPD metadata and create associated database users

Create the metadata and database users on:

- The primary node
- Extended node 1
- Extended node 2

On the primary node

```
SQL> conn osm_repapi/oracle@dbn0.oracle.com
Connected.
SQL>
```

Create the metadata table – EDBC_VPD

```
SQL> create table edbc_vpd (username varchar2(128) not null,
  2                         bas_x    varchar2(32) not null);

Table created.

SQL> alter table edbc_vpd add constraint edbc_vpd_pk primary key
  2  (username, BAS_X);

Table altered.

SQL>
```

```
SQL> desc edbc_vpd
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USERNAME                                  NOT NULL VARCHAR2(128)
 BAS X                                     NOT NULL VARCHAR2(32)

SQL>
```

Seeding the metadata

```
SQL> insert into edbc_vpd
  2  values ('FALSWAIMIL0', 'DBN0');

1 row created.

SQL> insert into edbc_vpd
  2  values ('FALSWAIMIL1', 'DBN1');

1 row created.

SQL> insert into edbc_vpd
  2  values ('FALSWAIMIL2', 'DBN2');

1 row created.

SQL> insert into edbc_vpd
  2  values ('FALSWAIMIL012', 'DBN0');

1 row created.

SQL> insert into edbc_vpd
  2  values ('FALSWAIMIL012', 'DBN1');

1 row created.

SQL> insert into edbc_vpd
  2  values ('FALSWAIMIL012', 'DBN2');

1 row created.

SQL> commit;

Commit complete.

SQL>
```

Create and grant privileges to database users.

```
SQL> conn system/oracle_4U@dbn0.oracle.com
Connected.
SQL>
```

```
SQL> grant dba to FALSWAIMIL0 identified by oracle;

Grant succeeded.

SQL> conn FALSWAIMIL0/oracle@dbn0.oracle.com;
Connected.
SQL> grant dba to FALSWAIMIL1 identified by oracle;

Grant succeeded.

SQL> conn FALSWAIMIL1/oracle@dbn0.oracle.com;
Connected.
SQL> grant dba to FALSWAIMIL2 identified by oracle;

Grant succeeded.

SQL> conn FALSWAIMIL2/oracle@dbn0.oracle.com;
Connected.
SQL> grant dba to FALSWAIMIL012 identified by oracle;

Grant succeeded.

SQL> conn FALSWAIMIL012/oracle@dbn0.oracle.com;
Connected.
```

On extended node 1

```
SQL> conn osm_repapi/oracle@dbn1.oracle.com
Connected.
SQL>
```

Create the metadata table – EDBC_VPD

```
SQL> create table edbc_vpd (username varchar2(128) not null,
  2                         bas_x    varchar2(32) not null);

Table created.

SQL> alter table edbc_vpd add constraint edbc_vpd_pk primary key
  2  (username, BAS_X);

Table altered.

SQL>
```

```
SQL> desc edbc_vpd
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USERNAME                                  NOT NULL VARCHAR2(128)
 BAS_X                                     NOT NULL VARCHAR2(32)

SQL>
```

Seeding the metadata

```
SQL> insert into edbc_vpd
  2  values ('FALSWAIMIL0', 'DBN0');

1 row created.

SQL> insert into edbc_vpd
  2  values ('FALSWAIMIL1', 'DBN1');

1 row created.

SQL> insert into edbc_vpd
  2  values ('FALSWAIMIL2', 'DBN2');

1 row created.

SQL> insert into edbc_vpd
  2  values ('FALSWAIMIL012', 'DBN0');

1 row created.

SQL> insert into edbc_vpd
  2  values ('FALSWAIMIL012', 'DBN1');

1 row created.

SQL> insert into edbc_vpd
  2  values ('FALSWAIMIL012', 'DBN2');

1 row created.

SQL> commit;

Commit complete.

SQL>
```

Create and grant privileges to database users.

```
SQL> conn system/oracle_4U@dbn1.oracle.com
Connected.
SQL>
```

```
SQL> grant dba to FALSWAIMIL0 identified by oracle;

Grant succeeded.
```

```
SQL> conn FALSWAIMIL0/oracle@dbn1.oracle.com;
Connected.
SQL> grant dba to FALSWAIMIL1 identified by oracle;

Grant succeeded.

SQL> conn FALSWAIMIL1/oracle@dbn1.oracle.com;
Connected.
SQL> grant dba to FALSWAIMIL2 identified by oracle;

Grant succeeded.

SQL> conn FALSWAIMIL2/oracle@dbn1.oracle.com;
Connected.
SQL> grant dba to FALSWAIMIL012 identified by oracle;

Grant succeeded.

SQL>
```

On extended node 2

```
SQL> conn osm_repapi/oracle@dbn2.oracle.com
Connected.
SQL>
```

Create the metadata table – EDBC_VPD

```
SQL> create table edbc_vpd (username varchar2(128) not null,
  2                         bas_x    varchar2(32) not null);

Table created.

SQL> alter table edbc_vpd add constraint edbc_vpd_pk primary key
  2  (username, BAS_X);

Table altered.

SQL>
```

```
SQL> desc edbc_vpd
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 USERNAME                                  NOT NULL VARCHAR2(128)
 BAS_X                                     NOT NULL VARCHAR2(32)

SQL>
```

Seeding the metadata

```
SQL> insert into edbc_vpd
  2  values ('FALSWAIMIL0', 'DBN0');
```

```
1 row created.

SQL> insert into edbc_vpd
  2  values ('FALSWAIMIL1', 'DBN1');

1 row created.

SQL> insert into edbc_vpd
  2  values ('FALSWAIMIL2', 'DBN2');

1 row created.

SQL> insert into edbc_vpd
  2  values ('FALSWAIMIL012', 'DBN0');

1 row created.

SQL> insert into edbc_vpd
  2  values ('FALSWAIMIL012', 'DBN1');

1 row created.

SQL> insert into edbc_vpd
  2  values ('FALSWAIMIL012', 'DBN2');

1 row created.

SQL> commit;

Commit complete.

SQL>
```

Create and grant privileges to database users.

```
SQL> conn system/oracle_4U@dbn2.oracle.com
Connected.
SQL> grant dba to FALSWAIMIL0 identified by oracle;

Grant succeeded.

SQL> conn FALSWAIMIL0/oracle@dbn2.oracle.com
Connected.
SQL> grant dba to FALSWAIMIL1 identified by oracle;

Grant succeeded.

SQL> conn FALSWAIMIL1/oracle@dbn2.oracle.com
Connected.
SQL> grant dba to FALSWAIMIL2 identified by oracle;

Grant succeeded.

SQL> conn FALSWAIMIL2/oracle@dbn2.oracle.com
Connected.
SQL> grant dba to FALSWAIMIL012 identified by oracle;
```

```
Grant succeeded.

SQL> conn FALSWAIMIL012/oracle@dbn2.oracle.com
Connected.
SQL>
```

Grant VPD access privileges

On the primary node

```
SQL> conn / as sysdba
Connected.
SQL> grant exempt access policy to system;

Grant succeeded.

SQL> grant exempt access policy to ogg_admin;

Grant succeeded.

SQL> grant exempt access policy to osm_repapi;

Grant succeeded.

SQL> grant execute on dbms_rls to osm_repapi;

Grant succeeded.

SQL>
```

On extended node 1

```
SQL> conn / as sysdba
Connected.
SQL> grant exempt access policy to system;

Grant succeeded.

SQL> grant exempt access policy to ogg_admin;

Grant succeeded.

SQL> grant exempt access policy to osm_repapi;

Grant succeeded.

SQL> grant execute on dbms_rls to osm_repapi;

Grant succeeded.

SQL>
```

On extended node 2

```
SQL> conn / as sysdba
Connected.
SQL> grant exempt access policy to system;

Grant succeeded.

SQL> grant exempt access policy to ogg_admin;

Grant succeeded.

SQL> grant exempt access policy to osm_repapi;

Grant succeeded.

SQL> grant execute on dbms_rls to osm_repapi;

Grant succeeded.

SQL>
```

Create the Security Policy

On the primary node 1

```
SQL> conn / as sysdba
Connected.
SQL>
```

```
SQL> BEGIN
  2     dbms_rls.add_policy(
  3          object_schema=>'osm_repapi',
  4          object_name=>'customers',
  5          policy_name=>'cust_edbc_vpd',
  6          function_schema=>'system',
  7          policy_function=>'ogg_security_p.cust_profile',
  8          statement_types=>'select, insert, update, delete');
  9  END;
 10  /

PL/SQL procedure successfully completed.

SQL> BEGIN
  2     dbms_rls.add_policy(
  3        object_schema=>'osm_repapi',
  4        object_name=>'policies',
  5        policy_name=>'pol_edbc_vpd',
  6        function_schema=>'system',
  7        policy_function=>'ogg_security_p.pol_order',
  8        statement_types=>'select, insert, update, delete');
  9  END;
 10  /

PL/SQL procedure successfully completed.
```

```
SQL>
```

On extended node 1

```
SQL> conn / as sysdba
Connected.
SQL>
```

```
SQL> BEGIN
  2     dbms_rls.add_policy(
  3         object_schema=>'osm_repapi',
  4         object_name=>'customers',
  5         policy_name=>'cust_edbc_vpd',
  6         function_schema=>'system',
  7         policy_function=>'ogg_security_p.cust_profile',
  8         statement_types=>'select, insert, update, delete');
  9  END;
 10  /

PL/SQL procedure successfully completed.

SQL> BEGIN
  2     dbms_rls.add_policy(
  3        object_schema=>'osm_repapi',
  4        object_name=>'policies',
  5        policy_name=>'pol_edbc_vpd',
  6        function_schema=>'system',
  7        policy_function=>'ogg_security_p.pol_order',
  8        statement_types=>'select, insert, update, delete');
  9  END;
 10  /

PL/SQL procedure successfully completed.

SQL>
```

On extended node 2

```
SQL> conn / as sysdba
Connected.
SQL>
```

```
SQL> BEGIN
  2     dbms_rls.add_policy(
  3         object_schema=>'osm_repapi',
  4         object_name=>'customers',
  5         policy_name=>'cust_edbc_vpd',
  6         function_schema=>'system',
  7         policy_function=>'ogg_security_p.cust_profile',
  8         statement_types=>'select, insert, update, delete');
  9  END;
 10  /

PL/SQL procedure successfully completed.
```

```
SQL> BEGIN
  2     dbms_rls.add_policy(
  3        object_schema=>'osm_repapi',
  4        object_name=>'policies',
  5        policy_name=>'pol_edbc_vpd',
  6        function_schema=>'system',
  7        policy_function=>'ogg_security_p.pol_order',
  8        statement_types=>'select, insert, update, delete');
  9  END;
 10  /

PL/SQL procedure successfully completed.

SQL>
```

Create the Context Management Package

On the primary node

```
SQL> conn / as sysdba
Connected.
SQL>
```

```
SQL> create or replace package ogg_security is
  2       PROCEDURE set_user_bas_x(puser_id in varchar2);
  3  end;
  4  /

Package created.

SQL>
```

```
SQL> create or replace package body ogg_security is
  2      PROCEDURE set_user_bas_x(puser_id in varchar2) is
  3         vbas_x varchar2(32);
  4       BEGIN
  5          select bas_x into vbas_x
  6          from   osm_repapi.edbc_vpd
  7          where  upper(username) = upper(puser_id);
  8          dbms_session.set_context('osmapp', 'bas_x', vbas_x);
  9      EXCEPTION
 10          WHEN OTHERS THEN
 11               dbms_session.set_context('osmapp', 'bas_x', NULL);
 12      END;
 13  END;
 14  /

Package body created.

SQL>
```

On extended node 1

```
SQL> conn / as sysdba
Connected.
SQL>
```

```
SQL> create or replace package ogg_security is
  2       PROCEDURE set_user_bas_x(puser_id in varchar2);
  3  end;
  4  /

Package created.

SQL>
```

```
SQL> create or replace package body ogg_security is
  2      PROCEDURE set_user_bas_x(puser_id in varchar2) is
  3         vbas_x varchar2(32);
  4       BEGIN
  5          select bas_x into vbas_x
  6          from   osm_repapi.edbc_vpd
  7          where  upper(username) = upper(puser_id);
  8          dbms_session.set_context('osmapp', 'bas_x', vbas_x);
  9      EXCEPTION
 10          WHEN OTHERS THEN
 11               dbms_session.set_context('osmapp', 'bas_x', NULL);
 12      END;
 13  END;
 14  /

Package body created.

SQL>
```

On extended node 2

```
SQL> conn / as sysdba
Connected.
SQL>
```

```
SQL> create or replace package ogg_security is
  2       PROCEDURE set_user_bas_x(puser_id in varchar2);
  3  end;
  4  /

Package created.

SQL>
```

```
SQL> create or replace package body ogg_security is
  2      PROCEDURE set_user_bas_x(puser_id in varchar2) is
  3         vbas_x varchar2(32);
  4       BEGIN
  5          select bas_x into vbas_x
  6          from   osm_repapi.edbc_vpd
```

```
  7          where  upper(username) = upper(puser_id);
  8          dbms_session.set_context('osmapp', 'bas_x', vbas_x);
  9      EXCEPTION
 10          WHEN OTHERS THEN
 11               dbms_session.set_context('osmapp', 'bas_x', NULL);
 12      END;
 13  END;
 14  /

Package body created.

SQL>
```

Create the security function

On the primary node

```
SQL> conn system/oracle 4U
Connected.
SQL>
```

```
SQL> create or replace package ogg_security_p is
  2     function cust_profile (object_schema varchar2, object_name varchar2)
  3     return varchar2;
  4  end;
  5  /

Package created.

SQL>
```

```
SQL> create or replace package body ogg_security_p is
  2       function cust_profile (object_schema varchar2, object_name varchar2)
  3          return varchar2 is
  4       begin
  5          return 'bas_x=sys_context(''osmapp'', ''bas_x'')';
  6       end;
  7  end;
  8  /

Package body created.

SQL>
```

On extended node 1

```
SQL> conn system/oracle_4U
Connected.
SQL>
```

```
SQL> create or replace package ogg_security_p is
  2     function cust_profile (object_schema varchar2, object_name varchar2)
  3     return varchar2;
```

```
  4  end;
  5  /

Package created.

SQL>
```

```
SQL> create or replace package body ogg_security_p is
  2       function cust_profile (object_schema varchar2, object_name varchar2)
  3          return varchar2 is
  4       begin
  5          return 'bas_x=sys_context(''osmapp'', ''bas_x'')';
  6       end;
  7  end;
  8  /

Package body created.

SQL>
```

On extended node 2

```
SQL> conn system/oracle_4U
Connected.
SQL>
```

```
SQL> create or replace package ogg_security_p is
  2     function cust_profile (object_schema varchar2, object_name varchar2)
  3     return varchar2;
  4  end;
  5  /

Package created.

SQL>
```

```
SQL> create or replace package body ogg_security_p is
  2       function cust_profile (object_schema varchar2, object_name varchar2)
  3          return varchar2 is
  4       begin
  5          return 'bas_x=sys_context(''osmapp'', ''bas_x'')';
  6       end;
  7  end;
  8  /

Package body created.

SQL>
```

Create the database application context

On the primary node

```
SQL> create context osmapp using ogg_security;

Context created.

SQL>
```

On extended node 1

```
SQL> create context osmapp using ogg_security;

Context created.

SQL>
```

On extended node 2

```
SQL> create context osmapp using ogg_security;

Context created.

SQL>
```

Testing EDBC Data Virtualization

There will be basic tests performed on the primary node and extended nodes. Begin by creating the users, FALSWAIMIL0, FALSWAIMIL1, FALSWAIMIL2 and FALSWAIMIL012 across the cluster. The test is marked successful when:

1. Within the node, the user can select distinctly it's database only
2. Within the node, the user can't select other databases.

Grant execute on SYS.OGG_SECURITY, which activate the session for the connected user. Repeat the same test on each database, if the result is differ from the output displayed, then VPD is not working as intended.

On the primary node

```
SQL> conn / as sysdba
Connected.
SQL> grant execute on SYS.OGG_SECURITY to FALSWAIMIL0;

Grant succeeded.

SQL> grant execute on SYS.OGG_SECURITY to FALSWAIMIL1;

Grant succeeded.

SQL> grant execute on SYS.OGG_SECURITY to FALSWAIMIL2;
```

```
Grant succeeded.

SQL> grant execute on SYS.OGG_SECURITY to FALSWAIMIL012;

Grant succeeded.

SQL>
```

- Testing data segregation for FALSWAIMIL0

```
SQL> conn FALSWAIMIL0/oracle
Connected.
SQL> exec sys.ogg_security.SET_USER_BAS_X(user);

PL/SQL procedure successfully completed.

SQL> select distinct(bas_x) from osm_repapi.customers;

BAS X
--------------------------------
DBN0

SQL>
```

```
SQL> select distinct(bas_x) from osm_repapi.customers
  2  where bas_x in ('DBN1', 'DBN2');

no rows selected

SQL>
```

✓ DBN0 on query 1 and no rows selected on query 2, indicates passed the VPD test.

- Testing data segregation for FALSWAIMIL1

```
SQL> conn FALSWAIMIL1/oracle
Connected.
SQL> exec sys.ogg_security.SET_USER_BAS_X(user);

PL/SQL procedure successfully completed.

SQL> select distinct(bas_x) from osm_repapi.customers;

BAS_X
--------------------------------
DBN1

SQL>
```

```
SQL> select distinct(bas_x) from osm_repapi.customers
  2  where bas_x in ('DBN0', 'DBN2');

no rows selected
```

```
SQL>
```

✓DBN1 on query 1 and no rows selected on query 2, indicates passed the VPD test.

▸ Testing data segregation for FALSWAIMIL2

```
SQL> conn FALSWAIMIL2/oracle
Connected.
SQL> exec sys.ogg_security.SET_USER_BAS_X(user);

PL/SQL procedure successfully completed.

SQL> select distinct(bas_x) from osm_repapi.customers;

BAS_X
------------------------------
DBN2

SQL>
```

```
SQL> select distinct(bas_x) from osm_repapi.customers
  2  where bas_x in ('DBN0', 'DBN1');

no rows selected

SQL>
```

✓DBN2 on query 1 and no rows selected on query 2, indicates passed the VPD test.

💻*On extended node 1*

```
SQL> conn / as sysdba
Connected.
SQL> grant execute on SYS.OGG_SECURITY to FALSWAIMIL0;

Grant succeeded.

SQL> grant execute on SYS.OGG_SECURITY to FALSWAIMIL1;

Grant succeeded.

SQL> grant execute on SYS.OGG_SECURITY to FALSWAIMIL2;

Grant succeeded.

SQL> grant execute on SYS.OGG_SECURITY to FALSWAIMIL012;

Grant succeeded.

SQL>
```

▸ Testing data segregation for FALSWAIMIL0

```
SQL> conn FALSWAIMIL0/oracle
Connected.
SQL> exec sys.ogg_security.SET_USER_BAS_X(user);

PL/SQL procedure successfully completed.

SQL> select distinct(bas_x) from osm_repapi.customers;

BAS_X
--------------------------------
DBN0

SQL> select distinct(bas_x) from osm_repapi.customers
  2  where bas_x in ('DBN1', 'DBN2');

no rows selected

SQL>
```

✓DBN0 on query 1 and no rows selected on query 2, indicates passed the VPD test.

- Testing data segregation for FALSWAIMIL1

```
SQL> conn FALSWAIMIL1/oracle
Connected.
SQL> exec sys.ogg_security.SET_USER_BAS_X(user);

PL/SQL procedure successfully completed.

SQL> select distinct(bas_x) from osm_repapi.customers;

BAS_X
--------------------------------
DBN1

SQL> select distinct(bas_x) from osm_repapi.customers
  2  where bas_x in ('DBN0', 'DBN2');

no rows selected

SQL>
```

✓DBN1 on query 1 and no rows selected on query 2, indicates passed the VPD test.

- Testing data segregation for FALSWAIMIL2

```
SQL> conn FALSWAIMIL2/oracle
Connected.
SQL> exec sys.ogg_security.SET_USER_BAS_X(user);

PL/SQL procedure successfully completed.

SQL> select distinct(bas_x) from osm_repapi.customers;
```

```
BAS_X
--------------------------------
DBN2

SQL>
SQL> select distinct(bas_x) from osm_repapi.customers
  2  where bas_x in ('DBN0', 'DBN1');

no rows selected

SQL>
```

✓DBN2 on query 1 and no rows selected on query 2, indicates passed the VPD test.

On extended node 2

```
SQL> conn / as sysdba
Connected.
SQL>
```

```
SQL> grant execute on SYS.OGG_SECURITY to FALSWAIMIL0;

Grant succeeded.

SQL> grant execute on SYS.OGG_SECURITY to FALSWAIMIL1;

Grant succeeded.

SQL> grant execute on SYS.OGG_SECURITY to FALSWAIMIL2;

Grant succeeded.

SQL> grant execute on SYS.OGG_SECURITY to FALSWAIMIL012;

Grant succeeded.

SQL>
```

- Testing data segregation for FALSWAIMIL0

```
SQL> conn FALSWAIMIL0/oracle
Connected.
SQL> exec sys.ogg_security.SET_USER_BAS_X(user);

PL/SQL procedure successfully completed.

SQL> select distinct(bas_x) from osm_repapi.customers;

BAS_X
--------------------------------
DBN0
SQL>
SQL> select distinct(bas_x) from osm_repapi.customers
```

```
  2  where bas_x in ('DBN1', 'DBN2');

no rows selected

SQL>
```

✓DBN0 on query 1 and no rows selected on query 2, indicates passed the VPD test.

- Testing data segregation for FALSWAIMIL1

```
SQL>conn FALSWAIMIL1/oracle
Connected.
SQL> exec sys.ogg_security.SET_USER_BAS_X(user);

PL/SQL procedure successfully completed.

SQL> select distinct(bas_x) from osm_repapi.customers;

BAS_X
--------------------------------
DBN1

SQL> select distinct(bas_x) from osm_repapi.customers
  2  where bas_x in ('DBN0', 'DBN2');

no rows selected

SQL>
```

✓DBN1 on query 1 and no rows selected on query 2, indicates passed the VPD test.

- Testing data segregation for FALSWAIMIL2

```
SQL>conn FALSWAIMIL2/oracle
Connected.
SQL> exec sys.ogg_security.SET_USER_BAS_X(user);

PL/SQL procedure successfully completed.

SQL> select distinct(bas_x) from osm_repapi.customers;

BAS_X
--------------------------------
DBN2

SQL>

SQL> select distinct(bas_x) from osm_repapi.customers
  2  where bas_x in ('DBN0', 'DBN1');

no rows selected

SQL>
```

✓DBN2 on query 1 and no rows selected on query 2, indicates passed the VPD test.

Workshop 9: EDBC Administration

The major EDBC administration tasks are developed, deployed and customized depending on the compute capacity of the cluster.

Table 13 lists Workshop 9 tasks.

Task No	Task Description
1	Enable Remote Login
2	Start the Cluster
3	Enable Network File System (NFS)
4	Backup the Cluster
5	Stop the Cluster

Table 13: Workshop 9 Tasks

The EDBC administration tasks are installed and executed from the primary node. Figure 14 shows frequently executed tasks.

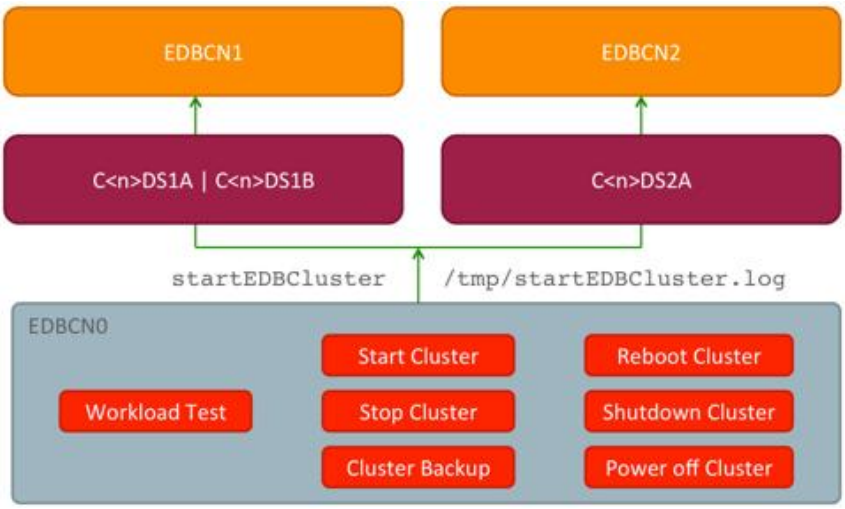


Figure 14: EDBC Centralized Administration

Task 1: Enable Remote Login

Suppressing password prompt allows the primary node to login to the extended node for running remote commands and programs, which is an optional option.

The steps to enable remote login are:

Step 1: Run ssh-keygen to generate the key.

This will prompt for a passphrase. Just press the enter key. It'll then generate an identification (private key) and a public key. Do not ever share the private key with anyone! ssh-keygen shows where it saved the public key. This is by default ~/.ssh/id_rsa.pub:

Your public key has been saved in /root/.ssh/id_rsa.pub

```
[root@edbcn0 .ssh]# ssh-keygen -t rsa
```

Step 2: Transfer the id_rsa.pub file to host_dest by either ftp, scp, rsync or any other method.

```
[root@edbcn0 .ssh]# scp id_rsa.pub root@edbcn1:/root
```

Step 3: Login as the remote user which you plan to use when you run scp, ssh or rsync on host_src.

Step 4: Copy the contents of id_rsa.pub to ~/.ssh/authorized_keys

```
[root@edbcn0 .ssh]# cat id_rsa.pub >>~/.ssh/authorized_keys
[root@edbcn0 .ssh]# chmod 700 ~/.ssh/authorized_keys
[root@edbcn0 .ssh]#
```

Repeat the above steps for extended node 1 (edbcn1.oracle.com), extended node 2 (edbcn2.oracle.com), downstream systems (c<n>ds1.oracle.com) and (c<n>ds2.oracle.com).

Task 2: Start the Cluster

From the administrative node (the primary node), start cluster nodes and downstream systems.

```
[root@edbcn0 ~]# ./startEDBCluster.sh
```

The following is the output of starting the EDBC infrastructure.

```
Starting Database C1DS1A...

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 27-DEC-2016 10:47:56

Copyright (c) 1991, 2014, Oracle.  All rights reserved.
```

```
Starting /u01/app/oracle/product/12.1.0/dbhome_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.2.0 - Production
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
Log messages written to
/u01/app/oracle/diag/tnslsnr/c1ds1/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=c1ds1.oracle.com)(PORT=1521)))

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=c1ds1.oracle.com)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.2.0 - Production
Start Date                27-DEC-2016 10:47:56
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/c1ds1/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=c1ds1.oracle.com)(PORT=1521)))
Services Summary...
Service "C1DS1A.oracle.com" has 1 instance(s).
  Instance "C1DS1A", status UNKNOWN, has 1 handler(s) for this service...
Service "C1DS1B.oracle.com" has 1 instance(s).
  Instance "C1DS1B", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully

SQL*Plus: Release 12.1.0.2.0 Production on Tue Dec 27 10:47:56 2016

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

SQL> Connected to an idle instance.
SQL> ORACLE instance started.

Total System Global Area          2516582400 bytes
Fixed Size                           2927528 bytes
Variable Size                     1560282200 bytes
Database Buffers                   939524096 bytes
Redo Buffers                        13848576 bytes
Database mounted.
Database opened.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0
- 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
Starting Database C1DS1B...

SQL*Plus: Release 12.1.0.2.0 Production on Tue Dec 27 10:48:06 2016

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

```
SQL> Connected to an idle instance.
SQL> ORACLE instance started.

Total System Global Area          2516582400 bytes
Fixed Size                           2927528 bytes
Variable Size                     1560282200 bytes
Database Buffers                   939524096 bytes
Redo Buffers                        13848576 bytes
Database mounted.
Database opened.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0
- 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
 [H [2JStarting Database C1DS2A...

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 27-DEC-2016 10:48:17

Copyright (c) 1991, 2014, Oracle.  All rights reserved.

Starting /u01/app/oracle/product/12.1.0/dbhome 1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.2.0 - Production
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
Log messages written to
/u01/app/oracle/diag/tnslsnr/c1ds2/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=c1ds2.oracle.com)(PORT=1521)))

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=c1ds2.oracle.com)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.2.0 - Production
Start Date                27-DEC-2016 10:48:17
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/c1ds2/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=c1ds2.oracle.com)(PORT=1521)))
Services Summary...
Service "C1DS2A" has 1 instance(s).
  Instance "C1DS2A", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully

SQL*Plus: Release 12.1.0.2.0 Production on Tue Dec 27 10:48:17 2016

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

SQL> Connected to an idle instance.
SQL> ORACLE instance started.
```

```
Total System Global Area          2516582400 bytes
Fixed Size                           2927528 bytes
Variable Size                     1560282200 bytes
Database Buffers                   939524096 bytes
Redo Buffers                        13848576 bytes
Database mounted.
Database opened.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0
- 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
Starting Database DBN1...

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 27-DEC-2016 10:48:08

Copyright (c) 1991, 2014, Oracle.  All rights reserved.

Starting /u01/app/oracle/product/12.1.0/dbhome_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.2.0 - Production
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
Log messages written to
/u01/app/oracle/diag/tnslsnr/edbcn1/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn1.oracle.com)(PORT=1521)))

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edbcn1.oracle.com)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.2.0 - Production
Start Date                27-DEC-2016 10:48:08
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome 1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/edbcn1/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn1.oracle.com)(PORT=1521)))
Services Summary...
Service "DBN1.oracle.com" has 1 instance(s).
  Instance "DBN1", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully

SQL*Plus: Release 12.1.0.2.0 Production on Tue Dec 27 10:48:08 2016

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

SQL> Connected to an idle instance.
SQL> ORACLE instance started.

Total System Global Area          2516582400 bytes
Fixed Size                           2927528 bytes
Variable Size                     1560282200 bytes
```

```
Database Buffers                    939524096 bytes
Redo Buffers                         13848576 bytes
Database mounted.
Database opened.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0
- 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
Starting Database DBN2...

LSNRCTL for Linux: Version 11.2.0.4.0 - Production on 27-DEC-2016 10:48:22

Copyright (c) 1991, 2013, Oracle.  All rights reserved.

Starting /u01/app/oracle/product/11.2.0/dbhome 1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 11.2.0.4.0 - Production
System parameter file is
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/listener.ora
Log messages written to
/u01/app/oracle/diag/tnslsnr/edbcn2/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn2.oracle.com)(PORT=1521)))
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edbcn2.oracle.com)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 11.2.0.4.0 - Production
Start Date                27-DEC-2016 10:48:22
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/11.2.0/dbhome 1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/edbcn2/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn2.oracle.com)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
Services Summary...
Service "DBN2.oracle.com" has 1 instance(s).
  Instance "DBN2", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully

SQL*Plus: Release 11.2.0.4.0 Production on Tue Dec 27 10:48:22 2016

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> Connected to an idle instance.
SQL> ORACLE instance started.

Total System Global Area          2516582400 bytes
Fixed Size                           2927528 bytes
Variable Size                     1560282200 bytes
Database Buffers                   939524096 bytes
```

```
Redo Buffers                          13848576 bytes
Database mounted.
Database opened.
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
Starting Database DBN0...

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 27-DEC-2016 10:48:35

Copyright (c) 1991, 2014, Oracle.  All rights reserved.

Starting /u01/app/oracle/product/12.1.0/dbhome_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.2.0 - Production
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
Log messages written to
/u01/app/oracle/diag/tnslsnr/edbcn0/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn0.oracle.com)(PORT=1521)))

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edbcn0.oracle.com)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.2.0 - Production
Start Date                27-DEC-2016 10:48:35
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/edbcn0/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn0.oracle.com)(PORT=1521)))
Services Summary...
Service "DBN0.oracle.com" has 1 instance(s).
  Instance "DBN0", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully

SQL*Plus: Release 12.1.0.2.0 Production on Tue Dec 27 10:48:35 2016

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

SQL> Connected to an idle instance.
SQL> ORACLE instance started.

Total System Global Area          2516582400 bytes
Fixed Size                           2927528 bytes
Variable Size                     1560282200 bytes
Database Buffers                   939524096 bytes
Redo Buffers                        13848576 bytes
Database mounted.
Database opened.
```

```
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0
- 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
Database Startup Completed...
Starting GoldenGate at EDBCN0...

Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.0.0 17185003 OGGCORE 12.1.2.0.0 PLATFORMS 130924.1316 FBO
Linux, x64, 64bit (optimized), Oracle 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (edbcn0.oracle.com) 1>
GGSCI (edbcn0.oracle.com) 2> start mgr

Manager started.


GGSCI (edbcn0.oracle.com) 3> start er *


Sending START request to MANAGER ...
REPLICAT RDBN101 starting

Sending START request to MANAGER ...
REPLICAT RDBN201 starting


GGSCI (edbcn0.oracle.com) 4>

GGSCI (edbcn0.oracle.com) 4> _[H_[2JStarting GoldenGate at EDBCN1...

Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.0.0 17185003 OGGCORE 12.1.2.0.0 PLATFORMS 130924.1316 FBO
Linux, x64, 64bit (optimized), Oracle 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (edbcn1.oracle.com) 1>
GGSCI (edbcn1.oracle.com) 2> start mgr

Manager started.


GGSCI (edbcn1.oracle.com) 3> start er *


Sending START request to MANAGER ...
REPLICAT RDBN001 starting

Sending START request to MANAGER ...
REPLICAT RDBN202 starting
```

```
GGSCI (edbcn1.oracle.com) 4>

GGSCI (edbcn1.oracle.com) 4> _[H_[2JStarting GoldenGate at EDBCN2...

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 (edbcn2.oracle.com) 1>
GGSCI (edbcn2.oracle.com) 2> start mgr

Manager started.


GGSCI (edbcn2.oracle.com) 3> start er *


Sending START request to MANAGER ...
REPLICAT RDBN002 starting

Sending START request to MANAGER ...
REPLICAT RDBN102 starting


GGSCI (edbcn2.oracle.com) 4>

GGSCI (edbcn2.oracle.com) 4>  [H [2JStarting GoldenGate at C1DS1A...

Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.0.0 17185003 OGGCORE_12.1.2.0.0_PLATFORMS_130924.1316_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (c1ds1.oracle.com) 1>
GGSCI (c1ds1.oracle.com) 2> start mgr

Manager started.


GGSCI (c1ds1.oracle.com) 3> start er *


Sending START request to MANAGER ...
EXTRACT EDBN001 starting

Sending START request to MANAGER ...
EXTRACT PDBN001 starting
```

```
Sending START request to MANAGER ...
EXTRACT PDBN002 starting


GGSCI (c1ds1.oracle.com) 4>

GGSCI (c1ds1.oracle.com) 4>
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.0.0 17185003 OGGCORE 12.1.2.0.0 PLATFORMS 130924.1316 FBO
Linux, x64, 64bit (optimized), Oracle 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (c1ds1.oracle.com) 1>
GGSCI (c1ds1.oracle.com) 2> start mgr

Manager started.


GGSCI (c1ds1.oracle.com) 3> start er *


Sending START request to MANAGER ...
EXTRACT EDBN101 starting

Sending START request to MANAGER ...
EXTRACT PDBN101 starting

Sending START request to MANAGER ...
EXTRACT PDBN102 starting


GGSCI (c1ds1.oracle.com) 4>

GGSCI (c1ds1.oracle.com) 4>  [H [2JStarting GoldenGate at C1DS1B...

Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.0.0 17185003 OGGCORE_12.1.2.0.0_PLATFORMS_130924.1316_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (c1ds1.oracle.com) 1>
GGSCI (c1ds1.oracle.com) 2> start mgr

MGR is already running.


GGSCI (c1ds1.oracle.com) 3> start er *

EXTRACT EDBN001 is already running.
EXTRACT PDBN001 is already running.
EXTRACT PDBN002 is already running.
```

```
GGSCI (c1ds1.oracle.com) 4>

GGSCI (c1ds1.oracle.com) 4>
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.0.0 17185003 OGGCORE_12.1.2.0.0_PLATFORMS_130924.1316_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (c1ds1.oracle.com) 1>
GGSCI (c1ds1.oracle.com) 2> start mgr

MGR is already running.


GGSCI (c1ds1.oracle.com) 3> start er *

EXTRACT EDBN101 is already running.
EXTRACT PDBN101 is already running.
EXTRACT PDBN102 is already running.


GGSCI (c1ds1.oracle.com) 4>

GGSCI (c1ds1.oracle.com) 4> _[H_[2JStarting GoldenGate at C1DS2A...

Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.0.0 17185003 OGGCORE_12.1.2.0.0_PLATFORMS_130924.1316_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (c1ds2.oracle.com) 1>
GGSCI (c1ds2.oracle.com) 2> start mgr

Manager started.


GGSCI (c1ds2.oracle.com) 3> start er *


Sending START request to MANAGER ...
EXTRACT EDBN201 starting

Sending START request to MANAGER ...
EXTRACT PDBN201 starting

Sending START request to MANAGER ...
EXTRACT PDBN202 starting
```

```
GGSCI (c1ds2.oracle.com) 4>

GGSCI (c1ds2.oracle.com) 4>
```

Task 3: Configure Network File System (NFS)

In order to enable RMAN writes to the primary node, the following steps configure Network File System (NFS) to allow the extended nodes associate a directory with a mount point on the primary node.

NFS Client Setup

```
[root@edbcn0 ~]# mkdir -p /home/oracle/BACKUP/dbf/DBN0
[root@edbcn0 ~]# mkdir -p /home/oracle/BACKUP/dbf/DBN1
[root@edbcn0 ~]# mkdir -p /home/oracle/BACKUP/dbf/DBN2
[root@edbcn0 ~]# mkdir -p /home/oracle/BACKUP/arc/DBN0
[root@edbcn0 ~]# mkdir -p /home/oracle/BACKUP/arc/DBN1
[root@edbcn0 ~]# mkdir -p /home/oracle/BACKUP/arc/DBN2
```

```
[root@edbcn0 ~]# cd /home/oracle
[root@edbcn0 oracle]# chown -R oracle:oinstall BACKUP
```

```
[root@edbcn0 oracle]# vi /etc/exports
/home/oracle/BACKUP/dbf/DBN1    edbcn1.oracle.com(rw,sync)
/home/oracle/BACKUP/arc/DBN1    edbcn1.oracle.com(rw,sync)
/home/oracle/BACKUP/dbf/DBN2    edbcn2.oracle.com(rw,sync)
/home/oracle/BACKUP/arc/DBN2    edbcn2.oracle.com(rw,sync)
```

```
[root@edbcn0 oracle]# /sbin/service nfs restart
Shutting down NFS mountd:                                  [  OK  ]
Shutting down NFS daemon:                                  [  OK  ]
Shutting down NFS quotas:                                  [  OK  ]
Shutting down NFS services:                                [  OK  ]
Starting NFS services:                                     [  OK  ]
Starting NFS quotas:                                       [  OK  ]
Starting NFS daemon:                                       [  OK  ]
Starting NFS mountd:                                       [  OK  ]
Stopping RPC idmapd:                                       [  OK  ]
Starting RPC idmapd:                                       [  OK  ]
[root@edbcn0 oracle]#
```

```
[root@edbcn0 oracle]# /usr/sbin/showmount -e edbcn0
Export list for edbcn0:
/home/oracle/BACKUP/arc/DBN2 edbcn2.oracle.com
/home/oracle/BACKUP/dbf/DBN2 edbcn2.oracle.com
/home/oracle/BACKUP/arc/DBN1 edbcn1.oracle.com
/home/oracle/BACKUP/dbf/DBN1 edbcn1.oracle.com
[root@edbcn0 oracle]#
```

NFS for Extended Node 1

```
[root@edbcn1 ~]# mkdir -p /home/oracle/BACKUP/dbf/DBN1
[root@edbcn1 ~]# mkdir -p /home/oracle/BACKUP/arc/DBN1
```

```
[root@edbcn1 oracle]# chown -R oracle:oinstall BACKUP
[root@edbcn1 oracle]#
```

```
[root@edbcn1 ~]# mount -t nfs edbcn0:/home/oracle/BACKUP/dbf/DBN1
/home/oracle/BACKUP/dbf/DBN1

[root@edbcn1 ~]# mount -t nfs edbcn0:/home/oracle/BACKUP/arc/DBN1
/home/oracle/BACKUP/arc/DBN
```

```
[root@edbcn1 oracle]# mount
/dev/mapper/VolGroup00-LogVol01 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/sda1 on /boot type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
nfs on edbcn1:/home/oracle/BACKUP/dbf/DBN1 type unknown (rw)
nfs on edbcn1:/home/oracle/BACKUP/arc/DBN1 type unknown (rw)
[root@edbcn1 oracle]#
```

NFS for Extended Node 2

```
[root@edbcn2 ~]# mkdir -p /home/oracle/BACKUP/dbf/DBN2
[root@edbcn2 ~]# mkdir -p /home/oracle/BACKUP/arc/DBN2
[root@edbcn2 ~]#
```

```
[root@edbcn2 ~]# cd /home/oracle
[root@edbcn2 oracle]# chown -R oracle:oinstall BACKUP
[root@edbcn2 oracle]#
```

```
[root@edbcn1 ~]# mount -t nfs edbcn0:/home/oracle/BACKUP/dbf/DBN2
/home/oracle/BACKUP/dbf/DBN2

[root@edbcn1 ~]# mount -t nfs edbcn0:/home/oracle/BACKUP/arc/DBN2
/home/oracle/BACKUP/arc/DBN2
```

```
[root@edbcn2 oracle]# mount
/dev/mapper/VolGroup00-LogVol01 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/sda1 on /boot type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
```

```
nfs on edbcn1:/home/oracle/BACKUP/dbf/DBN2 type unknown (rw)
nfs on edbcn1:/home/oracle/BACKUP/arc/DBN2 type unknown (rw)
[root@edbcn2 oracle]#
```

Test RMAN Backup

Run RMAN backup on extended node 1 to verify NFS is working.

On extended node 1

```
[oracle@edbcn1 ~]$ rman

Recovery Manager: Release 12.1.0.2.0 - Production on Tue Dec 27 18:23:03 2016

Copyright (c) 1982, 2014, Oracle and/or its affiliates.  All rights reserved.

RMAN> connect target

connected to target database: DBN1 (DBID=1383723292)

RMAN>
```

```
RMAN> BACKUP TABLESPACE users
2> FORMAT '/home/oracle/BACKUP/dbf/DBN1/users_%U.dbf';

Starting backup at 27-DEC-16
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=58 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00006
name=/u01/app/oracle/oradata/DBN1/datafile/o1_mf_users_d0mq8hv2_.dbf
channel ORA_DISK_1: starting piece 1 at 27-DEC-16
channel ORA_DISK_1: finished piece 1 at 27-DEC-16
piece handle=/home/oracle/BACKUP/dbf/DBN1/users_01rohof8_1_1.dbf
tag=TAG20161227T182648 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03
Finished backup at 27-DEC-16

RMAN>
```

Verify the backup sets are created successfully.

On the primary node

```
[oracle@edbcn0 DBN1]$ pwd
/home/oracle/BACKUP/dbf/DBN1
[oracle@edbcn0 DBN1]$ ls -al
total 31636
drwxr-xr-x 2 oracle oinstall     4096 Dec 27 18:32 .
drwxr-xr-x 5 oracle oinstall     4096 Dec 27 18:07 ..
-rw-r----- 1 oracle oinstall 32350208 Dec 27 18:32 users_02rohopp_1_1.dbf
```

```
[oracle@edbcn0 DBN1]$
```

On extended node 1

```
[oracle@edbcn2 ~]$ rman

Recovery Manager: Release 11.2.0.4.0 - Production on Tue Dec 27 19:09:57 2016

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

RMAN> connect target

connected to target database: DBN2 (DBID=1602423061)

RMAN>
```

```
RMAN> BACKUP TABLESPACE users
2> FORMAT '/home/oracle/BACKUP/dbf/DBN2/users_%U.dbf';

Starting backup at 27-DEC-16
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=39 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00004 name=/u01/app/oracle/oradata/DBN2/users01.dbf
channel ORA_DISK_1: starting piece 1 at 27-DEC-16
channel ORA_DISK_1: finished piece 1 at 27-DEC-16
piece handle=/home/oracle/BACKUP/dbf/DBN2/users 01rohr0o 1 1.dbf
tag=TAG20161227T191016 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:03
Finished backup at 27-DEC-16

RMAN>
```

Verify backup sets are created successfully.

On primary node

```
[oracle@edbcn0 DBN1]$ cd ../DBN2
[oracle@edbcn0 DBN2]$ ls -al
total 32612
drwxr-xr-x 2 oracle oinstall     4096 Dec 27 19:10 .
drwxr-xr-x 5 oracle oinstall     4096 Dec 27 18:07 ..
-rw-r----- 1 oracle oinstall 33349632 Dec 27 19:10 users_01rohr0o_1_1.dbf
[oracle@edbcn0 DBN2]$
```

Create RMAN Catalog

Creating RMAN catalog enables centralizing EDBC backup. The steps:

1. Create the catalog tablespace

2. Create the catalog database schema
3. Grant system privileges
4. Connect to catalog database
5. Create the catalog

On primary node

```
SQL> create tablespace edbc_rcv_cat
  2  datafile '/u01/app/oracle/oradata/DBN0/datafile/edbc_rcv_cat01.dbf' size
100M
  3  autoextend on;

Tablespace created.

SQL>
```

```
SQL> grant dba to edbc_rcv_cat_mgr identified by oracle;

Grant succeeded.

SQL> alter user edbc_rcv_cat_mgr
  2  quota unlimited on edbc_rcv_cat
  3  default tablespace edbc_rcv_cat
  4  profile default;

User altered.

SQL>
```

```
SQL> grant recovery_catalog_owner to edbc_rcv_cat_mgr;

Grant succeeded.

SQL>
```

```
[oracle@edbcn0 admin]$ rman

Recovery Manager: Release 12.1.0.2.0 - Production on Tue Dec 27 19:25:33 2016

Copyright (c) 1982, 2014, Oracle and/or its affiliates.  All rights reserved.

RMAN> connect target sys/oracle@DBN0.oracle.com

connected to target database: DBN0 (DBID=1344971290)
```

```
RMAN> connect catalog edbc_rcv_cat_mgr/oracle@DBN0.oracle.com

connected to recovery catalog database

RMAN>
```

```
RMAN> create catalog;
```

```
recovery catalog created

RMAN>
```

Register EDBC Databases

Register the primary node database

```
RMAN> register database;

database registered in recovery catalog
starting full resync of recovery catalog
full resync complete

RMAN> list incarnation of database;


List of Database Incarnations
DB Key  Inc Key DB Name  DB ID            STATUS  Reset SCN  Reset Time
------- ------- -------- ---------------- --- ---------- ----------
1       18      DBN0     1344971290       PARENT  1          07-JUL-14
1       2       DBN0     1344971290       CURRENT 1594143    21-OCT-16

RMAN>
```

Register extended node 1 database

```
[oracle@edbcn0 admin]$ rman

Recovery Manager: Release 12.1.0.2.0 - Production on Tue Dec 27 19:30:06 2016

Copyright (c) 1982, 2014, Oracle and/or its affiliates.  All rights reserved.

RMAN> connect target sys/oracle@DBN1.oracle.com

connected to target database: DBN1 (DBID=1383723292)

RMAN> connect catalog edbc_rcv_cat_mgr/oracle@DBN0.oracle.com

connected to recovery catalog database

RMAN> register database;

database registered in recovery catalog
starting full resync of recovery catalog
full resync complete

RMAN> list incarnation of database;


List of Database Incarnations
DB Key  Inc Key DB Name  DB ID            STATUS  Reset SCN  Reset Time
------- ------- -------- ---------------- --- ---------- ----------
188     204     DBN1     1383723292       PARENT  1          07-JUL-14
188     189     DBN1     1383723292       CURRENT 1594143    21-OCT-16
```

```
RMAN>
```

Register extended node 2 database

Notice, run RMAN from 11.2 client. Export TNS_ADMIN to point to Oracle Database 12c installed SQL*Net.

```
[oracle@edbcn0 bin]$ pwd
/u01/app/oracle/product/11.2.0/dbhome 1/bin
[oracle@edbcn0 bin]$ export
TNS_ADMIN=/u01/app/oracle/product/12.1.0/dbhome_1/network/admin
[oracle@edbcn0 bin]$ export ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
[oracle@edbcn0 bin]$ export
LD_LIBRARY_PATH=/u01/app/oracle/product/11.2.0/dbhome_1/lib
[oracle@edbcn0 bin]$ ./rman

Recovery Manager: Release 11.2.0.4.0 - Production on Tue Dec 27 20:01:35 2016

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

RMAN> connect target sys/oracle@DBN2.oracle.com

connected to target database: DBN2 (DBID=1602423061)

RMAN> connect catalog edbc_rcv_cat_mgr/oracle@DBN0.oracle.com

connected to recovery catalog database

RMAN>

RMAN> register database;

database registered in recovery catalog
starting full resync of recovery catalog
full resync complete

RMAN> list incarnation of database;


List of Database Incarnations
DB Key  Inc Key DB Name  DB ID            STATUS  Reset SCN  Reset Time
------- ------- -------- ---------------- --- ---------- ----------
387     405     DBN2     1602423061       PARENT  1          24-AUG-13
387     388     DBN2     1602423061       CURRENT 925702     25-NOV-16

RMAN>
```

Create EDBC Backup Stored Scripts

The primary node RMAN backup script

```
RMAN> connect target sys/oracle@dbn0.oracle.com
```

```
connected to target database: DBN0 (DBID=1344971290)

RMAN> connect catalog edbc_rcv_cat_mgr/oracle@DBN0.oracle.com

connected to recovery catalog database

RMAN> create script DBN0_BCK {
2> ALLOCATE CHANNEL c1 TYPE DISK;
3> ALLOCATE CHANNEL c2 TYPE DISK;
4> ALLOCATE CHANNEL c3 TYPE DISK;
5> BACKUP DATABASE INCLUDE CURRENT CONTROLFILE
6> FORMAT '/home/oracle/BACKUP/dbf/DBN0/DBF_%U.dbf';
7> SQL 'ALTER SYSTEM SWITCH LOGFILE';
8> BACKUP ARCHIVELOG ALL
9> FORMAT '/home/oracle/BACKUP/arc/DBN0/ARC_%U.dbf';
10> RELEASE CHANNEL c1;
11> RELEASE CHANNEL c2;
12> RELEASE CHANNEL c3;
13> }

created script DBN0 BCK

RMAN>
```

Extended node 1 RAMN backup script

```
RMAN> connect target sys/oracle@dbn1.oracle.com

connected to target database: DBN1 (DBID=1383723292)

RMAN> connect catalog edbc_rcv_cat_mgr/oracle@DBN0.oracle.com

connected to recovery catalog database

RMAN> create script DBN1_BCK {
2> ALLOCATE CHANNEL c1 TYPE DISK;
3> ALLOCATE CHANNEL c2 TYPE DISK;
4> ALLOCATE CHANNEL c3 TYPE DISK;
5> BACKUP DATABASE INCLUDE CURRENT CONTROLFILE
6> FORMAT '/home/oracle/BACKUP/dbf/DBN1/DBF_%U.dbf';
7> SQL 'ALTER SYSTEM SWITCH LOGFILE';
8> BACKUP ARCHIVELOG ALL
9> FORMAT '/home/oracle/BACKUP/arc/DBN1/ARC_%U.dbf';
10> RELEASE CHANNEL c1;
11> RELEASE CHANNEL c2;
12> RELEASE CHANNEL c3;
13> }

created script DBN1 BCK

RMAN>
```

Extended node 2 RMAN backup script

```
RMAN> create script DBN2_BCK {
2> ALLOCATE CHANNEL c1 TYPE DISK;
```

```
3> ALLOCATE CHANNEL c2 TYPE DISK;
4> ALLOCATE CHANNEL c3 TYPE DISK;
5> BACKUP DATABASE INCLUDE CURRENT CONTROLFILE
6> FORMAT '/home/oracle/BACKUP/dbf/DBN2/DBF_%U.dbf';
7> SQL 'ALTER SYSTEM SWITCH LOGFILE';
8> BACKUP ARCHIVELOG ALL
9> FORMAT '/home/oracle/BACKUP/arc/DBN2/ARC_%U.dbf';
10> RELEASE CHANNEL c1;
11> RELEASE CHANNEL c2;
12> RELEASE CHANNEL c3;
13> }

created script DBN2_BCK

RMAN>
```

Task 4: Backup the Cluster

From the administrative node (the primary node), backup the cluster using EDBC shell scripts – backupEDBC.sh. Figure 15 illustrates the backup architecture.

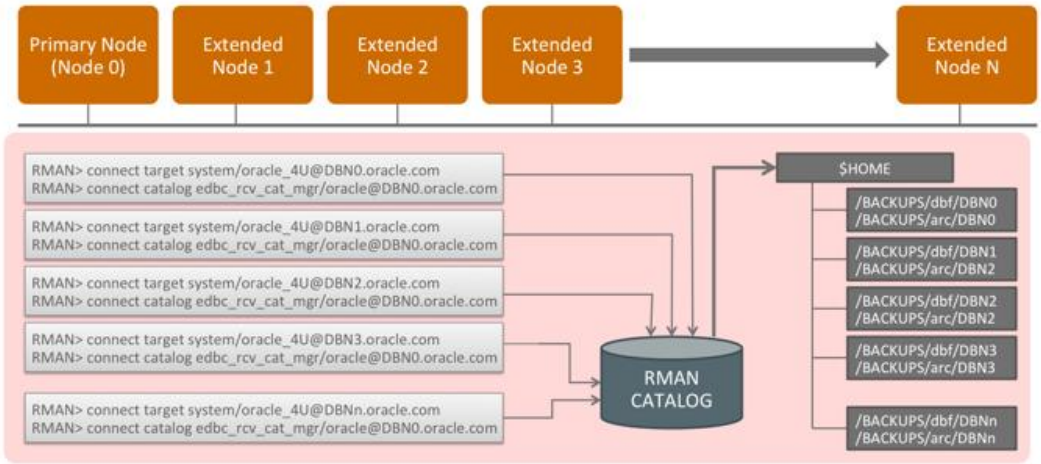


Figure 15: EDBC RMAN Backup Architecture

```
[root@edbcn0 ~]# cat backupEDBC.sh
#!/bin/sh
echo "Backing up Database DBN1..."
su - oracle /home/oracle/DBN1_BCKUP.sh
echo "Backing up Database DBN2..."
su - oracle /home/oracle/DBN2_BCKUP.sh
echo "Backing up Database DBN0..."
su - oracle /home/oracle/DBN0_BCKUP.sh

[root@edbcn0 ~]#
```

```
[root@edbcn0 oracle]# cat DBN0_BCKUP.sh
#!/bin/sh
export ORACLE SID=DBN0
export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
rman catalog edbc_rcv_cat_mgr/oracle@DBN0.oracle.com target
sys/oracle@DBN0.oracle.com @DBN0_BCKUP.rcv log=DBN0_BCKUP.log append

[root@edbcn0 oracle]#
```

```
[root@edbcn0 oracle]# cat DBN0_BCKUP.rcv
run {execute script DBN0 BCK;}
[root@edbcn0 oracle]#
```

Run the shell script to backup the primary node, extended node 1 and extended node 2.

```
[root@edbcn0 ~]# ./backupEDBC.sh
Backing up Database DBN1...
RMAN> 2>
```

```
[root@edbcn0 oracle]# ls -al *.log
-rw-r--r-- 1 oracle oinstall  1376 Dec 27 20:40 DBN0_BCKUP.log
-rw-r--r-- 1 oracle oinstall 12154 Dec 27 20:54 DBN1_BCKUP.log
-rw-r--r-- 1 oracle oinstall  3521 Dec 27 20:43 DBN2_BCKUP.log
[root@edbcn0 oracle]#
```

Backup logfile for extended node 1

```
Recovery Manager: Release 12.1.0.2.0 - Production on Tue Dec 27 20:49:43 2016

Copyright (c) 1982, 2014, Oracle and/or its affiliates.  All rights reserved.

connected to target database: DBN1 (DBID=1383723292)
connected to recovery catalog database

RMAN> run {execute script DBN1_BCK;}
2>
executing script: DBN1 BCK

allocated channel: c1
channel c1: SID=47 device type=DISK

allocated channel: c2
channel c2: SID=51 device type=DISK

allocated channel: c3
channel c3: SID=37 device type=DISK

Starting backup at 27-DEC-16
channel c1: starting full datafile backup set
channel c1: specifying datafile(s) in backup set
input datafile file number=00003
name=/u01/app/oracle/oradata/DBN1/datafile/o1_mf_sysaux_d0mq5ccn_.dbf
channel c1: starting piece 1 at 27-DEC-16
channel c2: starting full datafile backup set
channel c2: specifying datafile(s) in backup set
```

```
input datafile file number=00001
name=/u01/app/oracle/oradata/DBN1/datafile/o1_mf_system_d0mq6rnw_.dbf
input datafile file number=00006
name=/u01/app/oracle/oradata/DBN1/datafile/o1_mf_users_d0mq8hv2_.dbf
channel c2: starting piece 1 at 27-DEC-16
channel c3: starting full datafile backup set
channel c3: specifying datafile(s) in backup set
input datafile file number=00004
name=/u01/app/oracle/oradata/DBN1/datafile/o1 mf undotbs1 d0mq8jxv .dbf
input datafile file number=00005
name=/u01/app/oracle/oradata/DBN1/datafile/ogg dat01.dbf
channel c3: starting piece 1 at 27-DEC-16
channel c3: finished piece 1 at 27-DEC-16
piece handle=/home/oracle/BACKUP/dbf/DBN1/DBF_05roi0re_1_1.dbf
tag=TAG20161227T204949 comment=NONE
channel c3: backup set complete, elapsed time: 00:00:15
channel c3: starting full datafile backup set
channel c3: specifying datafile(s) in backup set
including current control file in backup set
channel c3: starting piece 1 at 27-DEC-16
channel c3: finished piece 1 at 27-DEC-16
piece handle=/home/oracle/BACKUP/dbf/DBN1/DBF_06roi0rt_1_1.dbf
tag=TAG20161227T204949 comment=NONE
channel c3: backup set complete, elapsed time: 00:00:16
channel c3: starting full datafile backup set
channel c3: specifying datafile(s) in backup set
including current SPFILE in backup set
channel c3: starting piece 1 at 27-DEC-16
channel c3: finished piece 1 at 27-DEC-16
piece handle=/home/oracle/BACKUP/dbf/DBN1/DBF_07roi0se_1_1.dbf
tag=TAG20161227T204949 comment=NONE
channel c3: backup set complete, elapsed time: 00:00:01
channel c1: finished piece 1 at 27-DEC-16
piece handle=/home/oracle/BACKUP/dbf/DBN1/DBF 03roi0re 1 1.dbf
tag=TAG20161227T204949 comment=NONE
channel c1: backup set complete, elapsed time: 00:02:17
channel c2: finished piece 1 at 27-DEC-16
piece handle=/home/oracle/BACKUP/dbf/DBN1/DBF 04roi0re 1 1.dbf
tag=TAG20161227T204949 comment=NONE
channel c2: backup set complete, elapsed time: 00:02:17
Finished backup at 27-DEC-16

sql statement: ALTER SYSTEM SWITCH LOGFILE

Starting backup at 27-DEC-16
current log archived
channel c1: starting archived log backup set
channel c1: specifying archived log(s) in backup set
input archived log thread=1 sequence=11 RECID=1 STAMP=930890146
input archived log thread=1 sequence=12 RECID=2 STAMP=930895270
input archived log thread=1 sequence=13 RECID=3 STAMP=930895348
input archived log thread=1 sequence=14 RECID=4 STAMP=930895393
input archived log thread=1 sequence=15 RECID=5 STAMP=930902406
input archived log thread=1 sequence=16 RECID=6 STAMP=930910023
input archived log thread=1 sequence=17 RECID=7 STAMP=930924054
input archived log thread=1 sequence=18 RECID=8 STAMP=930931208
input archived log thread=1 sequence=19 RECID=9 STAMP=930945637
input archived log thread=1 sequence=20 RECID=10 STAMP=930956460
input archived log thread=1 sequence=21 RECID=11 STAMP=930992418
```

```
input archived log thread=1 sequence=22 RECID=12 STAMP=931032044
input archived log thread=1 sequence=23 RECID=13 STAMP=931039611
input archived log thread=1 sequence=24 RECID=14 STAMP=931075501
input archived log thread=1 sequence=25 RECID=15 STAMP=931111235
input archived log thread=1 sequence=26 RECID=16 STAMP=931125631
input archived log thread=1 sequence=27 RECID=17 STAMP=931150859
input archived log thread=1 sequence=28 RECID=18 STAMP=931190423
input archived log thread=1 sequence=29 RECID=19 STAMP=931212025
channel c1: starting piece 1 at 27-DEC-16
channel c2: starting archived log backup set
channel c2: specifying archived log(s) in backup set
input archived log thread=1 sequence=30 RECID=20 STAMP=931230046
input archived log thread=1 sequence=31 RECID=21 STAMP=931266824
input archived log thread=1 sequence=32 RECID=22 STAMP=931282985
input archived log thread=1 sequence=33 RECID=23 STAMP=931283060
input archived log thread=1 sequence=34 RECID=26 STAMP=931283742
input archived log thread=1 sequence=35 RECID=28 STAMP=931298426
input archived log thread=1 sequence=36 RECID=30 STAMP=931299220
input archived log thread=1 sequence=37 RECID=32 STAMP=931323570
input archived log thread=1 sequence=38 RECID=34 STAMP=931346423
input archived log thread=1 sequence=39 RECID=36 STAMP=931370414
input archived log thread=1 sequence=40 RECID=38 STAMP=931384837
input archived log thread=1 sequence=41 RECID=40 STAMP=931388454
input archived log thread=1 sequence=42 RECID=42 STAMP=931413611
input archived log thread=1 sequence=43 RECID=44 STAMP=931420482
input archived log thread=1 sequence=44 RECID=46 STAMP=931428525
input archived log thread=1 sequence=45 RECID=48 STAMP=931442949
input archived log thread=1 sequence=46 RECID=50 STAMP=931453214
input archived log thread=1 sequence=47 RECID=52 STAMP=931460430
input archived log thread=1 sequence=48 RECID=54 STAMP=931471816
input archived log thread=1 sequence=49 RECID=56 STAMP=931489232
channel c2: starting piece 1 at 27-DEC-16
channel c3: starting archived log backup set
channel c3: specifying archived log(s) in backup set
input archived log thread=1 sequence=50 RECID=58 STAMP=931500030
input archived log thread=1 sequence=51 RECID=60 STAMP=931514491
input archived log thread=1 sequence=52 RECID=62 STAMP=931518031
input archived log thread=1 sequence=53 RECID=64 STAMP=931528934
input archived log thread=1 sequence=54 RECID=66 STAMP=931543355
input archived log thread=1 sequence=55 RECID=68 STAMP=931557792
input archived log thread=1 sequence=56 RECID=70 STAMP=931568422
input archived log thread=1 sequence=57 RECID=72 STAMP=931573363
input archived log thread=1 sequence=58 RECID=74 STAMP=931573364
input archived log thread=1 sequence=59 RECID=76 STAMP=931573373
input archived log thread=1 sequence=60 RECID=78 STAMP=931573388
input archived log thread=1 sequence=61 RECID=81 STAMP=931574170
input archived log thread=1 sequence=62 RECID=83 STAMP=931591475
input archived log thread=1 sequence=63 RECID=85 STAMP=931613732
input archived log thread=1 sequence=64 RECID=87 STAMP=931636854
input archived log thread=1 sequence=65 RECID=89 STAMP=931644051
input archived log thread=1 sequence=66 RECID=91 STAMP=931644082
input archived log thread=1 sequence=67 RECID=93 STAMP=931662053
input archived log thread=1 sequence=68 RECID=95 STAMP=931662980
input archived log thread=1 sequence=69 RECID=97 STAMP=931672437
input archived log thread=1 sequence=70 RECID=98 STAMP=931686758
input archived log thread=1 sequence=71 RECID=101 STAMP=931686813
input archived log thread=1 sequence=72 RECID=102 STAMP=931686939
input archived log thread=1 sequence=73 RECID=104 STAMP=931687177
input archived log thread=1 sequence=74 RECID=107 STAMP=931687216
```

```
input archived log thread=1 sequence=75 RECID=108 STAMP=931687590
input archived log thread=1 sequence=76 RECID=111 STAMP=931687832
input archived log thread=1 sequence=77 RECID=113 STAMP=931688113
channel c3: starting piece 1 at 27-DEC-16
channel c3: finished piece 1 at 27-DEC-16
piece handle=/home/oracle/BACKUP/arc/DBN1/ARC_0aroi0vs_1_1.dbf
tag=TAG20161227T205211 comment=NONE
channel c3: backup set complete, elapsed time: 00:01:55
channel c3: starting archived log backup set
channel c3: specifying archived log(s) in backup set
input archived log thread=1 sequence=78 RECID=114 STAMP=931688309
input archived log thread=1 sequence=79 RECID=117 STAMP=931688499
input archived log thread=1 sequence=80 RECID=119 STAMP=931689490
input archived log thread=1 sequence=81 RECID=120 STAMP=931689625
input archived log thread=1 sequence=82 RECID=123 STAMP=931689673
input archived log thread=1 sequence=83 RECID=125 STAMP=931689870
input archived log thread=1 sequence=84 RECID=126 STAMP=931690100
input archived log thread=1 sequence=85 RECID=129 STAMP=931690148
input archived log thread=1 sequence=86 RECID=131 STAMP=931690226
input archived log thread=1 sequence=87 RECID=132 STAMP=931690394
input archived log thread=1 sequence=88 RECID=135 STAMP=931690442
input archived log thread=1 sequence=89 RECID=137 STAMP=931691084
input archived log thread=1 sequence=90 RECID=138 STAMP=931717371
input archived log thread=1 sequence=91 RECID=139 STAMP=931726330
input archived log thread=1 sequence=92 RECID=140 STAMP=931726330
channel c3: starting piece 1 at 27-DEC-16
channel c2: finished piece 1 at 27-DEC-16
piece handle=/home/oracle/BACKUP/arc/DBN1/ARC 09roi0vr 1 1.dbf
tag=TAG20161227T205211 comment=NONE
channel c2: backup set complete, elapsed time: 00:02:10
channel c3: finished piece 1 at 27-DEC-16
piece handle=/home/oracle/BACKUP/arc/DBN1/ARC 0broi13f 1 1.dbf
tag=TAG20161227T205211 comment=NONE
channel c3: backup set complete, elapsed time: 00:00:15
channel c1: finished piece 1 at 27-DEC-16
piece handle=/home/oracle/BACKUP/arc/DBN1/ARC_08roi0vr_1_1.dbf
tag=TAG20161227T205211 comment=NONE
channel c1: backup set complete, elapsed time: 00:02:21
Finished backup at 27-DEC-16

released channel: c1

released channel: c2

released channel: c3

Recovery Manager complete.
[root@edbcn0 oracle]#
```

Task 5: Stop the Cluster

From the administrative node (primary node), stop the cluster.

```
[root@edbcn0 ~]# ./stopEDBCluster.sh
```

```
Stopping GoldenGate at c1ds1...

Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.0.0 17185003 OGGCORE_12.1.2.0.0_PLATFORMS_130924.1316_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (c1ds1.oracle.com) 1>
GGSCI (c1ds1.oracle.com) 2> stop er *


Sending STOP request to EXTRACT EDBN001 ...
Request processed.

Sending STOP request to EXTRACT PDBN001 ...
Request processed.

Sending STOP request to EXTRACT PDBN002 ...
Request processed.


GGSCI (c1ds1.oracle.com) 3> stop mgr!


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


GGSCI (c1ds1.oracle.com) 4>

GGSCI (c1ds1.oracle.com) 4>
Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.0.0 17185003 OGGCORE 12.1.2.0.0 PLATFORMS 130924.1316 FBO
Linux, x64, 64bit (optimized), Oracle 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (c1ds1.oracle.com) 1>
GGSCI (c1ds1.oracle.com) 2> stop er *


Sending STOP request to EXTRACT EDBN101 ...
Request processed.

Sending STOP request to EXTRACT PDBN101 ...
Request processed.

Sending STOP request to EXTRACT PDBN102 ...
Request processed.
```

```
GGSCI (c1ds1.oracle.com) 3> stop mgr!


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


GGSCI (c1ds1.oracle.com) 4>

GGSCI (c1ds1.oracle.com) 4>  [H [2JStopping GoldenGate at c1ds2...

Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.0.0 17185003 OGGCORE_12.1.2.0.0_PLATFORMS_130924.1316_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (c1ds2.oracle.com) 1>
GGSCI (c1ds2.oracle.com) 2> stop er *


Sending STOP request to EXTRACT EDBN201 ...
Request processed.

Sending STOP request to EXTRACT PDBN201 ...
Request processed.

Sending STOP request to EXTRACT PDBN202 ...
Request processed.


GGSCI (c1ds2.oracle.com) 3> stop mgr!


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


GGSCI (c1ds2.oracle.com) 4>

GGSCI (c1ds2.oracle.com) 4>  [H [2JStopping GoldenGate at edbcn0...

Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.0.0 17185003 OGGCORE 12.1.2.0.0 PLATFORMS 130924.1316 FBO
Linux, x64, 64bit (optimized), Oracle 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (edbcn0.oracle.com) 1>
GGSCI (edbcn0.oracle.com) 2> stop er *
```

```
Sending STOP request to REPLICAT RDBN101 ...
Request processed.

Sending STOP request to REPLICAT RDBN201 ...
Request processed.


GGSCI (edbcn0.oracle.com) 3> stop mgr!


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


GGSCI (edbcn0.oracle.com) 4>

GGSCI (edbcn0.oracle.com) 4> _[H_[2JStopping GoldenGate at edbcn1...

Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.0.0 17185003 OGGCORE_12.1.2.0.0_PLATFORMS_130924.1316_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.



GGSCI (edbcn1.oracle.com) 1>
GGSCI (edbcn1.oracle.com) 2> stop er *


Sending STOP request to REPLICAT RDBN001 ...
Request processed.

Sending STOP request to REPLICAT RDBN202 ...
Request processed.


GGSCI (edbcn1.oracle.com) 3> stop mgr!


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


GGSCI (edbcn1.oracle.com) 4>

GGSCI (edbcn1.oracle.com) 4>  [H [2JStopping GoldenGate at edbcn2...

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 (edbcn2.oracle.com) 1>
GGSCI (edbcn2.oracle.com) 2> stop er *


Sending STOP request to REPLICAT RDBN002 ...
Request processed.

Sending STOP request to REPLICAT RDBN102 ...
Request processed.


GGSCI (edbcn2.oracle.com) 3> stop mgr!


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


GGSCI (edbcn2.oracle.com) 4>

GGSCI (edbcn2.oracle.com) 4> _[H_[2JStopping Database DBN0...

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 27-DEC-2016 11:04:21

Copyright (c) 1991, 2014, Oracle.  All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edbcn0.oracle.com)(PORT=1521)))
The command completed successfully

SQL*Plus: Release 12.1.0.2.0 Production on Tue Dec 27 11:04:21 2016

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

SQL> Connected.
SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0
- 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
Stopping Database DBN1...

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 27-DEC-2016 11:04:39

Copyright (c) 1991, 2014, Oracle.  All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edbcn1.oracle.com)(PORT=1521)))
The command completed successfully

SQL*Plus: Release 12.1.0.2.0 Production on Tue Dec 27 11:04:39 2016

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

```
SQL> Connected.
SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0
- 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
Stopping Database DBN2...

LSNRCTL for Linux: Version 11.2.0.4.0 - Production on 27-DEC-2016 11:04:57

Copyright (c) 1991, 2013, Oracle.  All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edbcn2.oracle.com)(PORT=1521)))
The command completed successfully

SQL*Plus: Release 11.2.0.4.0 Production on Tue Dec 27 11:05:00 2016

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> Connected.
SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
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
Stopping Database C1DS1A...

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 27-DEC-2016 11:05:27

Copyright (c) 1991, 2014, Oracle.  All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=c1ds1.oracle.com)(PORT=1521)))
The command completed successfully

SQL*Plus: Release 12.1.0.2.0 Production on Tue Dec 27 11:05:27 2016

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

SQL> Connected.
SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0
- 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
Stopping Database C1DS1B...
/home/oracle/stopdbB.sh: line 1: [oracle@dss1: command not found

SQL*Plus: Release 12.1.0.2.0 Production on Tue Dec 27 11:05:45 2016

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

SQL> Connected.
```

```
SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0
- 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
Stopping Database C1DS2A...

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 27-DEC-2016 11:06:02

Copyright (c) 1991, 2014, Oracle.  All rights reserved.

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=c1ds2.oracle.com)(PORT=1521)))
The command completed successfully

SQL*Plus: Release 12.1.0.2.0 Production on Tue Dec 27 11:06:02 2016

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

SQL> Connected.
SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.2.0
- 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
Database Shutdown Completed...
```

Workshop 10: EDBC Zero-Downtime Upgrade

A flagship feature of Extended Database Cluster 12c (EDBC) is zero downtime database upgrade as online rollout-upgrade across the EDBC nodes. This workshop upgrades extended node 2 from Oracle Database 11g to Oracle Database 12c. Unlike other near zero upgrade techniques, the Extended Database Cluster 12c (EDBC) enables database upgrade with an absolute zero downtime. Refer to Appendix D for further details.

Table 14 lists Workshop 10 tasks.

Task No	Task Description
1	Divert the applications to run from extended node 1
2	Stop data streaming of primary and extended node 1 to extended node 2
3	Stop Replicats and verify no-record-found on the extended node 2
4	Report SCN, EXTSEQNO and EXTRBA of extended node 2 Replicats
5	Shutdown GoldenGate 12c on extended node 2 for Oracle database 11g
6	Backup trail files and parameter files
7	De-Install GoldenGate 12c on extended node 2 for Oracle database 11g
8	Install Oracle GoldenGate 12c on extended node 2 for Oracle Database 12c
9	Restore trail files and parameters files
10	Start Oracle GoldenGate 12c instance for Oracle Database 12c
11	Start data pumps of primary and extended node 1 to extended node 2
12	Verify remote trails are generated
13	Disable Redo Transport
14	Perform Oracle Database 12c pre-upgrade:
15	Execute Oracle Database 12c Upgrade Assistance
16	Perform Oracle Database 12c post-upgrade
17	Enable database flashback
18	Test applications
19	Rewind the database and disable database flashback
20	Enable Redo Transport
21	Re-Register extended node 2 Integrated Extract
22	Re-Create the checkpoint table
23	Re-define the Replicat on extended node 2
24	Start Replicats on the extended node 2
25	Verify data consistency
26	Divert the applications to run from extended node 2
27	Reconfigure RMAN catalog scripts
28	Reconfigure Administration scripts

Table 14: Workshop 10 Tasks

Task 1: Divert the Applications to Run from Extended Node 1

Divert applications run-time to an alternative node. The applications will remain running on extended node 1 throughout the upgrade of extended node 2.

On extended node 1

```
SQL> conn osm_repapi/oracle@dbn1.oracle.com
Connected.
SQL> exec call_trans_load;
```

Task 2: Stop Data Streaming of the Primary Node and Extended Node 1 to Extended Node 2

No further transactions sent to extended node 2. This establishes a precise apply SCN.

On downstream system 1 – C<n>DS1A

```
GGSCI (c1ds1.oracle.com) 19> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN001     00:00:03      00:00:01
EXTRACT     RUNNING     PDBN001     00:00:00      00:00:05
EXTRACT     RUNNING     PDBN002     00:00:00      00:00:05


GGSCI (c1ds1.oracle.com) 20> stop PDBN002

Sending STOP request to EXTRACT PDBN002 ...
Request processed.


GGSCI (c1ds1.oracle.com) 21> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN001     00:00:05      00:00:06
EXTRACT     RUNNING     PDBN001     00:00:00      00:00:00
EXTRACT     STOPPED     PDBN002     00:00:00      00:01:24


GGSCI (c1ds1.oracle.com) 22>
```

On downstream system 1 – C<n>DS1B

```
GGSCI (c1ds1.oracle.com) 16> info all
```

```
Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN101     00:00:02      00:00:02
EXTRACT     RUNNING     PDBN101     00:00:00      00:00:05
EXTRACT     RUNNING     PDBN102     00:00:00      00:00:05


GGSCI (c1ds1.oracle.com) 17> stop PDBN102

Sending STOP request to EXTRACT PDBN102 ...
Request processed.


GGSCI (c1ds1.oracle.com) 18> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN101     00:00:02      00:00:08
EXTRACT     RUNNING     PDBN101     00:00:00      00:00:01
EXTRACT     STOPPED     PDBN102     00:00:00      00:00:09


GGSCI (c1ds1.oracle.com) 19>
```

Task 3: Stop Replicats and Verify No-Record-Found on Extended Node 2

No further transactions applied by extended node 2. Issue the LAG command to verify EOF, on more records to process.

On extended node 2

```
GGSCI (edbcn2.oracle.com) 12> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN002     00:00:00      00:00:01
REPLICAT    RUNNING     RDBN102     00:00:00      00:00:01


GGSCI (edbcn2.oracle.com) 13> dblogin userid ogg_admin@dbn2.oracle.com, password
oracle
Successfully logged into database.

GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 14> lag RDBN002

Sending GETLAG request to REPLICAT RDBN002 ...
No records yet processed.
At EOF, no more records to process.
```

```
GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 15> lag RDBN102

Sending GETLAG request to REPLICAT RDBN102 ...
No records yet processed.
At EOF, no more records to process.


GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 16> stop *

Sending STOP request to REPLICAT RDBN002 ...
Request processed.

Sending STOP request to REPLICAT RDBN102 ...
Request processed.


GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 17> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    STOPPED     RDBN002     00:00:00      00:00:06
REPLICAT    STOPPED     RDBN102     00:00:00      00:00:06


GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 18>
```

Task 4: Mark the Current, CSN, EXTSEQNO and EXTRBA of Extended Node 2 Replicats

The CSN (commit sequence number) will be used by the Replicat after the upgrade successfully completed.

The last Committed Transaction CSN value will be used when starting the Replicats.

On extended node 2

```
GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 11> info replicat RDBN002, detail

REPLICAT   RDBN002   Last Started 2017-01-15 08:19   Status STOPPED
Checkpoint Lag       00:00:00 (updated 00:00:23 ago)
Log Read Checkpoint  File ./dirdat/tp000000078
                     2017-01-14 16:14:30.138042  RBA 2826270

Current Log BSN value: 7871941

Last Committed Transaction CSN value: 10408012

  Extract Source                          Begin             End

  ./dirdat/tp000000078                    2017-01-09 15:11  2017-01-14 16:14
  ./dirdat/tp000000076                    2017-01-08 17:52  2017-01-09 15:11
```

```
  ./dirdat/tp000000004                    2017-01-08 17:52  2017-01-08 17:52
  ./dirdat/tp000000004                    2017-01-08 17:50  2017-01-08 17:52
  ./dirdat/tp000000000                    * Initialized *   2017-01-08 17:50
  ./dirdat/tp000000002                    * Initialized *   2017-01-08 17:42
  ./dirdat/tp000000000                    * Initialized *   First Record


Current directory    /u01/app/oracle/product/12.2.1/gghome_1

Report file          /u01/app/oracle/product/12.2.1/gghome_1/dirrpt/RDBN002.rpt
Parameter file       /u01/app/oracle/product/12.2.1/gghome_1/dirprm/rdbn002.prm
Checkpoint file      /u01/app/oracle/product/12.2.1/gghome_1/dirchk/RDBN002.cpr
Checkpoint table     ogg_admin.chkpttab
Process file         /u01/app/oracle/product/12.2.1/gghome_1/dirpcs/RDBN002.pcr
Error log            /u01/app/oracle/product/12.2.1/gghome_1/ggserr.log


GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 12>
```

On extended node 2

```
GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 12> info replicat RDBN102, detail

REPLICAT   RDBN102   Last Started 2017-01-15 08:19   Status STOPPED
Checkpoint Lag       00:00:00 (updated 00:01:28 ago)
Log Read Checkpoint  File ./dirdat/t1000000068
                     2017-01-14 16:25:45.129923  RBA 11141674

Current Log BSN value: 8619163

Last Committed Transaction CSN value: 10415608

  Extract Source                          Begin             End

  ./dirdat/t1000000068                    2017-01-09 15:11  2017-01-14 16:25
  ./dirdat/t1000000066                    2017-01-08 17:52  2017-01-09 15:11
  ./dirdat/t1000000004                    2017-01-08 17:52  2017-01-08 17:52
  ./dirdat/t1000000004                    * Initialized *   2017-01-08 17:52
  ./dirdat/t1000000000                    * Initialized *   First Record


Current directory    /u01/app/oracle/product/12.2.1/gghome_1

Report file          /u01/app/oracle/product/12.2.1/gghome_1/dirrpt/RDBN102.rpt
Parameter file       /u01/app/oracle/product/12.2.1/gghome_1/dirprm/rdbn102.prm
Checkpoint file      /u01/app/oracle/product/12.2.1/gghome_1/dirchk/RDBN102.cpr
Checkpoint table     ogg admin.chkpttab
Process file         /u01/app/oracle/product/12.2.1/gghome_1/dirpcs/RDBN102.pcr
Error log            /u01/app/oracle/product/12.2.1/gghome_1/ggserr.log


GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 13>
```

Task 5: Shutdown Oracle GoldenGate 12c on Extended Node 2 for Oracle database 11*g*

In preparation to de-install, shutdown Oracle GoldenGate 12c on extended node 2.

On extended node 2

```
GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 18> info mgr

Manager is running (IP port edbcn2.oracle.com.7811, Process ID 4903).


GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 19> stop mgr!

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


GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 20> show

Parameter settings:

SET SUBDIRS    ON
SET DEBUG      OFF

Current directory: /u01/app/oracle/product/12.2.1/gghome_1

Using subdirectories for all process files

Editor:  vi

Reports (.rpt)                /u01/app/oracle/product/12.2.1/gghome_1/dirrpt
Parameters (.prm)             /u01/app/oracle/product/12.2.1/gghome 1/dirprm
Replicat Checkpoints (.cpr)   /u01/app/oracle/product/12.2.1/gghome_1/dirchk
Extract Checkpoints (.cpe)    /u01/app/oracle/product/12.2.1/gghome_1/dirchk
Process Status (.pcs)         /u01/app/oracle/product/12.2.1/gghome_1/dirpcs
SQL Scripts (.sql)            /u01/app/oracle/product/12.2.1/gghome 1/dirsql
Database Definitions (.def)   /u01/app/oracle/product/12.2.1/gghome_1/dirdef
Dump files (.dmp)             /u01/app/oracle/product/12.2.1/gghome 1/dirdmp
Masterkey wallet files (.wlt) /u01/app/oracle/product/12.2.1/gghome_1/dirwlt
Credential store files (.crd) /u01/app/oracle/product/12.2.1/gghome_1/dircrd


GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 21>
```

Task 6: Backup Trail Files and Parameter Files

Backup the trail files and parameters files to stage (temporary location).

On extended node 2

```
[oracle@edbcn2 gghome_1]$ mkdir -p /stage/dirdat
[oracle@edbcn2 gghome_1]$ mkdir -p /stage/dirprm
[oracle@edbcn2 gghome_1]$ cd dirdat
[oracle@edbcn2 dirdat]$ cp * /stage/dirdat
[oracle@edbcn2 dirdat]$ cd ../dirprm
[oracle@edbcn2 dirprm]$ cp * /stage/dirprm
[oracle@edbcn2 dirprm]$
```

Task 7: De-Install GoldenGate 12c on Extended Node 2 for Oracle database 11*g*

Run the deinstall utility. *Do not remove Oracle GoldenGate 12c installation manually.*

 *On extended node 2*

```
[oracle@edbcn2 ~]$ cd $OGG_HOME
[oracle@edbcn2 gghome_1]$ cd deinstall
[oracle@edbcn2 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/12.2.1/gghome_1 (yes/no)? [no]
yes
Starting Oracle Universal Installer...

Checking swap space: must be greater than 500 MB.   Actual 9983 MB    Passed
Preparing to launch Oracle Universal Installer from /tmp/OraInstall2017-01-
12 04-21-07PM. Please wait ...Oracle Universal Installer, Version 11.2.0.3.0
Production
Copyright (C) 1999, 2011, Oracle. All rights reserved.

Starting deinstall


Deinstall in progress (Thursday, January 12, 2017 4:21:19 PM AST)
............................................................ 100% Done.

Deinstall successful

End of install phases.(Thursday, January 12, 2017 4:21:43 PM AST)
End of deinstallations
Please check '/u01/app/oraInventory/logs/silentInstall2017-01-12_04-21-07PM.log'
for more details.
[oracle@edbcn2 deinstall]$
```

Task 8: Install Oracle GoldenGate 12c on Extended Node 2 for Oracle Database 12*c*

Install to the same location of Oracle GoldenGate 12c for Oracle Database 11g. There two options for installing Oracle GoldenGate 12c:

- Oracle GoldenGate 12c Installer on Silent mode
- Oracle GoldenGate 12c Installer on Desktop mode (GUI)

Oracle GoldenGate 12c Installer on Silent Mode

The installation is performed using the silent mode option. It requires preparation of the installer response file, then run the Oracle Universal Installer on silent mode using the shell script provided – runinstaller.sh saved at $HOME.

Prepare the installer response file – oggcore1.rsp

On extended node 2

```
[oracle@edbcn2 ~]$ cat oggcore1.rsp
oracle.install.responseFileVersion=rspfmt_ogginstall_response_schema_v12_2_0
INSTALL_OPTION=ORA12c
SOFTWARE_LOCATION=/u01/app/oracle/product/12.2.1/gghome_1
START_MANAGER=true
MANAGER_PORT=7811
DATABASE_LOCATION=/u01/app/oracle/product/12.1.0/dbhome_1
INVENTORY_LOCATION=/u01/app/oraInventory
UNIX GROUP NAME=oinstall

[oracle@edbcn2 ~]$
```

Make sure the DATABASE_LOCATION is matching the ORACLE_HOME.

```
[oracle@edbcn2 ~]$ export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
[oracle@edbcn2 ~]$ echo $ORACLE_HOME
/u01/app/oracle/product/12.1.0/dbhome 1
[oracle@edbcn2 ~]$
```

Run the installer on Silent mode. This is a customized installer for EDBC.

```
[oracle@edbcn2 ~]$ ./runInstall.sh

DISPLAY turned off Successfully

The Silent install option...
```

```
oracle.install.responseFileVersion=rspfmt_ogginstall_response_schema_v12_2_0
INSTALL_OPTION=ORA12c
SOFTWARE_LOCATION=/u01/app/oracle/product/12.2.1/gghome_1
START_MANAGER=true
MANAGER_PORT=7811
DATABASE_LOCATION=/u01/app/oracle/product/12.1.0/dbhome_1
INVENTORY_LOCATION=/u01/app/oraInventory
UNIX GROUP NAME=oinstall


/stage/ogg122/V100692-01/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. Press enter
Starting Oracle Universal Installer...

Checking Temp space: must be greater than 120 MB.   Actual 97612 MB    Passed
Checking swap space: must be greater than 150 MB.   Actual 9983 MB    Passed
Preparing to launch Oracle Universal Installer from /tmp/OraInstall2017-01-
15_09-13-51AM. Please wait ...You can find the log of this install session at:
 /u01/app/oraInventory/logs/installActions2017-01-15_09-13-51AM.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.

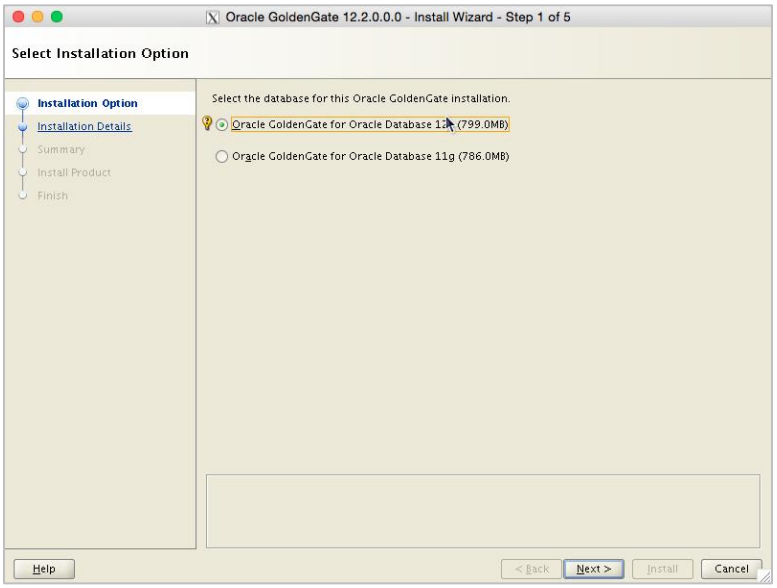
Setup files successful.
The installation of Oracle GoldenGate Core was successful.
Please check '/u01/app/oraInventory/logs/silentInstall2017-01-15_09-13-51AM.log'
for more details.

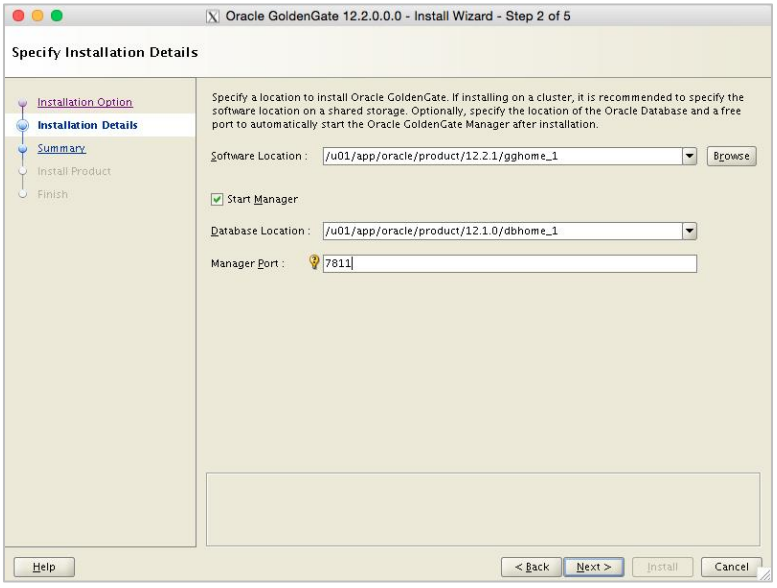
Configuring and starting manager... in progress.
.................................................   100% Done.

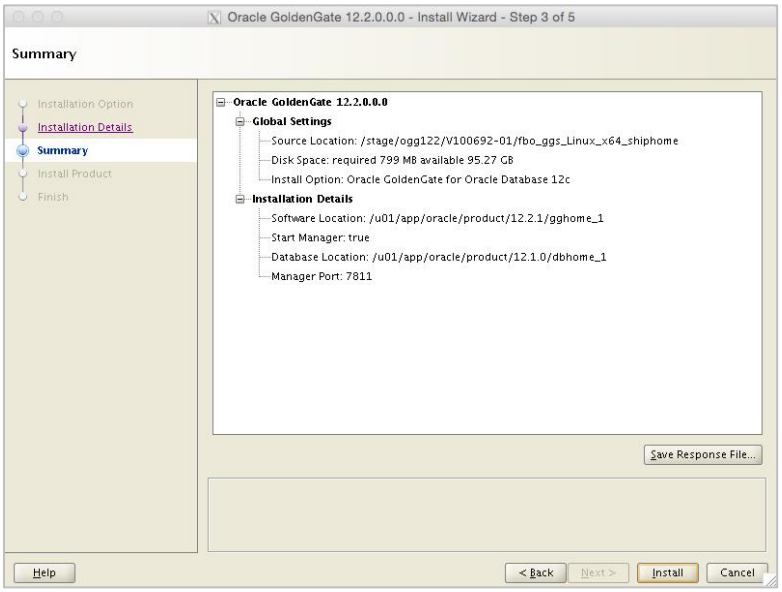
Configuring and starting manager... successful.
Successfully Setup Software.
[oracle@edbcn2 ~]$
```

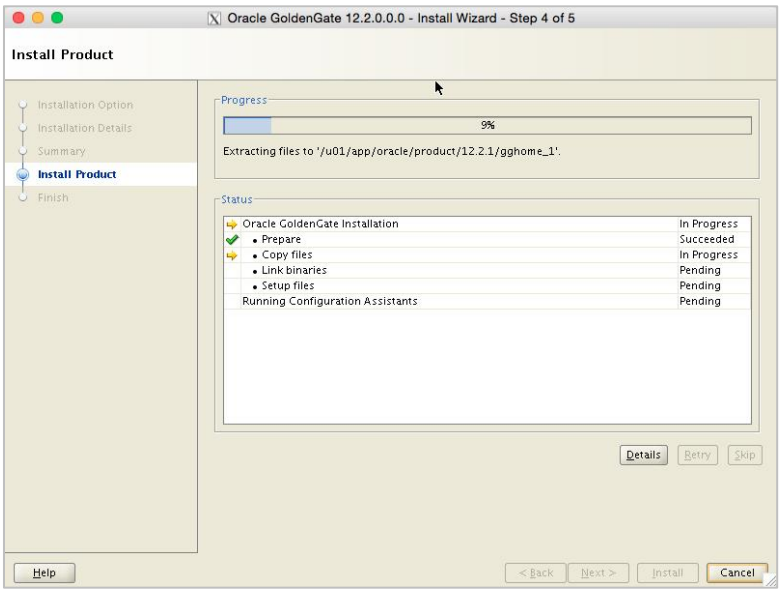
Oracle GoldenGate 12c Installer on Desktop mode (GUI)

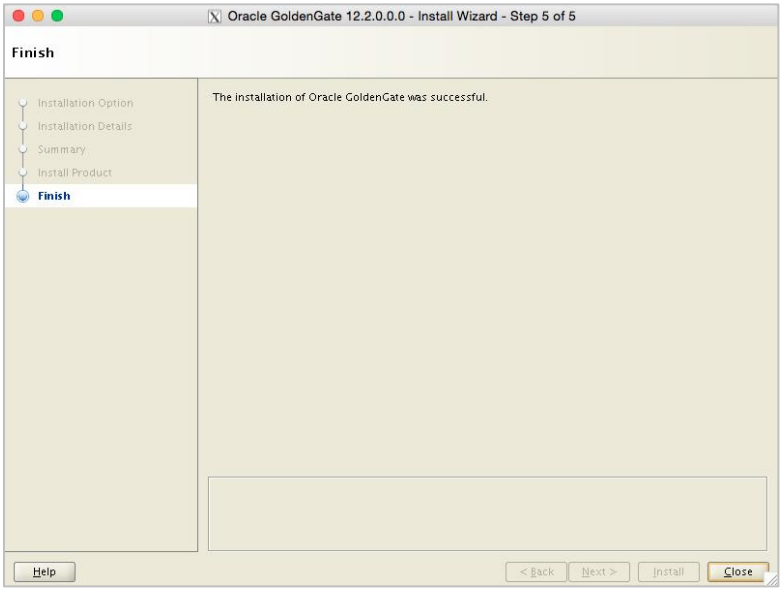
```
[oracle@edbcn2 Disk1]$ export DISPLAY=192.168.1.200:0.0
[oracle@edbcn2 Disk1]$ ./runInstaller
```






Oracle GoldenGate 12.2.0.0.0 - Install Wizard - Step 3 of 5
Summary
Installation Option
Installation Details
Summary
Install Product
Finish
Oracle GoldenGate 12.2.0.0.0
Global Settings
Source Location: /stage/ogg122/V100692-01/fbo_ggs_Linux_x64_shiphome
Disk Space: required 799 MB available 95.27 GB
Install Option: Oracle GoldenGate for Oracle Database 12c
Installation Details
Software Location: /u01/app/oracle/product/12.2.1/gghome_1
Start Manager: true
Database Location: /u01/app/oracle/product/12.1.0/dbhome_1
Manager Port: 7811
Save Response File...
Help
< Back
Next >
Install
Cancel


Oracle GoldenGate 12.2.0.0.0 - Install Wizard - Step 4 of 5
Install Product
Installation Option
Installation Details
Summary
Install Product
Finish
Progress
9%
Extracting files to '/u01/app/oracle/product/12.2.1/gghome_1'.
Status
Oracle GoldenGate Installation
In Progress
• Prepare
Succeeded
• Copy files
In Progress
• Link binaries
Pending
• Setup files
Pending
Running Configuration Assistants
Pending
Details
Retry
Skip
Help
< Back
Next >
Install
Cancel



On extended node 2

```
[oracle@edbcn2 ~]$ cd $OGG_HOME
[oracle@edbcn2 gghome_1]$ ./ggsci

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 UTF-8.

Copyright (C) 1995, 2015, Oracle and/or its affiliates. All rights reserved.



GGSCI (edbcn2.oracle.com) 1> show

Parameter settings:

SET SUBDIRS    ON
SET DEBUG      OFF

Current directory: /u01/app/oracle/product/12.2.1/gghome_1

Using subdirectories for all process files

Editor:  vi

Reports (.rpt)                 /u01/app/oracle/product/12.2.1/gghome_1/dirrpt
Parameters (.prm)              /u01/app/oracle/product/12.2.1/gghome_1/dirprm
Replicat Checkpoints (.cpr)    /u01/app/oracle/product/12.2.1/gghome_1/dirchk
Extract Checkpoints (.cpe)     /u01/app/oracle/product/12.2.1/gghome_1/dirchk
Process Status (.pcs)          /u01/app/oracle/product/12.2.1/gghome_1/dirpcs
```

```
SQL Scripts (.sql)             /u01/app/oracle/product/12.2.1/gghome_1/dirsql
Database Definitions (.def)    /u01/app/oracle/product/12.2.1/gghome_1/dirdef
Dump files (.dmp)              /u01/app/oracle/product/12.2.1/gghome_1/dirdmp
Masterkey wallet files (.wlt)  /u01/app/oracle/product/12.2.1/gghome_1/dirwlt
Credential store files (.crd)  /u01/app/oracle/product/12.2.1/gghome_1/dircrd


GGSCI (edbcn2.oracle.com) 2> info mgr

Manager is running (IP port edbcn2.oracle.com.7811, Process ID 5611).


GGSCI (edbcn2.oracle.com) 3> stop mgr!

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


GGSCI (edbcn2.oracle.com) 4>
```

Task 9: Restore Trail Files and Parameters Files

Restore the trail files and parameter files from stage.

On extended node 2

```
GGSCI (edbcn2.oracle.com) 6> exit
[oracle@edbcn2 gghome_1]$ cd dirdat
[oracle@edbcn2 dirdat]$ cp /stage/dir
dirdat/ dirprm/
[oracle@edbcn2 dirdat]$ cp /stage/dirdat/* .
[oracle@edbcn2 dirdat]$ ls
t1000000000  t1000000018  t1000000036  t1000000054  t1000000072  tp000000017
tp000000035  tp000000053   tp000000071
t1000000001  t1000000019  t1000000037  t1000000055  tp000000000  tp000000018
tp000000036  tp000000054   tp000000072
t1000000002  t1000000020  t1000000038  t1000000056  tp000000001  tp000000019
tp000000037  tp000000055   tp000000073
t1000000003  t1000000021  t1000000039  t1000000057  tp000000002  tp000000020
tp000000038  tp000000056   tp000000074
t1000000004  t1000000022  t1000000040  t1000000058  tp000000003  tp000000021
tp000000039  tp000000057   tp000000075
t1000000005  t1000000023  t1000000041  t1000000059  tp000000004  tp000000022
tp000000040  tp000000058   tp000000076
t1000000006  t1000000024  t1000000042  t1000000060  tp000000005  tp000000023
tp000000041  tp000000059   tp000000077
t1000000007  t1000000025  t1000000043  t1000000061  tp000000006  tp000000024
tp000000042  tp000000060   tp000000078
t1000000008  t1000000026  t1000000044  t1000000062  tp000000007  tp000000025
tp000000043  tp000000061   tp000000079
t1000000009  t1000000027  t1000000045  t1000000063  tp000000008  tp000000026
tp000000044  tp000000062   tp000000080
t1000000010  t1000000028  t1000000046  t1000000064  tp000000009  tp000000027
tp000000045  tp000000063   tp000000081
```

```
t1000000011  t1000000029  t1000000047  t1000000065  tp000000010  tp000000028
tp000000046  tp000000064   tp000000082
t1000000012  t1000000030  t1000000048  t1000000066  tp000000011  tp000000029
tp000000047  tp000000065
t1000000013  t1000000031  t1000000049  t1000000067  tp000000012  tp000000030
tp000000048  tp000000066
t1000000014  t1000000032  t1000000050  t1000000068  tp000000013  tp000000031
tp000000049  tp000000067
t1000000015  t1000000033  t1000000051  t1000000069  tp000000014  tp000000032
tp000000050  tp000000068
t1000000016  t1000000034  t1000000052  t1000000070  tp000000015  tp000000033
tp000000051  tp000000069
t1000000017  t1000000035  t1000000053  t1000000071  tp000000016  tp000000034
tp000000052  tp000000070
[oracle@edbcn2 dirdat]$ cd ../dirprm
[oracle@edbcn2 dirprm]$ cp /stage/dirprm/* .
[oracle@edbcn2 dirprm]$ ls
ggstart.oby  ggstop.oby  mgr.prm  rdbn002.prm  rdbn102.prm  rdbn201.prm
[oracle@edbcn2 dirprm]$
```

Task 10: Start Oracle GoldenGate 12c Instance for Oracle Database 12*c*

The purpose is to resume sending trail from the primary and expended node 1.

On extended node 2

```
[oracle@edbcn2 dirprm]$ cd $OGG_HOME
[oracle@edbcn2 gghome_1]$ ./ggsci

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 UTF-8.

Copyright (C) 1995, 2015, Oracle and/or its affiliates. All rights reserved.



GGSCI (edbcn2.oracle.com) 1> start mgr
Manager started.


GGSCI (edbcn2.oracle.com) 2> info mgr

Manager is running (IP port edbcn2.oracle.com.7811, Process ID 5693).


GGSCI (edbcn2.oracle.com) 3>
```

Task 11: Start the Data Pumps of the Primary and Extended Node 1 to Extended Node 2

Start the data pump processes to start sending the trail to extended node 2.

On downstream system – C1DS1A

```
GGSCI (c1ds1.oracle.com) 22> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN001     00:00:06      00:00:05
EXTRACT     RUNNING     PDBN001     00:00:00      00:00:06
EXTRACT     STOPPED     PDBN002     00:00:00      00:44:54


GGSCI (c1ds1.oracle.com) 23> start PDBN002

Sending START request to MANAGER ...
EXTRACT PDBN002 starting


GGSCI (c1ds1.oracle.com) 24> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN001     00:00:05      00:00:04
EXTRACT     RUNNING     PDBN001     00:00:00      00:00:06
EXTRACT     RUNNING     PDBN002     00:00:00      00:00:07


GGSCI (c1ds1.oracle.com) 25>
```

On downstream system – C1DS1B

```
GGSCI (c1ds1.oracle.com) 19> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN101     00:00:05      00:00:01
EXTRACT     RUNNING     PDBN101     00:00:00      00:00:09
EXTRACT     STOPPED     PDBN102     00:00:00      00:43:46


GGSCI (c1ds1.oracle.com) 20> start PDBN102

Sending START request to MANAGER ...
EXTRACT PDBN102 starting
```

```
GGSCI (c1ds1.oracle.com) 21> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN101     00:00:10      00:00:03
EXTRACT     RUNNING     PDBN101     00:00:00      00:00:01
EXTRACT     RUNNING     PDBN102     00:00:00      00:00:02


GGSCI (c1ds1.oracle.com) 22>
```

Task 12: Verify Remote Trails are Generated

The current trail file will be accumulating

On extended node 2

```
[oracle@edbcn2 dirdat]$
[oracle@edbcn2 dirdat]$ ls -ltr

-rw-r----- 1 oracle oinstall   253773 Jan 12 16:42 t1000000072
-rw-r----- 1 oracle oinstall  2589153 Jan 12 16:46 t1000000074
```

```
[oracle@edbcn2 dirdat]$
[oracle@edbcn2 dirdat]$ ls -ltr

-rw-r----- 1 oracle oinstall   253773 Jan 12 16:42 t1000000072
-rw-r----- 1 oracle oinstall  2633686 Jan 12 16:47 t1000000074
```

Task 13: Disable Redo-Transport

Setting the parameter log_archive_dest_state_2 to 'defer', temporary disable Redo-Transport to the downstream system.

On extended node 2

```
SQL> alter system set log_archive_dest_state_2='defer' scope=both;

System altered.


SQL> show parameters log_archive_dest_state_2

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
log_archive_dest_state_2             string      defer
log_archive_dest_state_20            string      enable
log_archive_dest_state_21            string      enable
log_archive_dest_state_22            string      enable
```

```
log_archive_dest_state_23       string     enable
log_archive_dest_state_24       string     enable
log_archive_dest_state_25       string     enable
log_archive_dest_state_26       string     enable
log_archive_dest_state_27       string     enable
log_archive_dest_state_28       string     enable
log_archive_dest_state_29       string     enable
SQL>
```

Task 14: Perform Oracle Database 12c Pre-Upgrade

Prior to upgrading the database, it's essential to perform the following pre-upgrade steps.

- Purge the database Recyclebin
- Validate invalid objects

On extended node 2

```
SQL> select *
  2  from   v$version;

BANNER
--------------------------------------------------------------------------------
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
PL/SQL Release 11.2.0.4.0 - Production
CORE 11.2.0.4.0     Production
TNS for Linux: Version 11.2.0.4.0 - Production
NLSRTL Version 11.2.0.4.0 - Production

SQL>
```

Purge the database recyclebin

```
SQL> conn / as sysdba
Connected.
SQL> purge dba_recyclebin;

DBA Recyclebin purged.

SQL>
```

Report invalid objects

```
SQL> SELECT DISTINCT object_name, object_type, owner
  2  FROM dba_objects
  3  WHERE status='INVALID'
  4  ORDER BY owner, object_name, object_type;

no rows selected

SQL>
```

Report obsolete parameters

```
SQL> SELECT name
  2  FROM v$parameter
  3  WHERE isdeprecated ='TRUE'
  4  ORDER BY name;

NAME
--------------------------------------------------------------------------------
active instance count
background_dump_dest
buffer pool keep
buffer_pool_recycle
commit_write
cursor_space_for_time
fast start io target
global_context_pool_size
instance groups
lock_name_space
log_archive_local_first

NAME
--------------------------------------------------------------------------------
log archive start
max_enabled_roles
parallel_automatic_tuning
parallel_io_cap_enabled
parallel server
parallel_server_instances
plsql debug
plsql_v2_compatibility
remote_os_authent
resource manager cpu allocation
serial_reuse

NAME
--------------------------------------------------------------------------------
sql_trace
standby archive dest
user_dump_dest

25 rows selected.

SQL>
```

Report installed components

```
SQL> select substr(comp_name, 1, 34) component, version, status
  2  FROM dba_registry
  3  where status='INVALID';

no rows selected

SQL>
```

Run pre-upgrade script. This step is mandatory when performing upgrade using DBUA silent mode.

On extended node 2

```
[oracle@edbcn2 ~]$ cd /u01/app/oracle/product/12.1.0/dbhome_1/rdbms/admin
[oracle@edbcn2 admin]$ /u01/app/oracle/product/12.1.0/dbhome_1/bin/sqlplus
/nolog

SQL*Plus: Release 12.1.0.2.0 Production on Fri Jan 13 02:05:53 2017

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

SQL> conn / as sysdba
Connected.
SQL> @preupgrd.sql







Loading Pre-Upgrade Package...


**************************************************************************
Executing Pre-Upgrade Checks in DBN2...
**************************************************************************


      ************************************************************

               ====>> ERRORS FOUND for DBN2 <<====

 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 DBN2 <<====

ACTIONS REQUIRED:

1. Review results of the pre-upgrade checks:
 /u01/app/oracle/cfgtoollogs/DBN2/preupgrade/preupgrade.log

2. Execute in the SOURCE environment BEFORE upgrade:
 /u01/app/oracle/cfgtoollogs/DBN2/preupgrade/preupgrade_fixups.sql

3. Execute in the NEW environment AFTER upgrade:
 /u01/app/oracle/cfgtoollogs/DBN2/preupgrade/postupgrade_fixups.sql
```

```
      ************************************************************

************************************************************************
Pre-Upgrade Checks in DBN2 Completed.
************************************************************************

************************************************************************
************************************************************************
SQL>
```

Review alert/log file

```
SQL> host more /u01/app/oracle/cfgtoollogs/DBN2/preupgrade/preupgrade.log
Oracle Database Pre-Upgrade Information Tool 01-13-2017 02:06:22
Script Version: 12.1.0.2.0 Build: 006
************************************************************************
   Database Name:  DBN2
  Container Name:  Not Applicable in Pre-12.1 database
    Container ID:  Not Applicable in Pre-12.1 database
         Version:  11.2.0.4.0
      Compatible:  11.2.0.4.0
       Blocksize:  8192
        Platform:  Linux x86 64-bit
   Timezone file:  V14
************************************************************************
                          [Update parameters]
         [Update Oracle Database 11.2.0.4.0 init.ora or spfile]

--> If Target Oracle is 32-bit, refer here for Update Parameters:
WARNING: --> "processes" needs to be increased to at least 300

--> If Target Oracle is 64-bit, refer here for Update Parameters:
WARNING: --> "processes" needs to be increased to at least 300
************************************************************************
************************************************************************
                        [Renamed Parameters]
                   [No Renamed Parameters in use]
************************************************************************
************************************************************************
                  [Obsolete/Deprecated Parameters]
            [No Obsolete or Desupported Parameters in use]
************************************************************************
                           [Component List]
************************************************************************
--> Oracle Catalog Views                    [upgrade]  VALID
--> Oracle Packages and Types               [upgrade]  VALID
--> JServer JAVA Virtual Machine            [upgrade]  VALID
--> Oracle XDK for Java                     [upgrade]  VALID
--> Oracle Workspace Manager                [upgrade]  VALID
--> OLAP Analytic Workspace                 [upgrade]  VALID
--> Oracle Enterprise Manager Repository    [upgrade]  VALID
--> Oracle Text                             [upgrade]  VALID
--> Oracle XML Database                     [upgrade]  VALID
--> Oracle Java Packages                    [upgrade]  VALID
--> Oracle Multimedia                       [upgrade]  VALID
--> Oracle Spatial                          [upgrade]  VALID
--> Expression Filter                       [upgrade]  VALID
```

```
--> Rule Manager                              [upgrade]  VALID
--> Oracle Application Express                [upgrade]  VALID
--> Oracle OLAP API                           [upgrade]  VALID
************************************************************************
                              [Tablespaces]
************************************************************************
--> SYSTEM tablespace is adequate for the upgrade.
     minimum required size: 1261 MB
--> SYSAUX tablespace is adequate for the upgrade.
     minimum required size: 1671 MB
--> UNDOTBS1 tablespace is adequate for the upgrade.
     minimum required size: 400 MB
--> TEMP tablespace is adequate for the upgrade.
     minimum required size: 60 MB
--> EXAMPLE tablespace is adequate for the upgrade.
     minimum required size: 310 MB

                       [No adjustments recommended]

************************************************************************
************************************************************************
                          [Pre-Upgrade Checks]
************************************************************************
WARNING: --> Process Count may be too low

     Database has a maximum process count of 150 which is lower than the
     default value of 300 for this release.
     You should update your processes value prior to the upgrade
     to a value of at least 300.
     For example:
        ALTER SYSTEM SET PROCESSES=300 SCOPE=SPFILE
     or update your init.ora file.

WARNING: --> Enterprise Manager Database Control repository found in the
database

     In Oracle Database 12c, Database Control is removed during
     the upgrade. To save time during the Upgrade, this action
     can be done prior to upgrading using the following steps after
     copying rdbms/admin/emremove.sql from the new Oracle home
   - Stop EM Database Control:
    $> emctl stop dbconsole

   - Connect to the Database using the SYS account AS SYSDBA:

   SET ECHO ON;
   SET SERVEROUTPUT ON;
   @emremove.sql
     Without the set echo and serveroutput commands you will not
     be able to follow the progress of the script.

INFORMATION: --> OLAP Catalog(AMD) exists in database

     Starting with Oracle Database 12c, OLAP Catalog component is desupported.
     If you are not using the OLAP Catalog component and want
     to remove it, then execute the
     ORACLE HOME/olap/admin/catnoamd.sql script before or
     after the upgrade.
```

```
INFORMATION: --> Older Timezone in use

     Database is using a time zone file older than version 18.
     After the upgrade, it is recommended that DBMS_DST package
     be used to upgrade the 11.2.0.4.0 database time zone version
     to the latest version which comes with the new release.
     Please refer to My Oracle Support note number 977512.1 for details.

INFORMATION: --> There are existing Oracle components that will NOT be
     upgraded by the database upgrade script.  Typically, such components
     have their own upgrade scripts, are deprecated, or obsolete.
     Those components are:  OLAP Catalog,OWB

INFORMATION: --> Oracle Application Express (APEX) can be
     manually upgraded prior to database upgrade

     APEX is currently at version 3.2.1.00.12 and will need to be
     upgraded to APEX version 4.2.5 in the new release.
     Note 1: To reduce database upgrade time, APEX can be manually
             upgraded outside of and prior to database upgrade.
     Note 2: See MOS Note 1088970.1 for information on APEX
             installation upgrades.


**************************************************************************
                    [Pre-Upgrade Recommendations]
**************************************************************************

                       ****************************************
                       ********* Dictionary Statistics *********
                       ****************************************

Please gather dictionary statistics 24 hours prior to
upgrading the database.
To gather dictionary statistics execute the following command
while connected as SYSDBA:
    EXECUTE dbms_stats.gather_dictionary_stats;

^^^ MANUAL ACTION SUGGESTED ^^^


**************************************************************************
                   [Post-Upgrade Recommendations]
**************************************************************************

                       ****************************************
                       ******** Fixed Object Statistics ********
                       ****************************************

Please create stats on fixed objects two weeks
after the upgrade using the command:
   EXECUTE DBMS_STATS.GATHER_FIXED_OBJECTS_STATS;

^^^ MANUAL ACTION SUGGESTED ^^^

**************************************************************************
                  ************  Summary  ************

 0 ERRORS exist in your database.
```

```
 2 WARNINGS that Oracle suggests are addressed to improve database performance.
 4 INFORMATIONAL messages that should be reviewed prior to your upgrade.

 After your database is upgraded and open in normal mode you must run
 rdbms/admin/catuppst.sql which executes several required tasks and completes
 the upgrade process.

 You should follow that with the execution of rdbms/admin/utlrp.sql, and a
 comparison of invalid objects before and after the upgrade using
 rdbms/admin/utluiobj.sql

 If needed you may want to upgrade your timezone data using the process
 described in My Oracle Support note 1509653.1
                  **********************************

SQL>
```

You must apply database recommended settings before starting the upgrade.

Change the processes parameter. It's static parameter, hence set the scope to spfile and rebounce the database.

On extended node 2

```
SQL> alter system set processes=500 scope=spfile;

System altered.

SQL>
```

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area   1653518336 bytes
Fixed Size                    2253784 bytes
Variable Size              1056967720 bytes
Database Buffers            587202560 bytes
Redo Buffers                  7094272 bytes
Database mounted.
Database opened.
SQL>
```

Resize the fast recovery area (FRA) to 10G.

```
SQL> alter system set db_recovery_file_dest_size=10000M scope=both;

System altered.

SQL> show parameters db_recovery_file_dest_size
```

```
NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
db_recovery_file_dest_size           big integer 10000M
SQL>
```

Gather data dictionary statistic.

```
SQL> EXECUTE dbms_stats.gather_dictionary_stats;
SQL>
```

Task 15: Execute Oracle Database 12c Upgrade Assistance

There two options for performing Oracle Database 12c upgrade using DBUA. The timeframe to complete the upgrade phases is approximately 90-120 minutes. The following illustrates DBUA on desktop mode.

- DBUA on silent mode
- DBUA on desktop mode (GUI)

DBUA on silent mode

```
[oracle@edbcn2 admin]$ /u01/app/oracle/product/12.1.0/dbhome_1/bin/dbua -silent
-sid DBN2 \
> -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/DBN2/upgrade1
Performing Pre Upgrade
4% complete
20% complete
Performing RDBMS Upgrade
20% complete
20% complete
21% complete
21% complete
22% complete
22% complete
22% complete
23% complete
23% complete
24% complete
24% complete
24% complete
25% complete
25% complete
```

```
26% complete
26% complete
26% complete
27% complete
27% complete
28% complete
28% complete
28% complete
29% complete
29% complete
30% complete
30% complete
30% complete
31% complete
31% complete
32% complete
32% complete
32% complete
33% complete
33% complete
34% complete
34% complete
34% complete
35% complete
35% complete
36% complete
36% complete
36% complete
37% complete
37% complete
38% complete
38% complete
38% complete
39% complete
39% complete
40% complete
Performing Post Upgrade
42% complete
44% complete
```

Monitor the upgrade by running the 'tail' command:

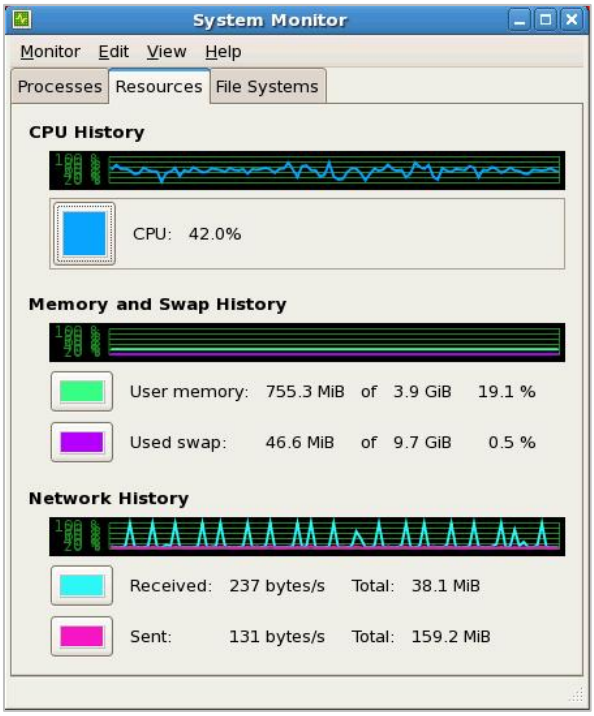
```
[root@edbcn2 ~]# cd /u01/app/oracle/cfgtoollogs/dbua/DBN2/upgrade1
[root@edbcn2 upgrade1]# ls -ltr
total 33656
-rw-r----- 1 oracle oinstall        0 Jan 13 02:19 trace.log.lck
-rw-r----- 1 oracle oinstall      100 Jan 13 02:19 preupgdir.sql
-rw-r----- 1 oracle oinstall     4977 Jan 13 02:19 upgrade.xml
-rw-r----- 1 oracle oinstall     2696 Jan 13 02:19 PreUpgradeResults.html
-rw-r----- 1 oracle oinstall      788 Jan 13 02:19 PreUpgrade.log
-rw-r----- 1 oracle oinstall        0 Jan 13 02:19 Oracle_Text.log
-rw-r----- 1 oracle oinstall    60977 Jan 13 02:20 sqls.log
-rw-r----- 1 oracle oinstall      442 Jan 13 02:20 catupgrd catcon 9579.lst
-rw-r----- 1 oracle oinstall   148697 Jan 13 02:20 trace.log
-rw-r----- 1 oracle oinstall     3181 Jan 13 02:30 Oracle Server.log
-rw-r----- 1 oracle oinstall       42 Jan 13 02:30 catupgrd_catcon_11932.done
-rw-r----- 1 oracle oinstall       42 Jan 13 02:30 catupgrd_catcon_11931.done
```

```
-rw-r----- 1 oracle oinstall       42 Jan 13 02:30 catupgrd_catcon_11930.done
-rw-r----- 1 oracle oinstall  5116130 Jan 13 02:30 catupgrd3.log
-rw-r----- 1 oracle oinstall  5706788 Jan 13 02:30 catupgrd2.log
-rw-r----- 1 oracle oinstall  5843523 Jan 13 02:30 catupgrd1.log
-rw-r----- 1 oracle oinstall 17462852 Jan 13 02:31 catupgrd0.log
[root@edbcn2 upgrade1]#
```

View the current log files:

```
[root@edbcn2 upgrade1]# tail -f catupgrd3.log
```

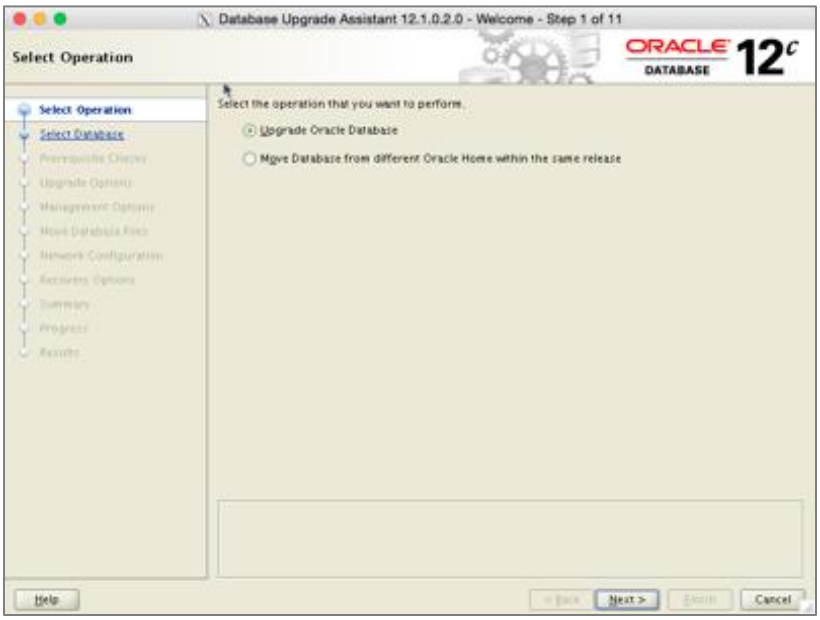
Monitor the operating system performance for CPU, memory and swap space usage

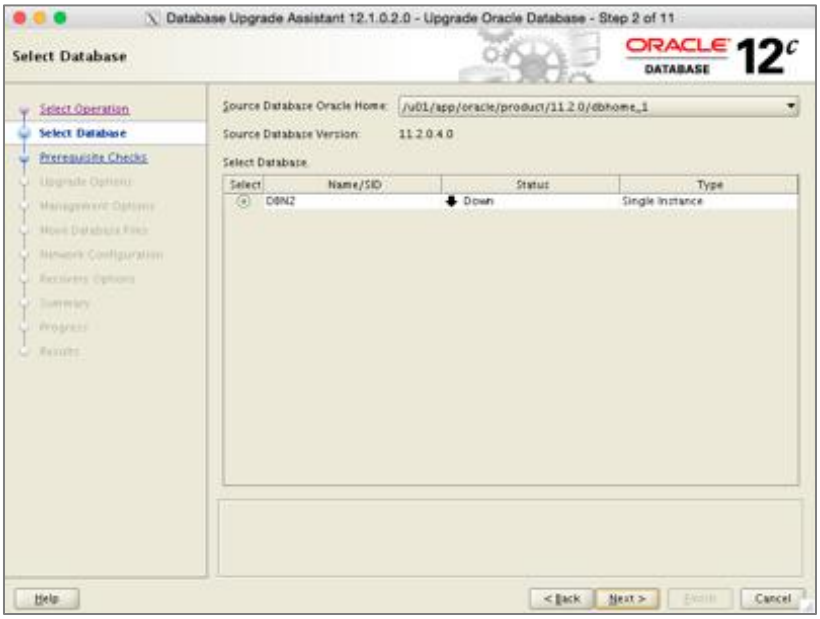


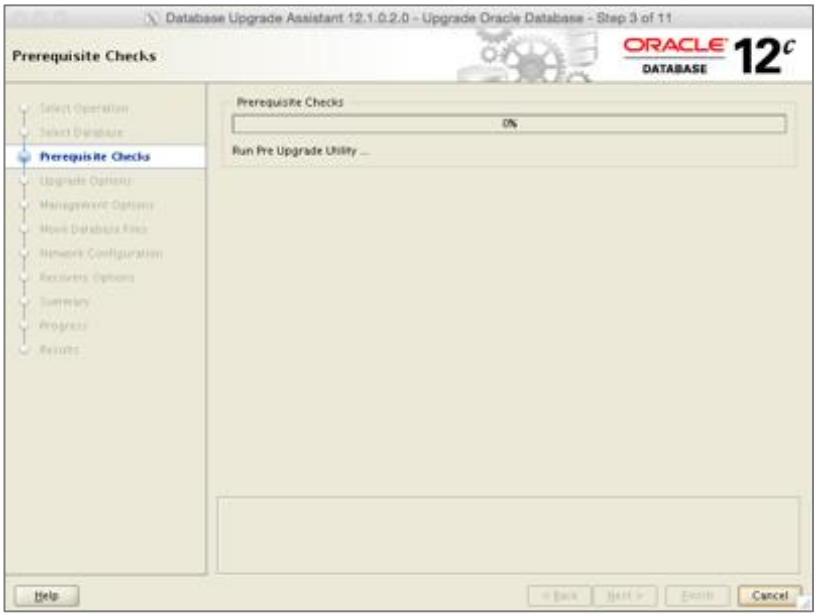

```
60% complete
Upgrading Timezone
80% complete
Generating Summary
Database upgrade has been completed successfully, and the database is ready to
use.
100% complete
Check the log file
"/u01/app/oracle/cfgtoollogs/dbua/logs/silent.log_1484462468081" for upgrade
details.
[oracle@edbcn2 admin]$
```

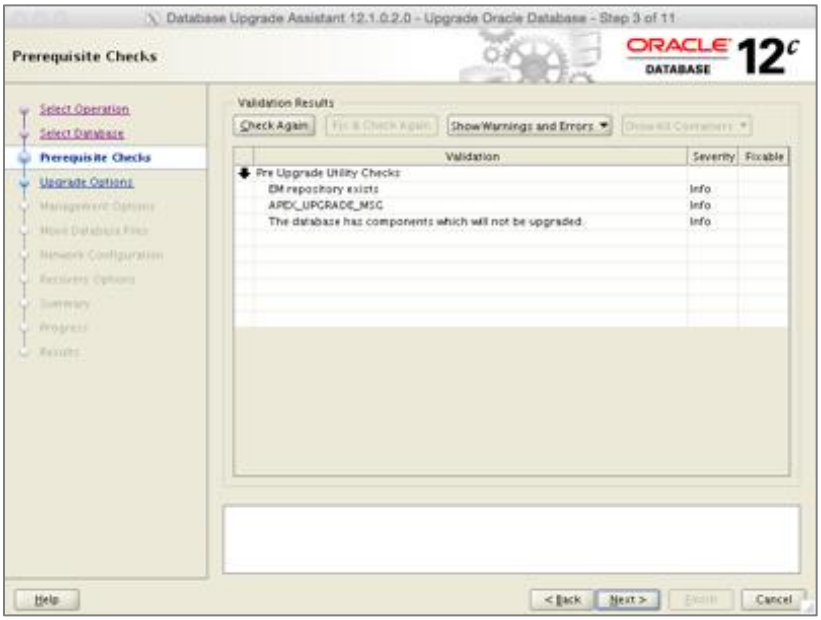
DBUA on GUI Mode

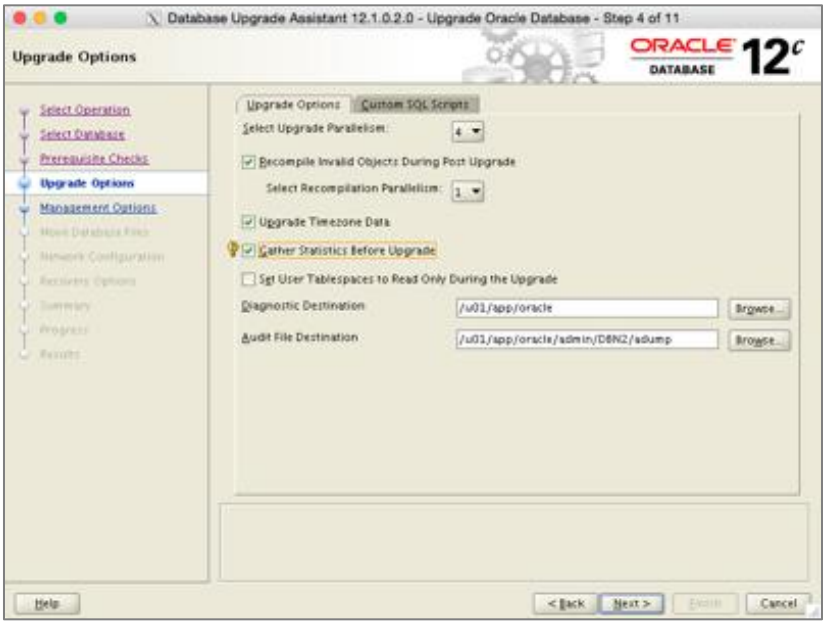
```
[oracle@edbcn2 dbhome_1]$ cd /u01/app/oracle/product/12.1.0/dbhome_1/bin
[oracle@edbcn2 bin]$ export DISPLAY=192.168.1.200:0.0
[oracle@edbcn2 bin]$ ./dbua
```

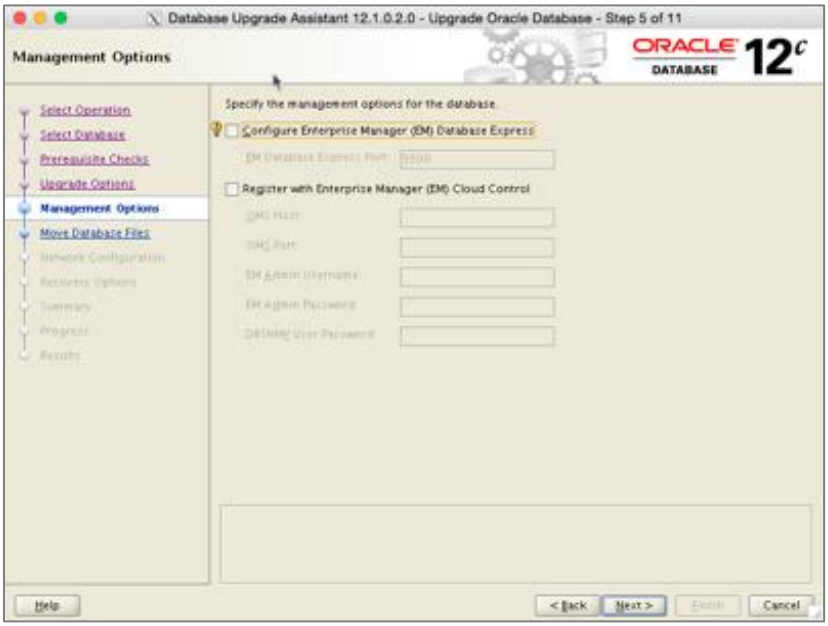


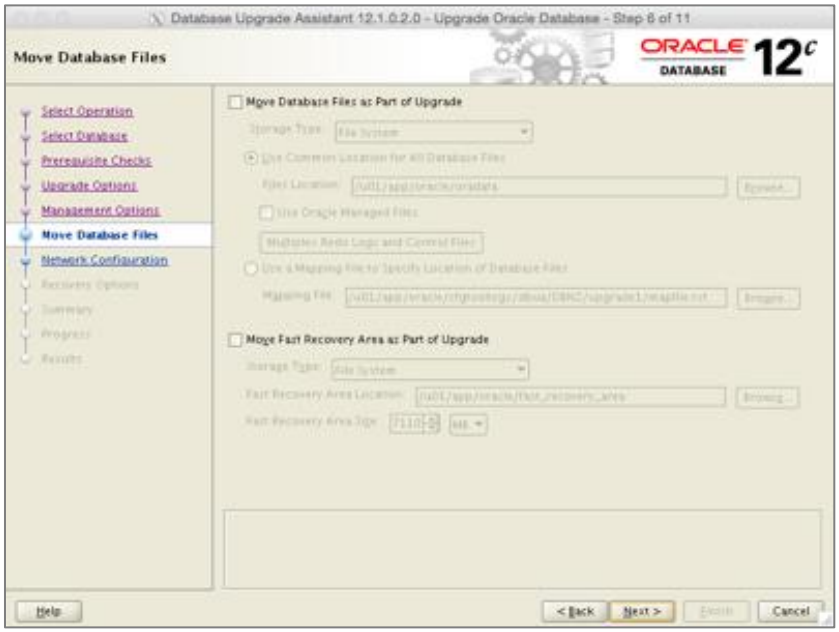


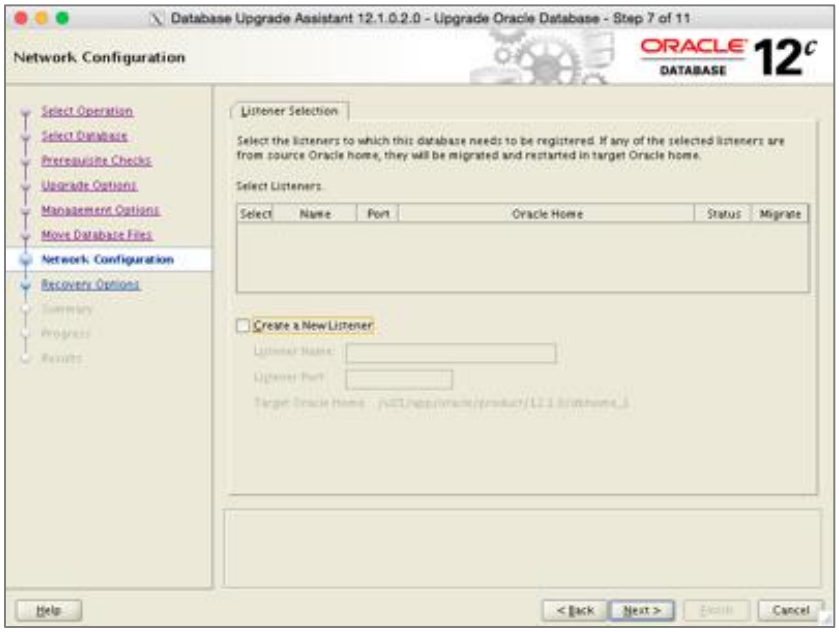

Database Upgrade Assistant 12.1.0.2.0 - Upgrade Oracle Database - Step 3 of 11
Prerequisite Checks
ORACLE DATABASE 12c
Select Operation
Select Database
Prerequisite Checks
Upgrade Options
Management Options
Move Database Files
Network Configuration
Recovery Options
Summary
Progress
Results
Prerequisite Checks
0%
Run Pre Upgrade Utility ...
Help
< Back
Next >
Finish
Cancel

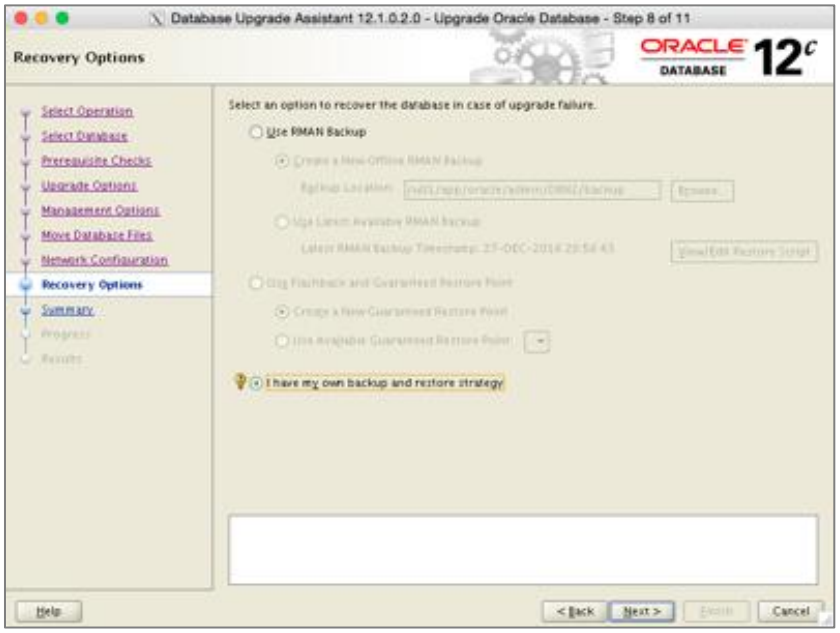

Database Upgrade Assistant 12.1.0.2.0 - Upgrade Oracle Database - Step 3 of 11
Prerequisite Checks
ORACLE DATABASE 12c
Select Operation
Select Database
Prerequisite Checks
Upgrade Options
Management Options
Move Database Files
Network Configuration
Recovery Options
Summary
Progress
Results
Validation Results
Check Again
Fix & Check Again
Show Warnings and Errors
Show All Comments
Validation
Severity
Fixable
Pre Upgrade Utility Checks
EM repository exists
Info
APEX_UPGRADE_MSG
Info
The database has components which will not be upgraded.
Info
Help
< Back
Next >
Finish
Cancel

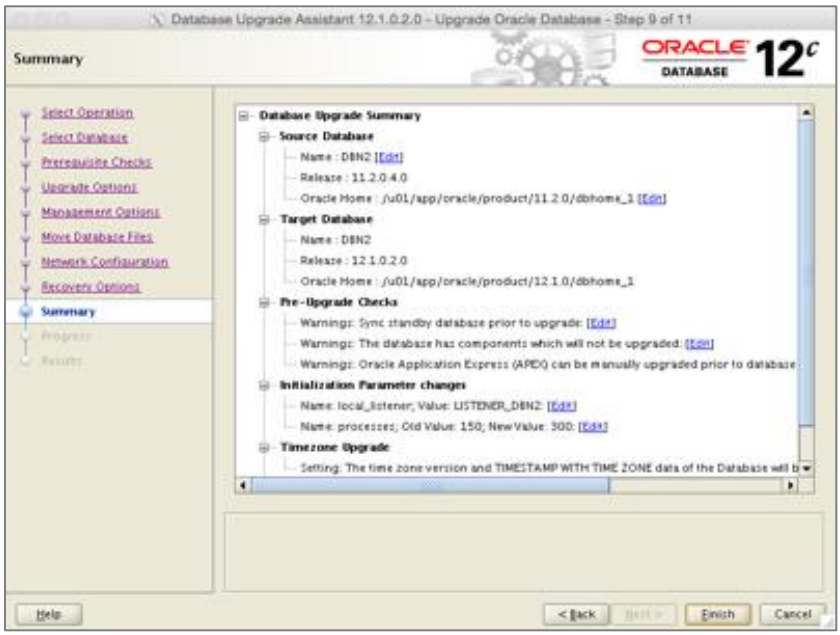

Database Upgrade Assistant 12.1.0.2.0 - Upgrade Oracle Database - Step 4 of 11
Upgrade Options
ORACLE DATABASE 12c
Select Operation
Select Database
Prerequisite Checks
Upgrade Options
Management Options
Move Database Files
Network Configuration
Recovery Options
Summary
Progress
Results
Upgrade Options
Custom SQL Scripts
Select Upgrade Parallelism:
4
Recompile Invalid Objects During Post Upgrade
Select Recompilation Parallelism:
1
Upgrade Timezone Data
Gather Statistics Before Upgrade
Set User Tablespaces to Read Only During the Upgrade
Diagnostic Destination
/u01/app/oracle
Browse...
Audit File Destination
/u01/app/oracle/admin/DBN2/adump
Browse...
Help
< Back
Next >
Finish
Cancel

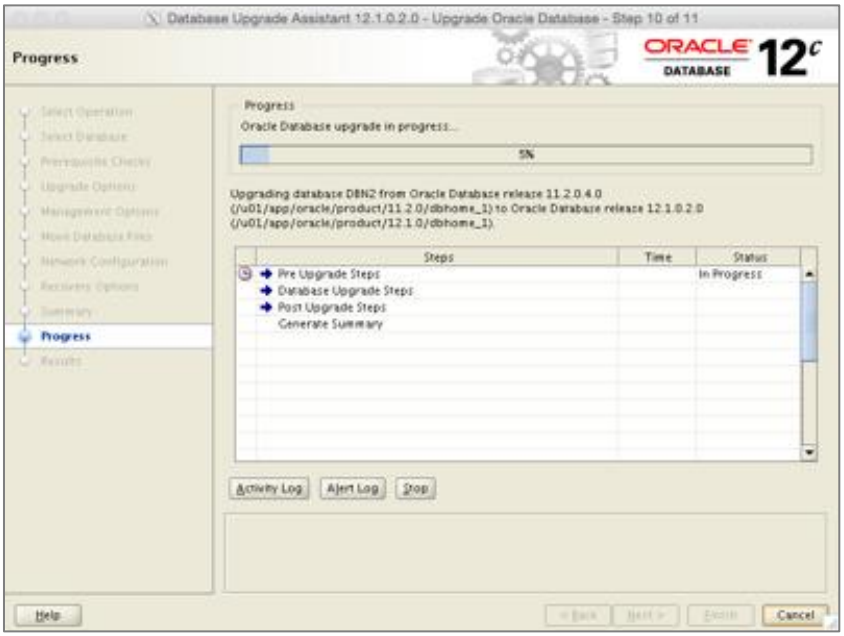

Database Upgrade Assistant 12.1.0.2.0 - Upgrade Oracle Database - Step 5 of 11
Management Options
ORACLE DATABASE 12c
Select Operation
Select Database
Prerequisite Checks
Upgrade Options
Management Options
Move Database Files
Network Configuration
Recovery Options
Summary
Progress
Results
Specify the management options for the database.
Configure Enterprise Manager (EM) Database Express
Register with Enterprise Manager (EM) Cloud Control
Help
< Back
Next >
Finish
Cancel

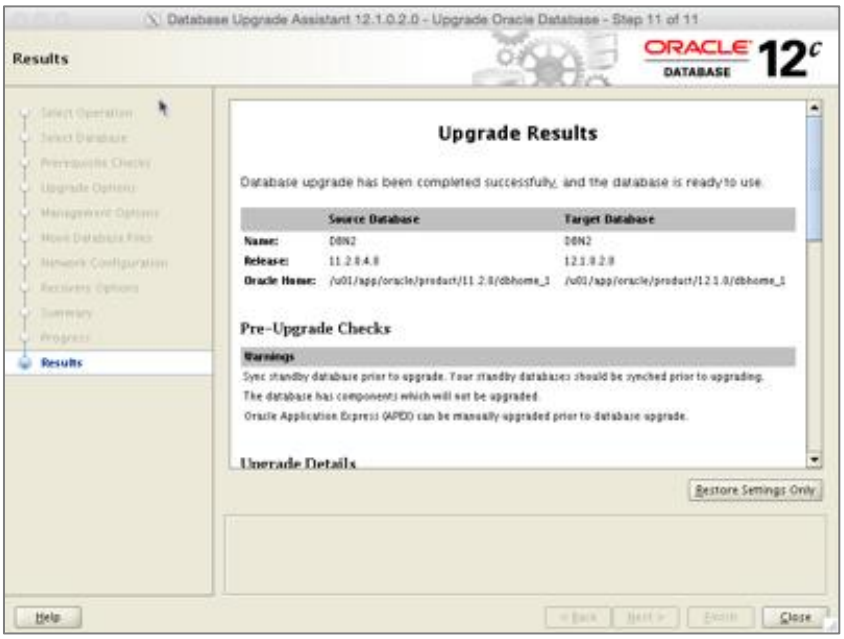

Database Upgrade Assistant 12.1.0.2.0 - Upgrade Oracle Database - Step 6 of 11
Move Database Files
ORACLE DATABASE 12c
Select Operation
Select Database
Prerequisite Checks
Upgrade Options
Management Options
Move Database Files
Network Configuration
Recovery Options
Summary
Progress
Results
Move Database Files as Part of Upgrade
Move Fast Recovery Area as Part of Upgrade
Help
< Back
Next >
Finish
Cancel


Database Upgrade Assistant 12.1.0.2.0 - Upgrade Oracle Database - Step 7 of 11
Network Configuration
ORACLE DATABASE 12c
Select Operation
Select Database
Prerequisite Checks
Upgrade Options
Management Options
Move Database Files
Network Configuration
Recovery Options
Summary
Progress
Results
Listener Selection
Select the listeners to which this database needs to be registered. If any of the selected listeners are from source Oracle home, they will be migrated and restarted in target Oracle home.
Select Listeners
Select
Name
Port
Oracle Home
Status
Migrate
Create a New Listener
Help
< Back
Next >
Finish
Cancel


Database Upgrade Assistant 12.1.0.2.0 - Upgrade Oracle Database - Step 8 of 11
Recovery Options
ORACLE DATABASE 12c
Select Operation
Select Database
Prerequisite Checks
Upgrade Options
Management Options
Move Database Files
Network Configuration
Recovery Options
Summary
Progress
Results
Select an option to recover the database in case of upgrade failure.
Use RMAN Backup
I have my own backup and restore strategy
Help
< Back
Next >
Finish
Cancel


Database Upgrade Assistant 12.1.0.2.0 - Upgrade Oracle Database - Step 9 of 11
Summary
ORACLE DATABASE 12c
Select Operation
Select Database
Prerequisite Checks
Upgrade Options
Management Options
Move Database Files
Network Configuration
Recovery Options
Summary
Progress
Results
Database Upgrade Summary
Source Database
Name : DBN2 [Edit]
Release : 11.2.0.4.0
Oracle Home : /u01/app/oracle/product/11.2.0/dbhome_1 [Edit]
Target Database
Name : DBN2
Release : 12.1.0.2.0
Oracle Home : /u01/app/oracle/product/12.1.0/dbhome_1
Pre-Upgrade Checks
Warnings: Sync standby database prior to upgrade: [Edit]
Warnings: The database has components which will not be upgraded: [Edit]
Warnings: Oracle Application Express (APEX) can be manually upgraded prior to database
Initialization Parameter changes
Name: local_listener; Value: LISTENER_DBN2: [Edit]
Name: processes; Old Value: 150; New Value: 300: [Edit]
Timezone Upgrade
Setting: The time zone version and TIMESTAMP WITH TIME ZONE data of the Database will b
Help
< Back
Next >
Finish
Cancel






```
[oracle@edbcn2 bin]$

Database upgrade has been completed successfully, and the database is ready to
use.
```

Task 16: Perform Oracle Database 12c Post-Upgrade

Update the shell startup file - .bash_profile. Set ORACLE_HOME variable.

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi
# User specific environment and startup programs
umask 022
DISPLAY=:0.0;export DISPLAY
EDITOR=vi;export EDITOR
PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin:$HOME

ORACLE_BASE=/u01/app/oracle
ORACLE_HOME=$ORACLE_BASE/product/12.1.0/dbhome_1
OGG_HOME=$ORACLE_BASE/product/12.2.1/gghome_1
LD LIBRARY PATH=/usr/lib:/usr/X11R6/lib:$ORACLE HOME/lib
ORACLE_DIAG=/u01/app/oracle/diag/rdbms/dbn2/DBN2/trace
PATH=$ORACLE_HOME/bin:$PATH:$ORACLE_HOME/OPatch
TMP=/tmp; export TMP
TMPDIR=/tmp; export TMPDIR
ORACLE SID=DBN2
export PATH LD_LIBRARY_PATH
export ORACLE_BASE ORACLE_HOME ORACLE_SID OGG_HOME ORACLE_DIAG
```

```
[oracle@edbcn2 ~]$ . .bash_profile
[oracle@edbcn2 ~]$
```

Verify the database is upgraded.

On extended node 2

```
Start SQL*Plus and verify Oracle Database 12c is activated.

[oracle@edbcn2 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sat Jan 14 09:17:58 2017

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

SQL> conn / as sysdba
Connected.

SQL> select banner
  2  from   v$version;

BANNER
--------------------------------------------------------------------------------
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
PL/SQL Release 12.1.0.2.0 - Production
CORE 12.1.0.2.0     Production
```

```
TNS for Linux: Version 12.1.0.2.0 - Production
NLSRTL Version 12.1.0.2.0 - Production

SQL>
```

Migrate the listener to Oracle Database 12c.

```
[oracle@edbcn2 admin]$ cp
/u01/app/oracle/product/11.2.0/dbhome_1/network/admin/tnsnames.ora
$ORACLE_HOME/network/admin
[oracle@edbcn2 admin]$
```

Change the ORACLE_HOME parameter

```
# listener.ora Network Configuration File:
/u01/app/oracle/product/11.2.0/dbhome 1/network/admin/listener.ora
# Generated by Oracle configuration tools.

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = DBN2.oracle.com)
      (ORACLE_HOME = /u01/app/oracle/product/12.1.0/dbhome_1)
      (SID_NAME = DBN2)
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = edbcn2.oracle.com)(PORT = 1521))
    )
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))
    )
  )

ADR_BASE_LISTENER = /u01/app/oracle
```

Restart the listener

```
 [oracle@edbcn2 admin]$ lsnrctl

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 14-JAN-2017 09:33:27

Copyright (c) 1991, 2014, Oracle.  All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> stop
Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
The command completed successfully
LSNRCTL> start
Starting /u01/app/oracle/product/12.1.0/dbhome_1/bin/tnslsnr: please wait...
```

```
TNSLSNR for Linux: Version 12.1.0.2.0 - Production
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome 1/network/admin/listener.ora
Log messages written to
/u01/app/oracle/diag/tnslsnr/edbcn2/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn2.oracle.com)(PORT=1521)))

Connecting to (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.2.0 - Production
Start Date                14-JAN-2017 09:33:32
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/edbcn2/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn2.oracle.com)(PORT=1521)))
Services Summary...
Service "DBN2.oracle.com" has 1 instance(s).
  Instance "DBN2", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
LSNRCTL>
```

Recompile invalid objects

```
SQL> conn osm_repapi/oracle
Connected.
SQL> alter procedure trans_load compile;

Procedure altered.

SQL> alter procedure call_trans_load compile;

Procedure altered.

SQL>
```

Task 17: Enable Database Flashback

Obtain the current SCN, then enable the database flashback. This is for the purpose of application testing only, done as follow:

1. Obtain the current SCN
2. Enable database flashback
3. Perform application test
4. Rewind the database

5. Disable database flashback

On extended node 2

```
SQL> select current_scn
  2  from   v$database;

CURRENT_SCN
-----------
   10625785

SQL> alter database flashback on;

Database altered.


SQL>
```

Task 18: Perform Application Testing

The database is on flashback mode, this allows developers and users to perform real-testing before go-live mode.

Let the session run for approximately 15 minutes to generate enough Flashback log files.

On extended node 2

```
SQL> conn osm_repapi/oracle
Connected.

SQL> exec call_trans_load;
^CBEGIN call_trans_load; END;

*
ERROR at line 1:
ORA-01013: user requested cancel of current operation
ORA-06512: at "SYS.DBMS_LOCK", line 205
ORA-06512: at "OSM_REPAPI.CALL_TRANS_LOAD", line 17
ORA-06512: at line 1



SQL>
```

Task 19: Disable Database Flashback and Rewind the Database

Rewind the database to the SCN just before enabling database flashback.

On extended node 2

```
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount
ORACLE instance started.

Total System Global Area          1660944384 bytes
Fixed Size                           2925072 bytes
Variable Size                     1056968176 bytes
Database Buffers                   587202560 bytes
Redo Buffers                        13848576 bytes
Database mounted.
SQL>
SQL> flashback database to scn 9507634;

Flashback complete.

SQL>
SQL> alter database open resetlogs;

Database altered.

SQL> alter database flashback off;

Database altered.

SQL>
```

Step 20: Enable Redo Transport

It's now time to enable Redo Transport from extended node to the downstream system – C1DS2A.

On extended node 2

```
SQL> conn / as sysdba
Connected.
SQL> alter system set log_archive_dest_state_2='enable' scope=both;

System altered.

SQL> show parameters log_archive_dest_state_2

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
log_archive_dest_state_2             string      enable
log archive dest state 20            string      enable
log archive dest state 21            string      enable
log_archive_dest_state_22            string      enable
log_archive_dest_state_23            string      enable
log_archive_dest_state_24            string      enable
```

```
log_archive_dest_state_25       string     enable
log_archive_dest_state_26       string     enable
log_archive_dest_state_27       string     enable
log_archive_dest_state_28       string     enable
log_archive_dest_state_29       string     enable

System altered.

SQL>
```

Verify the Redo-Transport is shipping transaction logs to extended node 2.

On extended node 2

```
SQL> alter system switch logfile;

System altered.

SQL>
```

On extended node 2

```
[oracle@edbcn2 ~]$ cd $ORACLE_DIAG
[oracle@edbcn2 trace]$ tail -f *.log
Archived Log entry 436 added for thread 1 sequence 1 ID 0x5fc6c8ee dest 1:
Sat Jan 14 15:02:20 2017
**********************************************************************
LGWR: Setting 'active' archival for destination LOG_ARCHIVE_DEST_2
**********************************************************************
Sat Jan 14 15:02:21 2017
Thread 1 advanced to log sequence 3 (LGWR switch)
  Current log# 3 seq# 3 mem# 0: /u01/app/oracle/oradata/DBN2/redo03.log
Sat Jan 14 15:02:21 2017
Archived Log entry 439 added for thread 1 sequence 2 ID 0x5fc6c8ee dest 1:
Sat Jan 14 15:06:15 2017
**********************************************************************
LGWR: Setting 'active' archival for destination LOG_ARCHIVE_DEST_2
**********************************************************************
Sat Jan 14 15:06:16 2017
Thread 1 advanced to log sequence 4 (LGWR switch)
  Current log# 1 seq# 4 mem# 0: /u01/app/oracle/oradata/DBN2/redo01.log
Sat Jan 14 15:06:16 2017
Archived Log entry 441 added for thread 1 sequence 3 ID 0x5fc6c8ee dest 1:
```

It demonstrates redo shipping is working.

Task 21: Re-Register Extended Node 2 Integrated Extract

Login the downstream mining database and extended node 2

On downstream system – C1DS2A

```
GGSCI (c1ds2.oracle.com) 33> miningdblogin userid ogg_admin@c1ds2a.oracle.com,
password oracle
Successfully logged into mining database.

GGSCI (c1ds2.oracle.com) 34> dblogin userid ogg_admin@dbn2.oracle.com, password
oracle
Successfully logged into database.
```

Unregister the integrated extract EDBN201, then register it again.

```
GGSCI (c1ds2.oracle.com as ogg admin@DBN2) 39> unregister extract EDBN201,
database

2017-01-15 14:18:55  INFO    OGG-01750  Successfully unregistered EXTRACT
EDBN201 from database.

GGSCI (c1ds2.oracle.com as ogg_admin@DBN2) 40> register extract EDBN201,
database

2017-01-15 14:19:34  INFO    OGG-02003  Extract EDBN201 successfully registered
with database at SCN 10659236.

GGSCI (c1ds2.oracle.com as ogg_admin@DBN2) 41>
```

Wait for couple of minutes for the Integrated Extract start…

```
GGSCI (c1ds2.oracle.com as ogg_admin@DBN2) 57> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     STARTING    EDBN201     52:53:27      00:05:58
EXTRACT     RUNNING     PDBN201     00:00:00      00:00:03
EXTRACT     RUNNING     PDBN202     00:00:00      00:00:02


GGSCI (c1ds2.oracle.com as ogg_admin@DBN2) 58>

GGSCI (c1ds2.oracle.com as ogg_admin@DBN2) 58> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EDBN201     00:00:17      00:00:03
EXTRACT     RUNNING     PDBN201     00:00:00      00:00:09
EXTRACT     RUNNING     PDBN202     00:00:00      00:00:07


GGSCI (c1ds2.oracle.com as ogg_admin@DBN2) 59>
```

Task 22: Re-Create the Checkpoint Database Table

Because it's the default for Replicat, the checkpoint table must be re-created.

On extended node 2

```
[oracle@edbcn2 gghome_1]$ vi GLOBALS
```

```
checkpointable ogg_admin.chkpt_tab
```

```
[oracle@edbcn2 gghome_1]$ ./ggsci

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 UTF-8.

Copyright (C) 1995, 2015, Oracle and/or its affiliates. All rights reserved.



GGSCI (edbcn2.oracle.com) 1> dblogin userid ogg_admin@dbn2.oracle.com, password
oracle
Successfully logged into database.

GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 2> add checkpointtable

No checkpoint table specified. Using GLOBALS specification
(ogg_admin.chkpt_tab)...

Successfully created checkpoint table ogg_admin.chkpt_tab.

GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 3>
```

Task 23: Re-Define the Replicat on Extended Node 2

Redefine the Replicat for extended node 2:

- RDBN002
- RDBN102

These Replicat will start from the SCN obtain in Task 4.

```
GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 4> edit params rdbn002
```

Change SETENV for ORACLE_HOME

```
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
```

```
GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 5> view params rdbn002

replicat RDBN002
SETENV (ORACLE_SID = DBN2)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
USERID ogg_admin@dbn2.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn002.dsc, PURGE, megabytes 2000
-- MAP osm_repapi.*, TARGET osm_repapi.*;
MAP osm repapi.customers, TARGET osm repapi.customers, &
-- Implement Conflict Detection and Resolution - CDR
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
--
MAP osm_repapi.policies, TARGET osm_repapi.policies, &
-- Implement Conflict Detection and Resolution - CDR
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));



GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 6>
```

Define the Replicat - RDBN002

On extended node 2

```
GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 3> add replicat  RDBN002, exttrail
./dirdat/tp
REPLICAT added.


GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 4>
```

```
GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 7> edit params rdbn102
```

Change SETENV for ORACLE_HOME

```
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
```

```
GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 8> view params rdbn102

replicat RDBN102
SETENV (ORACLE_SID = DBN2)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
```

```
USERID ogg_admin@dbn2.oracle.com, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn102.dsc, PURGE, megabytes 2000
-- MAP osm_repapi.*, TARGET osm_repapi.*;
MAP osm_repapi.customers, TARGET osm_repapi.customers, &
-- Implement Conflict Detection and Resolution - CDR
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
--
MAP osm_repapi.policies, TARGET osm_repapi.policies, &
-- Implement Conflict Detection and Resolution - CDR
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));



GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 9>
```

Define the Replicat – RDBN102

On extended node 2

```
GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 6> add replicat RDBN102, exttrail
./dirdat/t1
REPLICAT added.


GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 7>
```

Task 24: Start Replicats on Extended Node 2

Start the Replicat from SCN obtained in Task 4.

RDBN002: 10408012
RDBN102: 10415608

```
GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 17> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    STOPPED     RDBN002     00:00:00      00:19:22
REPLICAT    STOPPED     RDBN102     00:00:00      00:17:41
```

```
GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 18> start replicat RDBN002, AfterCSN
10408012

Sending START request to MANAGER ...
REPLICAT RDBN002 starting


GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 19> start replicat RDBN102, AfterCSN
10415608

Sending START request to MANAGER ...
REPLICAT RDBN102 starting


GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 20> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN002     02:43:36      00:00:09
REPLICAT    RUNNING     RDBN102     00:00:00      00:00:02


GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 21>
```

Task 25: Verify Data Consistency

Due to data discrepancy on the database, there will be divergence. Run the application on the primary node and check for data convergence. For online data verification, use Oracle GoldenGate 12c Veridata. Refer to Appendix C for further details.

On the primary node

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

Query and monitor data convergence…

```
SQL> select * from (select 'DBN0@DBN0' DB_NAME, count(*) ROWS# from customers
where bas_x='DBN0' and dml_ts is not null) union
  2  select * from (select 'DBN0@DBN1' DBN0_DBN1, count(*) ROWS# from
customers@dbn1.oracle.com where bas_x='DBN0' and dml_ts is not null) union
  3  select * from (select 'DBN0@DBN2' DBN0_DBN2, count(*) ROWS# from
customers@dbn2.oracle.com where bas_x='DBN0' and dml_ts is not null);

DB_NAME        ROWS#
--------- ----------
DBN0@DBN0   2261
DBN0@DBN1   2231
DBN0@DBN2   2251
```

```
SQL> /

DB NAME         ROWS#
--------- ----------
DBN0@DBN0   2531
DBN0@DBN1   2531
DBN0@DBN2   2531

SQL> /

DB NAME         ROWS#
--------- ----------
DBN0@DBN0   3231
DBN0@DBN1   3221
DBN0@DBN2   3221

SQL>
```

Verify data convergence on node 2

```
SQL> conn osm_repapi/oracle@dbn2.oracle.com
Connected.

SQL> select count(*) from customers where bas_x='DBN0';

  COUNT(*)
----------
      3051

SQL>
```

Task 26: Divert Applications to Run from Extended Node 2

It's now safe to run the application from extended node 2.

On extended node 2

```
[oracle@edbcn0 ~]$ sqlplus osm_repapi/oracle

SQL*Plus: Release 12.1.0.2.0 Production on Sat Jan 14 15:31:37 2017

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

Last Successful login time: Sat Jan 14 2017 09:41:14 +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

SQL>
SQL> exec call_trans_load;
```

Task 27: Reconfigure RMAN Catalog Scripts

After the upgrade, the Extended Database Cluster 12c (EDBC) operated using Oracle Database 12c R1 only. Change the Recovery Manager (RMAN) stored script accordingly.

On extended node 2

```
#!/bin/sh
export TNS_ADMIN=/u01/app/oracle/product/12.1.0/dbhome_1/network/admin
export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
export LD_LIBRARY_PATH=/u01/app/oracle/product/12.1.0/dbhome_1/lib
export ORACLE_SID=DBN2
/u01/app/oracle/product/12.1.0/dbhome 1/bin/rman catalog
edbc_rcv_cat_mgr/oracle@DBN0.oracle.com target sys/oracle@DBN2.oracle.com
@DBN2_BCKUP.rcv log=DBN2_BCKUP.log append
```

Testing backup for extended node 2 – DBN2

```
[oracle@edbcn0 ~]$ ./DBN2_BCKUP.sh
RMAN> 2> 3>
```

```
[oracle@edbcn0 ~]$ tail -f DBN2_BCKUP.log
channel c3: specifying datafile(s) in backup set
including current control file in backup set
channel c3: starting piece 1 at 14-JAN-17
channel c3: finished piece 1 at 14-JAN-17
piece handle=/home/oracle/BACKUP/dbf/DBN2/DBF_0erq160v_1_1.dbf
tag=TAG20170114T180513 comment=NONE
channel c3: backup set complete, elapsed time: 00:00:18
channel c3: starting full datafile backup set
channel c3: specifying datafile(s) in backup set
including current SPFILE in backup set
channel c3: starting piece 1 at 14-JAN-17
channel c3: finished piece 1 at 14-JAN-17
piece handle=/home/oracle/BACKUP/dbf/DBN2/DBF_0frq1621_1_1.dbf
tag=TAG20170114T180513 comment=NONE
channel c3: backup set complete, elapsed time: 00:00:15
```

The tail –f command shows the backup is doing great and in progress.

Task 28: Reconfigure Administration Scripts

The startup, shutdown scripts for extended node 2 are modified accordingly.

Starting Oracle GoldenGate 12c instance

On extended node 2

```
#!/bin/sh
#printf '\e[8;20;150t'
#printf '\e[3;0;350t'
#printf '\e[31m'
#printf '\e[1m'
export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
export LD_LIBRARY_PATH=/u01/app/oracle/product/12.1.0/dbhome_1/lib
#/u01/app/oracle/product/12.1.0/gghome_1/ggsci
#read -p "Enter"
echo obey /u01/app/oracle/product/12.2.0/gghome_1/dirprm/ggstart.oby |
/u01/app/oracle/product/12.2.0/gghome_1/ggsci
```

Shutdown Oracle GoldenGate 12c instance

On extended node 2

```
#!/bin/sh
#printf '\e[8;20;150t'
#printf '\e[3;0;350t'
#printf '\e[31m'
#printf '\e[1m'
export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
export LD_LIBRARY_PATH=/u01/app/oracle/product/12.1.0/dbhome_1/lib
#/u01/app/oracle/product/12.1.0/gghome_1/ggsci
#read -p "Enter"
echo obey /u01/app/oracle/product/12.2.0/gghome_1/dirprm/ggstop.oby |
/u01/app/oracle/product/12.2.0/gghome_1/ggsci
```

Startup Oracle Database 12c instance

On extended node 2

```
#!/bin/sh
export ORACLE_SID=DBN2
export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
lsnrctl start
sqlplus /nolog <<EOF
connect / as sysdba
startup
EOF
```

Shutdown Oracle Database 12c instance

On extended node 2

```
#!/bin/sh
export ORACLE_SID=DBN2
export ORACLE_HOME=/u01/app/oracle/product/12.1.0/dbhome_1
lsnrctl stop
sqlplus /nolog <<EOF
connect / as sysdba
shutdown immediate
```

```
EOF
```

Workshop 11: Secure EDBC

This workshop applies security features and options to maximize the security of the Extended Database Cluster 12c (EDBC). Depending on the level of data sensitivity, only the specific feature and options are installed and configured. Other database options such as TDE, database vault, audit vault and firewall are implementable options, but out of the workshop's scope.

Table 15 lists Workshop 11 tasks.

Task No	Task Description
1	Enable Compression Redo-Transport
2	Enable Encryption of Redo-Transport
3	Enable Password Encryption
4	Command Authentication
5	Configure Secure Trail and Data Streaming
6	Configure Secure Manager
7	Configure EDBC Backup Encryption
8	Deploy Tablespace encryption

Table 15: Workshop 11 Tasks

Task 1: Enable Redo-Transport Compression

The compression of data offload the network bandwidth from unnecessary traffic. When data compressed at the primary node and extended node, the transaction log files reduced to up to 80%, hence reducing the gap, if any, and optimizing the bandwidth utilization.

On the primary node

Configure Redo-Transport compression by setting the COMPRESSION attribute on LOG_ARHIVE_DEST_2 parameter.

```
SQL> alter system set log archive dest 2='SERVICE=c1ds1a.oracle.com SYNC
NOREGISTER VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) REOPEN=10
DB UNIQUE NAME=C1DS1A COMPRESSION=ENABLE' scope=both;

System altered.

SQL>
```

Verify compression is enabled.

```
SQL> select dest_id, compression
  2  from   V$archive_dest
  3  where  dest id=2;

   DEST ID COMPRES
---------- -------
         2 ENABLE

SQL>
```

Calcuating the compression ratio for a standby redo log file

On downstream system – C<n>DS1A

```
[root@c1ds1 standby]# gzip -1 1_318_925818654.dbf
[root@c1ds1 standby]# gzip --list 1_318_925818654.dbf
         compressed        uncompressed  ratio uncompressed_name
            4179446            21050368  80.1% 1_318_925818654.dbf
[root@c1ds1 standby]#
```

On extended node 1

Configure Redo-Transport compression by setting the COMPRESSION attribute on LOG_ARHIVE_DEST_2 parameter.

```
SQL> alter system set log_archive_dest_2='SERVICE=c1ds1b.oracle.com SYNC
NOREGISTER VALID FOR=(ONLINE LOGFILES,PRIMARY ROLE) REOPEN=10
DB_UNIQUE_NAME=C1DS1B COMPRESSION=ENABLE' scope=both;

System altered.

SQL>
```

Verify compression is enabled.

```
SQL> select dest_id, compression
  2  from   V$archive_dest
  3  where  dest_id=2;

   DEST_ID COMPRES
---------- -------
         2 ENABLE

SQL>
```

On extended node 2

Configure Redo-Transport compression by setting the COMPRESSION attribute on LOG_ARHIVE_DEST_2 parameter.

```
SQL> alter system set log_archive_dest_2='SERVICE=c1ds2a.oracle.com SYNC
NOREGISTER VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) REOPEN=10
DB UNIQUE NAME=C1DS2A COMPRESSION=ENABLE' scope=both;

System altered.

SQL>
```

Verify compression is enabled.

```
SQL> select dest id, compression
  2  from   V$archive_dest
  3  where  dest id=2;

   DEST_ID COMPRES
---------- -------
         2 ENABLE

SQL>
```

Task 2: Enable Redo-Transport Encryption

Because the Redo-Transport encryption uses Oracle Advanced Security option, ensure the option is installed on the primary node, extended nodes and downstream systems.

On the primary node

```
[oracle@edbcn0 ~]$ adapters

Installed Oracle Net transport protocols are:

    IPC
    BEQ
    TCP/IP
    SSL
    RAW
    SDP/IB
    ExaDirect

Installed Oracle Net naming methods are:

    Local Naming (tnsnames.ora)
    Oracle Directory Naming
    Oracle Host Naming

Installed Oracle Advanced Security options are:

    RC4 40-bit encryption
    RC4 56-bit encryption
```

```
    RC4 128-bit encryption
    RC4 256-bit encryption
    DES40 40-bit encryption
    DES 56-bit encryption
    3DES 112-bit encryption
    3DES 168-bit encryption
    AES 128-bit encryption
    AES 192-bit encryption
    AES 256-bit encryption
    MD5 crypto-checksumming
    SHA-1 crypto-checksumming
    Kerberos v5 authentication
    RADIUS authentication
[oracle@edbcn0 ~]$
```

Repeat the step for the extended nodes.

Let's enable Redo-Transport encryption for the primary and downstream system C1DS1A.

- Modify sqlnet.ora file on the primary node

```
# setting the encryption parameters
sqlnet.crypto seed = "kclabefmnoc"
sqlnet.encryption_server = accepted
sqlnet.encryption_client = required
sqlnet.encryption_types_client = AES128
sqlnet.encryption_types_server = AES128
sqlnet.trace directory server=/u01/app/oracle/product/12.1.0/dbhome 1/network/lo
g
sqlnet.trace level client=16
sqlnet.trace_level_server=16
```

- Redstart the primary database

```
[oracle@edbcn0 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sat Jan 21 19:05:50 2017

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

SQL> conn / as sysdba
Connected.
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area          2516582400 bytes
Fixed Size                           2927528 bytes
Variable Size                     1560282200 bytes
Database Buffers                   939524096 bytes
```

```
Redo Buffers                          13848576 bytes
Database mounted.
Database opened.
SQL>
```

- Modify sqlnet.ora file on downsteam system C1DS1A

```
# setting the encryption parameters
sqlnet.crypto_seed = "kclabefmnoc"
sqlnet.encryption_server = accepted
sqlnet.encryption client = required
sqlnet.encryption types client = AES128
sqlnet.encryption types server = AES128
sqlnet.trace_directory_server=/u01/app/oracle/product/12.1.0/dbhome_1/network/lo
g
sqlnet.trace_level_client=16
sqlnet.trace_level_server=16
```

- Redstart the downstream database C1DS1A

```
[oracle@c1ds1 ~]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sat Jan 21 19:08:05 2017

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

SQL> conn / as sysdba
Connected.
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup
ORACLE instance started.

Total System Global Area          2516582400 bytes
Fixed Size                           2927528 bytes
Variable Size                     1560282200 bytes
Database Buffers                   939524096 bytes
Redo Buffers                        13848576 bytes
Database mounted.
Database opened.
SQL>
```

- Perform Redo-Transport encryption test by running a transaction on the primary node and check data encryption on the downstream net trace file. Data should not be displayed on plain-text.

Repeat the steps for extended node 1 and downstream C1DS1B and extended node 2 and downstream C1DS2A.

Task 3: Password Encryption

Instead of using clear password inside Oracle GoldenGate 12c configuration parameter files, there are two methods of password encrtyption:

- Method 1: Password encryption
- Method 2: Password encrypted certificate

Method 1 will be implemented to demonstrate password security. Refer to Appendix A for further details regarding method 1.

Password Encryption

The command to encrypt the password:

```
ENCRYPT PASSWORD password algorithm ENCRYPTKEY {key_name | DEFAULT}
```

Supported algorithm: AES128, AES192, AES256 and BLOWFISH. AES128 (AES default) algorithm will used to create the password encryption for method 1. Avoid BLOWFISH algorithm when your platform supports AES algorithm. The following steps configure password encryption:

1. Generate the encrtyption key values
2. Create the look up file – ENCKEYS
3. Copy ENCKEYS across EDBC systems
4. Encrypt the password found in the parameter files
5. Modify the parameter files

Step 1: Generate the ENCKYES look up file Encryption Key values

The command ./keygen generate 6 AES128 keys. The key values will be used across the EDBC Oracle GoldenGate 12c instances.

On downstream system C<n>DS1a

```
[oracle@c1ds1 gghome 1]$ ./keygen 128 6
0x26719158BEA89920C790302244912538

0x3FFFDB03A969F3299EAD167DAC7FB178

0xADE22B2F2247346CCC58700C9A0A9D1B

0x1BC67B5A9C24752EF903CA1B8795883E

0x8AA9CB051502B67026AF232B74207461
```

```
0xF88C1B318FDFF632535A7D3A62AB5F04

[oracle@c1ds1 gghome_1]$
```

Step 2: Create the look up file – ENCKEYS

Save the file in Oracle GoldenGate 12c root directory, using the format upper case without file extention. Copy-Paste the out of step 1 into ENCKEYS. Prefix each encrypted key value with a lookup key name, as shown below. The faimiliar encrtyption key names is the database SID – c1ds1a, c1ds1b, c1ds2a, dbn0, dbn1 and dbn2.

```
[oracle@c1ds1 gghome_1]$ vi ENCKEYS

c1ds1a  0x26719158BEA89920C790302244912538

c1ds1b  0x3FFFDB03A969F3299EAD167DAC7FB178

c1ds2a  0xADE22B2F2247346CCC58700C9A0A9D1B

dbn0    0x1BC67B5A9C24752EF903CA1B8795883E

dbn1    0x8AA9CB051502B67026AF232B74207461

dbn2    0xF88C1B318FDFF632535A7D3A62AB5F04
```

Step 3: Copy and change permission for ENCKEYS file across all EDBC systems:

edbcn0, edbcn1, edbcn2, c1ds1 ($OGG_HOME2) and c1ds2

```
[oracle@c1ds1 gghome_1]$ scp ENCKEYS
root@edbcn0:/u01/app/oracle/product/12.2.1/gghome_1
The authenticity of host 'edbcn0 (192.168.1.51)' can't be established.
RSA key fingerprint is 4d:02:44:9f:14:3e:dc:43:f4:da:44:08:e6:37:a3:05.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'edbcn0,192.168.1.51' (RSA) to the list of known
hosts.
root@edbcn0's password:
ENCKEYS
100%  251     0.3KB/s   00:00
[oracle@c1ds1 gghome_1]$
```

```
[oracle@c1ds1 gghome_1]$ scp ENCKEYS
root@edbcn1:/u01/app/oracle/product/12.2.1/gghome_1
The authenticity of host 'edbcn1 (192.168.1.61)' can't be established.
RSA key fingerprint is e3:c9:12:b6:ea:85:fb:e1:55:9c:51:21:dd:93:68:2c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'edbcn1,192.168.1.61' (RSA) to the list of known
hosts.
root@edbcn1's password:
ENCKEYS
100%  251     0.3KB/s   00:00
[oracle@c1ds1 gghome_1]$
```

```
[oracle@c1ds1 gghome 1]$ scp ENCKEYS
root@edbcn2:/u01/app/oracle/product/12.2.1/gghome_1
The authenticity of host 'edbcn2 (192.168.1.71)' can't be established.
RSA key fingerprint is cf:7c:3e:1f:21:96:01:58:6f:d2:ff:a8:93:2c:2d:c1.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'edbcn2,192.168.1.71' (RSA) to the list of known
hosts.
root@edbcn2's password:
ENCKEYS
100%  251     0.3KB/s   00:00
[oracle@c1ds1 gghome_1]$
```

```
[oracle@c1ds1 gghome 1]$ scp ENCKEYS
root@c1ds2:/u01/app/oracle/product/12.2.1/gghome_1
root@c1ds2's password:
ENCKEYS
100%  251     0.3KB/s   00:00
[oracle@c1ds1 gghome_1]$
```

```
[oracle@c1ds1 gghome_1]$ cp ENCKEYS ../gghome_2/
[oracle@c1ds1 gghome_1]$
```

On the primary node

```
[oracle@edbcn0 ~]$ cd $OGG_HOME
[oracle@edbcn0 gghome 1]$ ls -al EN*
-rw-r--r-- 1 root root 251 Jan 20 20:26 ENCKEYS
[oracle@edbcn0 gghome_1]$ su
Password:
[root@edbcn0 gghome 1]# chown oracle:oinstall ENCKEYS
[root@edbcn0 gghome_1]# exit
exit
[oracle@edbcn0 gghome_1]$ ls -al EN*
-rw-r--r-- 1 oracle oinstall 251 Jan 20 20:26 ENCKEYS
[oracle@edbcn0 gghome_1]$
```

On extended node 1

```
[oracle@edbcn1 ~]$ cd $OGG_HOME
[oracle@edbcn1 gghome_1]$ ls -al ENC*
-rw-r--r-- 1 root root 251 Jan 20 20:24 ENCKEYS
[oracle@edbcn1 gghome 1]$ su
Password:
[root@edbcn1 gghome 1]# chown oracle:oinstall ENCKEYS
[root@edbcn1 gghome_1]# exit
exit
[oracle@edbcn1 gghome_1]$ ls -al EN*
-rw-r--r-- 1 oracle oinstall 251 Jan 20 20:24 ENCKEYS
[oracle@edbcn1 gghome_1]$
```

On extended node 2

```
[oracle@edbcn2 ~]$ cd $OGG_HOME
[oracle@edbcn2 gghome_1]$ ls -al EN*
-rw-r--r-- 1 root root 251 Jan 20 20:24 ENCKEYS
[oracle@edbcn2 gghome_1]$ su
Password:
[root@edbcn2 gghome_1]# chown oracle:oinstall ENCKEYS
[root@edbcn2 gghome_1]# exit
exit
[oracle@edbcn2 gghome 1]$ ls -al ENC*
-rw-r--r-- 1 oracle oinstall 251 Jan 20 20:24 ENCKEYS
[oracle@edbcn2 gghome_1]$
```

On downstream system – c<n>ds2

```
[oracle@c1ds2 ~]$ cd $OGG HOME1
[oracle@c1ds2 gghome_1]$ ls -al EN*
-rw-r--r-- 1 root root 251 Jan 20 20:28 ENCKEYS
[oracle@c1ds2 gghome_1]$ su
Password:
[root@c1ds2 gghome_1]# chown oracle:oinstall ENCKEYS
[root@c1ds2 gghome 1]# exit
exit
[oracle@c1ds2 gghome_1]$ ls -al EN*
-rw-r--r-- 1 oracle oinstall 251 Jan 20 20:28 ENCKEYS
[oracle@c1ds2 gghome_1]$
```

Step 4: Enrypt the celar text password found in the parameter file, perform a database login to verify the database password encyryption.

Specify the algorithm used to generate the encrtypted key and the key name.

On downstream system C<n>DS1A

```
[oracle@c1ds1 gghome_1]$ ./ggsci

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 UTF-8.

Copyright (C) 1995, 2015, Oracle and/or its affiliates. All rights reserved.



GGSCI (c1ds1.oracle.com) 1> encrypt password oracle AES128 encryptkey c1ds1a
Encrypted password:
AADAAAAAAAAAAAAGAEDEFVJCJBGQDOCNJCDMIKBZEAACJLJTJPITDLGUDSBWDBDRDLIGELJQFSIPBOHPG
Algorithm used:  AES128


GGSCI (c1ds1.oracle.com) 2>
```

Testing database login using the encrypted password

```
GGSCI (c1ds1.oracle.com) 2> dblogin userid ogg_admin@c1ds1a.oracle.com, password
AADAAAAAAAAAAAAGAEDEFVJCJBGQDOCNJCDMIKBZEAACJLJTJPITDLGUDSBWDBDRDLIGELJQFSIPBOHPG
, AES128, encryptkey c1ds1a
Successfully logged into database.

GGSCI (c1ds1.oracle.com as ogg_admin@C1DS1A) 3>
```

On downstream system C<n>DS1B

```
[oracle@c1ds1 gghome_2]$ ./ggsci

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 UTF-8.

Copyright (C) 1995, 2015, Oracle and/or its affiliates. All rights reserved.



GGSCI (c1ds1.oracle.com) 1> encrypt password oracle AES128 encryptkey c1ds1b
Encrypted password:
AADAAAAAAAAAAAAGAYGUCMJMIFDSEZBKGMHQHUIBBBECDXDQEHIMJDCCGXISBYHOEBACCMJXDOFFBKBPH
Algorithm used:  AES128


GGSCI (c1ds1.oracle.com) 2>
```

Testing database login using the encrypted password

```
GGSCI (c1ds1.oracle.com as ogg admin@C1DS1A) 3> dblogin userid
ogg admin@c1ds1b.oracle.com, password
AADAAAAAAAAAAAAGAYGUCMJMIFDSEZBKGMHQHUIBBBECDXDQEHIMJDCCGXISBYHOEBACCMJXDOFFBKBPH
, AES128, encryptkey c1ds1b
Successfully logged into database.

GGSCI (c1ds1.oracle.com as ogg_admin@C1DS1B) 4>
```

On downstream system C<n>DS2A

```
[oracle@c1ds2 gghome 1]$ ./ggsci

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 UTF-8.

Copyright (C) 1995, 2015, Oracle and/or its affiliates. All rights reserved.



GGSCI (c1ds2.oracle.com) 1> encrypt password oracle AES128 encryptkey c1ds2a
```

```
Encrypted password:
AADAAAAAAAAAAAGAACRCZHOGLAQCQDYEYCMERBZGFIKDHDBHABHHVDDIRIABFJPCEDQHBCTHOGHIKCLI
Algorithm used:  AES128


GGSCI (c1ds2.oracle.com) 2>
```

Testing database login using the encrypted password

```
GGSCI (c1ds2.oracle.com) 2> dblogin userid ogg admin@c1ds2a.oracle.com, password
AADAAAAAAAAAAAGAACRCZHOGLAQCQDYEYCMERBZGFIKDHDBHABHHVDDIRIABFJPCEDQHBCTHOGHIKCLI
, AES128, encryptkey c1ds2a
Successfully logged into database.

GGSCI (c1ds2.oracle.com as ogg_admin@C1DS2A) 3>
```

On the primary node

```
[oracle@edbcn0 gghome 1]$ ./ggsci

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 UTF-8.

Copyright (C) 1995, 2015, Oracle and/or its affiliates. All rights reserved.



GGSCI (edbcn0.oracle.com) 1> encrypt password oracle AES128 encryptkey dbn0
Encrypted password:
AADAAAAAAAAAAAGAIDXCUIMEEARFOFYCUBREXFWEOAWALBOIADOGRAWGVDPAECZIKFLGBGNASHMHPFFG
Algorithm used:  AES128


GGSCI (edbcn0.oracle.com) 2>
```

Testing database login using the encrypted password

```
GGSCI (edbcn0.oracle.com) 2> dblogin userid ogg_admin@dbn0.oracle.com, password
AADAAAAAAAAAAAGAIDXCUIMEEARFOFYCUBREXFWEOAWALBOIADOGRAWGVDPAECZIKFLGBGNASHMHPFFG
, AES128, encryptkey dbn0
Successfully logged into database.

GGSCI (edbcn0.oracle.com as ogg_admin@DBN0) 3>
```

On extended node 1

```
[oracle@edbcn1 gghome_1]$ ./ggsci

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 UTF-8.

Copyright (C) 1995, 2015, Oracle and/or its affiliates. All rights reserved.



GGSCI (edbcn1.oracle.com) 1> encrypt password oracle AES128 encryptkey dbn1
Encrypted password:
AADAAAAAAAAAAAAGAXFNAMJXEFAVHAETCMFSBCDBDTAYBOISGSDDEUJJCIELDQEVEHGKFDDNCJABBZIXB
Algorithm used:  AES128


GGSCI (edbcn1.oracle.com) 2>
```

Testing database login using the encrypted password

```
GGSCI (edbcn1.oracle.com) 2> dblogin userid ogg_admin@dbn1.oracle.com, password
AADAAAAAAAAAAAAGAXFNAMJXEFAVHAETCMFSBCDBDTAYBOISGSDDEUJJCIELDQEVEHGKFDDNCJABBZIXB
, AES128, encryptkey dbn1
Successfully logged into database.

GGSCI (edbcn1.oracle.com as ogg_admin@DBN1) 3>
```

On extended node 2

```
[oracle@edbcn2 gghome 1]$ ./ggsci

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 UTF-8.

Copyright (C) 1995, 2015, Oracle and/or its affiliates. All rights reserved.



GGSCI (edbcn2.oracle.com) 1> encrypt password oracle AES128 encryptkey dbn2
Encrypted password:
AADAAAAAAAAAAAAGAMGAJRIAJBJAITCWBJCNDFJOAUGBGBDJHNDPAFCGEJGVBPAPENANETBTIFEMHQAGD
Algorithm used:  AES128


GGSCI (edbcn2.oracle.com) 2>
```

Testing database login using the encrypted password

```
GGSCI (edbcn2.oracle.com) 1> dblogin userid ogg admin@dbn2.oracle.com, password
AADAAAAAAAAAAAAGAMGAJRIAJBJAITCWBJCNDFJOAUGBGBDJHNDPAFCGEJGVBPAPENANETBTIFEMHQAGD
, AES128, encryptkey dbn2
Successfully logged into database.

GGSCI (edbcn2.oracle.com as ogg_admin@DBN2) 2>
```

Step 6: Modify the parameter files

On downstream system C<n>DS1A

```
EXTRACT EDBN001
SETENV (ORACLE_SID = C1DS1A)
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@dbn0.oracle.com, password
AADAAAAAAAAAAAGAIDXCUIMEEARFOFYCUBREXFWEOAWALBOIADOGRAWGVDPAECZIKFLGBGNASHMHPFFG
, AES128, encryptkey dbn0
tranlogoptions mininguser ogg_admin@C1DS1A.oracle.com, miningpassword
AADAAAAAAAAAAAGAEDEFVJCJBGQDOCNJCDMIKBZEAACJLJTJPITDLGUDSBWDBDRDLIGELJQFSIPBOHPG
, AES128, encryptkey c1ds1a
tranlogoptions integratedparams (max sga size 128, downstream real time mine Y)
EXTTRAIL ./dirdat/sp
--TABLE osm_repapi.*;
TRANLOGOPTIONS EXCLUDEUSER ogg_admin
-- LOGALLSUPCOLS
-- CDR Parameters
GETUPDATEBEFORES
TABLE osm repapi.customers, &
GETBEFORECOLS(ON UPDATE KEYINCLUDING (CUST_MOBILE), ON DELETE ALL);
IGNOREUPDATEBEFORES
GETUPDATEBEFORES
TABLE osm_repapi.policies, &
GETBEFORECOLS(ON UPDATE ALL, ON DELETE ALL);
IGNOREUPDATEBEFORES
```

```
EXTRACT PDBN001
SETENV (ORACLE SID = C1DS1A)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
userid ogg_admin@dbn0.oracle.com, password
AADAAAAAAAAAAAGAIDXCUIMEEARFOFYCUBREXFWEOAWALBOIADOGRAWGVDPAECZIKFLGBGNASHMHPFFG
, AES128, encryptkey dbn0
-- Parameter: COMPRESS
-- Compresses outgoing blocks of records to reduce bandwidth requirements.
-- Oracle GoldenGate decompresses the data before writing it to the trail.
-- COMPRESS typically results in compression atios of at least 4:1 and sometimes
better.
-- However, compressing data can consume CPU resources.
-- Parameter: COMPRESSTHRESHOLD
-- Sets the minimum block size for which compression is to occur. Valid values
are
-- from 0 and through 28000. The default is 1,000 bytes.
RMTHOST edbcn1.oracle.com, MGRPORT 7809, compress, compressthreshold 750
RMTTRAIL ./dirdat/tp
TABLE osm_repapi.*;
```

```
EXTRACT PDBN002
SETENV (ORACLE_SID = C1DS1A)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
```

```
userid ogg_admin@dbn0.oracle.com, password
AADAAAAAAAAAAAAGAIDXCUIMEEARFOFYCUBREXFWEOAWALBOIADOGRAWGVDPAECZIKFLGBGNASHMHPFFG
, AES128, encryptkey dbn0
RMTHOST edbcn2.oracle.com, MGRPORT 7811, compress, compressthreshold 750
RMTTRAIL ./dirdat/tp
TABLE osm_repapi.*;
```

On downstream system C<n>DS1B

```
EXTRACT EDBN101
SETENV (ORACLE SID = C1DS1B)
SETENV (ORACLE HOME = "/u01/app/oracle/product/12.1.0/dbhome 2")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
SETENV (TNS_ADMIN = "/u01/app/oracle/product/12.1.0/dbhome_1/network/admin")
userid ogg_admin@dbn1.oracle.com, password
AADAAAAAAAAAAAAGAXFNAMJXEFAVHAETCMFSBCDBDTAYBOISGSDDEUJJCIELDQEVEHGKFDDNCJABBZIXB
, AES128, encryptkey dbn1
tranlogoptions mininguser ogg_admin@C1DS1B.oracle.com, miningpassword
AADAAAAAAAAAAAAGAYGUCMJMIFDSEZBKGMHQHUIBBBECDXDQEHIMJDCCGXISBYHOEBACCMJXDOFFBKBPH
, AES128, encryptkey c1ds1b
tranlogoptions integratedparams (max_sga_size 128, downstream_real_time_mine Y)
EXTTRAIL ./dirdat/s1
--TABLE osm repapi.*;
TRANLOGOPTIONS EXCLUDEUSER ogg_admin
-- LOGALLSUPCOLS
-- CDR Parameters
GETUPDATEBEFORES
TABLE osm repapi.customers, &
-- GETBEFORECOLS(ON UPDATE KEYINCLUDING (CUST_NAME, CUST_EMAIL, CUST_MOBILE,
CUST_ADDRESS, DML_TS), ON DELETE ALL);
GETBEFORECOLS(ON UPDATE KEYINCLUDING (CUST_MOBILE), ON DELETE ALL);
IGNOREUPDATEBEFORES
GETUPDATEBEFORES
TABLE osm repapi.policies, &
-- GETBEFORECOLS(ON UPDATE KEYINCLUDING (POL_FROM, POL_TO, POL_VALUE,
POL SUB TOTAL, PT CODE, DT CODE, POL TOTAL, CUST NO, DML TS), ON DELETE ALL);
GETBEFORECOLS(ON UPDATE ALL, ON DELETE ALL);
IGNOREUPDATEBEFORES
```

```
EXTRACT PDBN101
SETENV (ORACLE_SID = C1DS1B)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome 2")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
userid ogg_admin@dbn1.oracle.com, password
AADAAAAAAAAAAAAGAXFNAMJXEFAVHAETCMFSBCDBDTAYBOISGSDDEUJJCIELDQEVEHGKFDDNCJABBZIXB
, AES128, encryptkey dbn1
RMTHOST edbcn0.oracle.com, MGRPORT 7809, compress, compressthreshold 750
RMTTRAIL ./dirdat/t1
TABLE osm_repapi.*;
```

```
EXTRACT PDBN102
SETENV (ORACLE SID = C1DS1B)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_2")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
```

```
userid ogg_admin@dbn1.oracle.com, password
AADAAAAAAAAAAAGAXFNAMJXEFAVHAETCMFSBCDBDTAYBOISGSDDEUJJCIELDQEVEHGKFDDNCJABBZIXB
, AES128, encryptkey dbn1
RMTHOST edbcn2.oracle.com, MGRPORT 7811, compress, compressthreshold 750
RMTTRAIL ./dirdat/t1
TABLE osm_repapi.*;
```

On downstream system C<n>DS2A

```
EXTRACT EDBN201
SETENV (ORACLE_SID = C1DS2A)
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@dbn2.oracle.com, password
AADAAAAAAAAAAAGAMGAJRIAJBJAITCWBJCNDFJOAUGBGBDJHNDPAFCGEJGVBPAPENANETBTIFEMHQAGD
, AES128, encryptkey dbn2
tranlogoptions mininguser ogg_admin@C1DS2A.oracle.com, miningpassword
AADAAAAAAAAAAAGAACRCZHOGLAQCQDYEYCMERBZGFIKDHDBHABHHVDDIRIABFJPCEDQHBCTHOGHIKCLI
, AES128, encryptkey c1ds2a
tranlogoptions integratedparams (max sga size 128, downstream real time mine Y)
EXTTRAIL ./dirdat/s2
--TABLE osm_repapi.*;
TRANLOGOPTIONS EXCLUDEUSER ogg_admin
-- LOGALLSUPCOLS
-- CDR Parameters
GETUPDATEBEFORES
TABLE osm_repapi.customers, &
-- GETBEFORECOLS(ON UPDATE KEYINCLUDING (CUST_NAME, CUST_EMAIL, CUST_MOBILE,
CUST_ADDRESS, DML_TS), ON DELETE ALL);
GETBEFORECOLS(ON UPDATE KEYINCLUDING (CUST_MOBILE), ON DELETE ALL);
IGNOREUPDATEBEFORES
GETUPDATEBEFORES
TABLE osm_repapi.policies, &
-- GETBEFORECOLS(ON UPDATE KEYINCLUDING (POL_FROM, POL_TO, POL_VALUE,
POL_SUB_TOTAL, PT_CODE, DT_CODE, POL_TOTAL, CUST_NO, DML_TS), ON DELETE ALL);
GETBEFORECOLS(ON UPDATE ALL, ON DELETE ALL);
IGNOREUPDATEBEFORES
```

```
EXTRACT PDBN201
SETENV (ORACLE SID = C1DS2A)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS LANG = AMERICAN AMERICA.AL32UTF8)
userid ogg_admin@dbn2.oracle.com, password
AADAAAAAAAAAAAGAMGAJRIAJBJAITCWBJCNDFJOAUGBGBDJHNDPAFCGEJGVBPAPENANETBTIFEMHQAGD
, AES128, encryptkey dbn2
RMTHOST edbcn0.oracle.com, MGRPORT 7809, compress, compressthreshold 750
RMTTRAIL ./dirdat/t2
TABLE osm_repapi.*;
```

```
EXTRACT PDBN202
SETENV (ORACLE_SID = C1DS2A)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
```

```
userid ogg_admin@dbn2.oracle.com, password
AADAAAAAAAAAAAGAMGAJRIAJBJAITCWBJCNDFJOAUGBGBDJHNDPAFCGEJGVBPAPENANETBTIFEMHQAGD
, AES128, encryptkey dbn2
RMTHOST edbcn1.oracle.com, MGRPORT 7809, compress, compressthreshold 750
RMTTRAIL ./dirdat/t2
TABLE osm_repapi.*;
```

On the primary node

```
replicat RDBN101
SETENV (ORACLE SID = DBN0)
SETENV (ORACLE HOME = "/u01/app/oracle/product/12.1.0/dbhome 1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
userid ogg_admin@dbn0.oracle.com, password
AADAAAAAAAAAAAAGAIDXCUIMEEARFOFYCUBREXFWEOAWALBOIADO
GRAWGVDPAECZIKFLGBGNASHMHPFFG, AES128, encryptkey dbn0
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn101.dsc, PURGE, megabytes 2000
-- MAP osm_repapi.*, TARGET osm_repapi.*;
MAP osm_repapi.customers, TARGET osm_repapi.customers, &
-- Implement Conflict Detection and Resolution - CDR
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
--
MAP osm_repapi.policies, TARGET osm_repapi.policies, &
-- Implement Conflict Detection and Resolution - CDR
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
```

```
replicat RDBN201
SETENV (ORACLE SID = DBN0)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
userid ogg_admin@dbn0.oracle.com, password
AADAAAAAAAAAAAAGAIDXCUIMEEARFOFYCUBREXFWEOAWALBOIADO
GRAWGVDPAECZIKFLGBGNASHMHPFFG, AES128, encryptkey dbn0
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn201.dsc, PURGE, megabytes 2000
-- MAP osm_repapi.*, TARGET osm_repapi.*;
MAP osm_repapi.customers, TARGET osm_repapi.customers, &
-- Implement Conflict Detection and Resolution - CDR
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
--
MAP osm_repapi.policies, TARGET osm_repapi.policies, &
-- Implement Conflict Detection and Resolution - CDR
```

```
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
```

On extended node 1

```
replicat RDBN001
SETENV (ORACLE_SID = DBN1)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
userid ogg_admin@dbn1.oracle.com, password  AADAAAAAAAAAAAGAXFNAMJXEFAVHAETCMFSB
CDBDTAYBOISGSDDEUJJCIELDQEVEHGKFDDNCJABBZIXB, AES128, encryptkey dbn1
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn001.dsc, PURGE, megabytes 2000
--MAP osm_repapi.*, TARGET osm_repapi.*;
MAP osm_repapi.customers, TARGET osm_repapi.customers, &
-- Implement Conflict Detection and Resolution - CDR
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
--
MAP osm repapi.policies, TARGET osm repapi.policies, &
-- Implement Conflict Detection and Resolution - CDR
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
```

```
replicat RDBN202
SETENV (ORACLE_SID = DBN1)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome 1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
userid ogg_admin@dbn1.oracle.com, password  AADAAAAAAAAAAAGAXFNAMJXEFAVHAETCMFSB
CDBDTAYBOISGSDDEUJJCIELDQEVEHGKFDDNCJABBZIXB, AES128, encryptkey dbn1
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn202.dsc, PURGE, megabytes 2000
--MAP osm repapi.*, TARGET osm repapi.*;
MAP osm_repapi.customers, TARGET osm_repapi.customers, &
-- Implement Conflict Detection and Resolution - CDR
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
--
MAP osm_repapi.policies, TARGET osm_repapi.policies, &
-- Implement Conflict Detection and Resolution - CDR
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
```

```
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
```

On extended node 2

```
replicat RDBN002
SETENV (ORACLE_SID = DBN2)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
userid ogg admin@dbn2.oracle.com, password
AADAAAAAAAAAAAGAMGAJRIAJBJAITCWBJCNDFJOAUGBGBDJHNDPAFCGE
JGVBPAPENANETBTIFEMHQAGD, AES128, encryptkey dbn2
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn002.dsc, PURGE, megabytes 2000
-- MAP osm repapi.*, TARGET osm repapi.*;
MAP osm_repapi.customers, TARGET osm_repapi.customers, &
-- Implement Conflict Detection and Resolution - CDR
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
--
MAP osm_repapi.policies, TARGET osm_repapi.policies, &
-- Implement Conflict Detection and Resolution - CDR
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
```

```
replicat RDBN102
SETENV (ORACLE_SID = DBN2)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome 1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
userid ogg admin@dbn2.oracle.com, password
AADAAAAAAAAAAAGAMGAJRIAJBJAITCWBJCNDFJOAUGBGBDJHNDPAFCGE
JGVBPAPENANETBTIFEMHQAGD, AES128, encryptkey dbn2
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn102.dsc, PURGE, megabytes 2000
-- MAP osm_repapi.*, TARGET osm_repapi.*;
MAP osm repapi.customers, TARGET osm repapi.customers, &
-- Implement Conflict Detection and Resolution - CDR
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
--
MAP osm repapi.policies, TARGET osm repapi.policies, &
-- Implement Conflict Detection and Resolution - CDR
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
```

```
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
```

Step 7: Activate the encrypted passwords settings by restarting Oracle GoldenGate 12c EDBC instances.

```
[root@edbcn0 ~]# ./stopEDBCluster.sh
```

```
[root@edbcn0 ~]# ./stopEDBCluster.sh
```

Task 4: GGSCI Commands Authentication

Oracle GoldenGate 12c GGSCI commands are secured through filtering techniques. Only authorized GGSCI commands are allowed execuction. Create the command security file CMDSEC so that only the OS dba group members have access to Oracle GoldenGate 12c instance GGSCI commands. First, grant access to all commands, to all objects, too mbers of the dba OS group. Then revoke every one else accept the preceeding rule.

On the primary mode

```
[oracle@edbcn0 gghome_1]$ vi CMDSEC
```

```
* * dba * YES
* * * * NO
[oracle@edbcn0 gghome 1]$ ls -al CMDSEC
-rw-r--r-- 1 oracle oinstall 26 Jan 22 15:41 CMDSEC
[oracle@edbcn0 gghome_1]$
```

```
GGSCI (edbcn0.oracle.com) 4> shell id

uid=54321(oracle) gid=54321(oinstall) groups=54321(oinstall),54322(dba)


GGSCI (edbcn0.oracle.com) 5> stop *

Sending STOP request to REPLICAT RDBN101 ...
Request processed.

Sending STOP request to REPLICAT RDBN201 ...
Request processed.


GGSCI (edbcn0.oracle.com) 6> start *

Sending START request to MANAGER ...
REPLICAT RDBN101 starting

Sending START request to MANAGER ...
REPLICAT RDBN201 starting
```

```
GGSCI (edbcn0.oracle.com) 8> info all

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RDBN101     00:00:00      00:00:07
REPLICAT    RUNNING     RDBN201     00:00:00      00:00:06


GGSCI (edbcn0.oracle.com) 9>
```

On extended node 1

```
[oracle@edbcn1 gghome_1]$ vi CMDSEC
```

```
* * dba * YES
* * * * NO
```

On extended node 2

```
[oracle@edbcn2 gghome_1]$ vi CMDSEC
```

```
* * dba * YES
* * * * NO
```

On downstream system C<n>DS1A

```
[oracle@c1ds1 gghome_1]$ vi CMDSEC
```

```
* * dba * YES
* * * * NO
```

On downstream system C<n>DS1B

```
[oracle@c1ds1 gghome_2]$ . oraenv
ORACLE_SID = [C1DS1A] ? C1DS1B
The Oracle base remains unchanged with value /u01/app/oracle
[oracle@c1ds1 gghome_2]$ echo $ORACLE_HOME
/u01/app/oracle/product/12.1.0/dbhome_2
[oracle@c1ds1 gghome_2]$ cd $OGG_HOME2
[oracle@c1ds1 gghome_2]$ pwd
/u01/app/oracle/product/12.2.1/gghome_2
[oracle@c1ds1 gghome_2]$
```

```
[oracle@c1ds1 gghome_1]$ vi CMDSEC
```

```
* * dba * YES
* * * * NO
```

On downstream system C<n>DS2A

```
[oracle@c1ds2 gghome_1]$ vi CMDSEC
```

```
* * dba * YES
* * * * NO
```

Task 5: Configure Encrypted Trail Data and Data Streaming

This task encrypts data in trails and TCP/IP network. The ENCKEY file used for password encryption is used to for data and TCP/IP network encrytption. However; When applying the transation on the primary node, extended node 1 and extended node 2, the encryption key of the originated data pump is used. Figure 16 highlight the Replicats encryption keys.

The process of applying encryption is as follows:

1. Encrypt data generated by the integrated extract
2. Decrypt trail used by the data pump
3. Encrypt TCP/IP network
4. Encrypt trail before sending by the data pump
5. Decrypt trail for the Replicat

The encryption key file:

```
c1ds1a  0x26719158BEA89920C790302244912538

c1ds1b  0x3FFFDB03A969F3299EAD167DAC7FB178

c1ds2a  0xADE22B2F2247346CCC58700C9A0A9D1B

dbn0    0x1BC67B5A9C24752EF903CA1B8795883E

dbn1    0x8AA9CB051502B67026AF232B74207461

dbn2    0xF88C1B318FDFF632535A7D3A62AB5F04
```

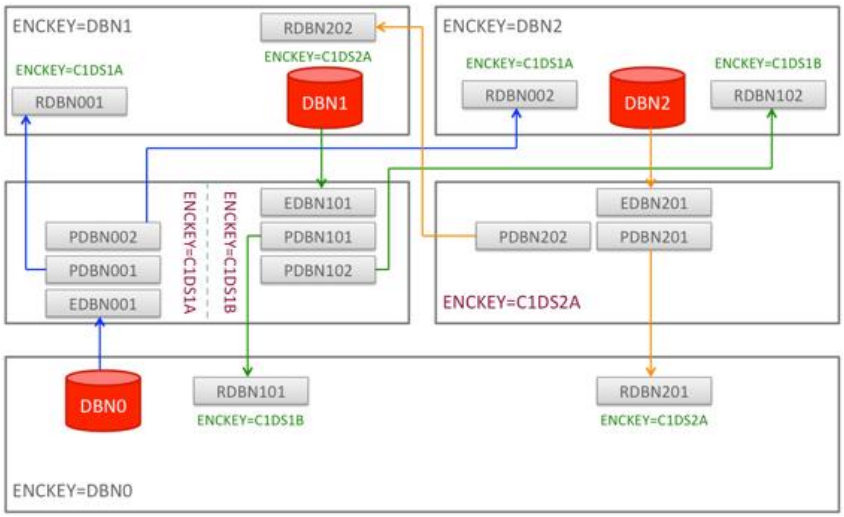


Figure 16: Replicat DECRYPTRAIL Encryption Keys

Add the encryption parameter ENCRYPTTRAIL followd by the AES algorithm and the keyname.

Encryption of the Primary Node

On downstream system C1DS1A

```
EXTRACT EDBN001
SETENV (ORACLE SID = C1DS1A)
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@DBN0.oracle.com, PASSWORD oracle
userid ogg_admin@dbn0.oracle.com, password
AADAAAAAAAAAAAAGAIDXCUIMEEARFOFYCUBREXFWEOAWALBOIADOGRAWGVDPAECZIKFLG
BGNASHMHPFFG, AES128, encryptkey dbn0
-- tranlogoptions mininguser ogg_admin@C1DS1A.oracle.com, miningpassword oracle
tranlogoptions mininguser ogg_admin@C1DS1A.oracle.com, miningpassword
AADAAAAAAAAAAAAGAEDEFVJCJBGQDOCNJCDMIKBZEA
ACJLJTJPITDLGUDSBWDBDRDLIGELJQFSIPBOHPG, AES128, encryptkey c1ds1a
tranlogoptions integratedparams (max sga size 128, downstream real time mine Y)
ENCRYPTTRAIL AES128 KEYNAME c1ds1a
EXTTRAIL ./dirdat/sp
--TABLE osm_repapi.*;
TRANLOGOPTIONS EXCLUDEUSER ogg admin
-- LOGALLSUPCOLS
-- CDR Parameters
GETUPDATEBEFORES
TABLE osm_repapi.customers, &
GETBEFORECOLS(ON UPDATE KEYINCLUDING (CUST_MOBILE), ON DELETE ALL);
IGNOREUPDATEBEFORES
```

```
GETUPDATEBEFORES
TABLE osm_repapi.policies, &
GETBEFORECOLS(ON UPDATE ALL, ON DELETE ALL);
IGNOREUPDATEBEFORES
```

Decrypt, encrypt TCP/IP network and encrypt remote trail

```
EXTRACT PDBN001
SETENV (ORACLE_SID = C1DS1A)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
--USERID ogg_admin@dbn0.oracle.com, PASSWORD oracle
userid ogg_admin@dbn0.oracle.com, password
AADAAAAAAAAAAAGAIDXCUIMEEARFOFYCUBREXFWEOAWALBOIADOGRAWGVDPAECZIKFLG
BGNASHMHPFFG, AES128, encryptkey dbn0
DECRYPTTRAIL AES128 KEYNAME c1ds1a
-- Parameter: COMPRESS
-- Compresses outgoing blocks of records to reduce bandwidth requirements.
-- Oracle GoldenGate decompresses the data before writing it to the trail.
-- COMPRESS typically results in compression atios of at least 4:1 and sometimes
better.
-- However, compressing data can consume CPU resources.
-- Parameter: COMPRESSTHRESHOLD
-- Sets the minimum block size for which compression is to occur. Valid values
are
-- from 0 and through 28000. The default is 1,000 bytes.
RMTHOST edbcn1.oracle.com, MGRPORT 7809, compress, compressthreshold 750
RMTHOSTOPTIONS ENCRYPT AES128 KEYNAME c1ds1a
ENCRYPTTRAIL AES128 KEYNAME c1ds1a
RMTTRAIL ./dirdat/tp
TABLE osm_repapi.*;
```

```
EXTRACT PDBN002
SETENV (ORACLE_SID = C1DS1A)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
--USERID ogg_admin@dbn0.oracle.com, PASSWORD oracle
userid ogg_admin@dbn0.oracle.com, password
AADAAAAAAAAAAAGAIDXCUIMEEARFOFYCUBREXFWEOAWALBOIADOGRAWGVDPAECZIKFLG
BGNASHMHPFFG, AES128, encryptkey dbn0
DECRYPTTRAIL AES128 KEYNAME c1ds1a
RMTHOST edbcn2.oracle.com, MGRPORT 7811, compress, compressthreshold 750
-- encrypt data sent across tcp/ip
RMTHOSTOPTIONS ENCRYPT AES128 KEYNAME c1ds1a
ECRYPTTRAIL AES128 KEYNAME c1ds1a
RMTTRAIL ./dirdat/tp
TABLE osm_repapi.*;
```

Decrypt using the data pump encryption key

On extended node 1

```
replicat RDBN001
SETENV (ORACLE_SID = DBN1)
```

```
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
--USERID ogg admin@dbn1.oracle.com, PASSWORD oracle
userid ogg_admin@dbn1.oracle.com, password  AADAAAAAAAAAAAGAXFNAMJXEFAVHAETCMFSB
CDBDTAYBOISGSDDEUJJCIELDQEVEHGKFDDNCJABBZIXB, AES128, encryptkey dbn1
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn001.dsc, PURGE, megabytes 2000
--MAP osm_repapi.*, TARGET osm_repapi.*;
DECRYPTTRAIL AES128 KEYNAME c1ds1a
MAP osm_repapi.customers, TARGET osm_repapi.customers, &
-- Implement Conflict Detection and Resolution - CDR
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
--
MAP osm_repapi.policies, TARGET osm_repapi.policies, &
-- Implement Conflict Detection and Resolution - CDR
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
```

On extended node 2

```
replicat RDBN002
SETENV (ORACLE_SID = DBN2)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS LANG = AMERICAN AMERICA.AL32UTF8)
-- USERID ogg_admin@dbn2.oracle.com, PASSWORD oracle
userid ogg admin@dbn2.oracle.com, password
AADAAAAAAAAAAAGAMGAJRIAJBJAITCWBJCNDFJOAUGBGBDJHNDPAFCGEJGVBP
APENANETBTIFEMHQAGD, AES128, encryptkey dbn2
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn002.dsc, PURGE, megabytes 2000
-- MAP osm repapi.*, TARGET osm repapi.*;
DECRYPTTRAIL AES128 KEYNAME c1ds1a
MAP osm_repapi.customers, TARGET osm_repapi.customers, &
-- Implement Conflict Detection and Resolution - CDR
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
--
MAP osm_repapi.policies, TARGET osm_repapi.policies, &
-- Implement Conflict Detection and Resolution - CDR
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
```

Encryption of Extended Node 1

On downstream system C1DS1B

```
EXTRACT EDBN101
SETENV (ORACLE_SID = C1DS1B)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_2")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
SETENV (TNS_ADMIN = "/u01/app/oracle/product/12.1.0/dbhome_1/network/admin")
-- USERID ogg_admin@DBN1.oracle.com, PASSWORD oracle
userid ogg admin@dbn1.oracle.com, password
AADAAAAAAAAAAAAGAXFNAMJXEFAVHAETCMFSBCDBDTAYBOISGSDDEUJJCIELDQEVEHGKFDDNCJABBZIXB
, AES128, encryptkey dbn1
--tranlogoptions mininguser ogg_admin@C1DS1B.oracle.com, miningpassword oracle
tranlogoptions mininguser ogg_admin@C1DS1B.oracle.com, miningpassword
AADAAAAAAAAAAAAGAYGUCMJMIFDSEZBKGMHQHUIBBBECDXDQEHIMJDCCGXISBYHOEBACCMJXDOFFBKBPH
, AES128, encryptkey c1ds1b
tranlogoptions integratedparams (max_sga_size 128, downstream_real_time_mine Y)
ENCRYPTTRAIL AES128 KEYNAME c1ds1b
EXTTRAIL ./dirdat/s1
--TABLE osm_repapi.*;
TRANLOGOPTIONS EXCLUDEUSER ogg_admin
-- LOGALLSUPCOLS
-- CDR Parameters
GETUPDATEBEFORES
TABLE osm_repapi.customers, &
-- GETBEFORECOLS(ON UPDATE KEYINCLUDING (CUST_NAME, CUST_EMAIL, CUST_MOBILE,
CUST_ADDRESS, DML_TS), ON DELETE ALL);
GETBEFORECOLS(ON UPDATE KEYINCLUDING (CUST_MOBILE), ON DELETE ALL);
IGNOREUPDATEBEFORES
GETUPDATEBEFORES
TABLE osm_repapi.policies, &
-- GETBEFORECOLS(ON UPDATE KEYINCLUDING (POL_FROM, POL_TO, POL_VALUE,
POL SUB TOTAL, PT CODE, DT CODE, POL TOTAL, CUST NO, DML TS), ON DELETE ALL);
GETBEFORECOLS(ON UPDATE ALL, ON DELETE ALL);
IGNOREUPDATEBEFORES
```

```
EXTRACT PDBN101
SETENV (ORACLE_SID = C1DS1B)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_2")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
--USERID ogg admin@dbn1.oracle.com, PASSWORD oracle
userid ogg_admin@dbn1.oracle.com, password
AADAAAAAAAAAAAAGAXFNAMJXEFAVHAETCMFSBCD
BDTAYBOISGSDDEUJJCIELDQEVEHGKFDDNCJABBZIXB, AES128, encryptkey dbn1
DECRYPTTRAIL AES128 KEYNAME c1ds1b
RMTHOST edbcn0.oracle.com, MGRPORT 7809, compress, compressthreshold 750
RMTHOSTOPTIONS ENCRYPT AES128 KEYNAME c1ds1b
ENCRYPTTRAIL AES128 KEYNAME c1ds1b
RMTTRAIL ./dirdat/t1
TABLE osm_repapi.*;
```

```
EXTRACT PDBN102
SETENV (ORACLE_SID = C1DS1B)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_2")
```

```
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
--USERID ogg_admin@dbn1.oracle.com, PASSWORD oracle
userid ogg admin@dbn1.oracle.com, password
AADAAAAAAAAAAAGAXFNAMJXEFAVHAETCMFSBCD
BDTAYBOISGSDDEUJJCIELDQEVEHGKFDDNCJABBZIXB, AES128, encryptkey dbn1
DECRYPTTRAIL AES128 KEYNAME c1ds1b
RMTHOST edbcn2.oracle.com, MGRPORT 7811, compress, compressthreshold 750
RMTHOSTOPTIONS ENCRYPT AES128 KEYNAME c1ds1b
ENCRYPTTRAIL AES128 KEYNAME c1ds1b
RMTTRAIL ./dirdat/t1
TABLE osm_repapi.*;
```

On the primary node

```
replicat RDBN101
SETENV (ORACLE_SID = DBN0)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
--USERID ogg_admin@dbn0.oracle.com, PASSWORD oracle
userid ogg_admin@dbn0.oracle.com, password
AADAAAAAAAAAAAGAIDXCUIMEEARFOFYCUBREXFWEOAWALBOIADO
GRAWGVDPAECZIKFLGBGNASHMHPFFG, AES128, encryptkey dbn0
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn101.dsc, PURGE, megabytes 2000
-- MAP osm_repapi.*, TARGET osm_repapi.*;
DECRYPTTRAIL AES128 KEYNAME c1ds1b
MAP osm_repapi.customers, TARGET osm_repapi.customers, &
-- Implement Conflict Detection and Resolution - CDR
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
--
MAP osm_repapi.policies, TARGET osm_repapi.policies, &
-- Implement Conflict Detection and Resolution - CDR
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
```

Encryption of Extended Node 2

On downstream system C1DS2A

```
EXTRACT EDBN201
SETENV (ORACLE_SID = C1DS2A)
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@DBN2.oracle.com, PASSWORD oracle
```

```
userid ogg_admin@dbn2.oracle.com, password
AADAAAAAAAAAAAGAMGAJRIAJBJAITCWBJCNDFJOAUGBGBDJHNDPAFCGEJGVBPAPENANETBTIFEMHQAGD
, AES128, encryptkey dbn2
-- tranlogoptions mininguser ogg_admin@C1DS2A.oracle.com, miningpassword oracle
tranlogoptions mininguser ogg_admin@C1DS2A.oracle.com, miningpassword
AADAAAAAAAAAAAGAACRCZHOGLAQCQDYEYCMERBZGFIKDHDBHABHHVDDIRIABFJPCEDQHBCTHOGHIKCLI
, AES
128, encryptkey c1ds2a
tranlogoptions integratedparams (max sga size 128, downstream real time mine Y)
ENCRYPTTRAIL AES128 KEYNAME c1ds2a
EXTTRAIL ./dirdat/s2
--TABLE osm_repapi.*;
TRANLOGOPTIONS EXCLUDEUSER ogg_admin
-- LOGALLSUPCOLS
-- CDR Parameters
GETUPDATEBEFORES
TABLE osm_repapi.customers, &
-- GETBEFORECOLS(ON UPDATE KEYINCLUDING (CUST_NAME, CUST_EMAIL, CUST_MOBILE,
CUST_ADDRESS, DML_TS), ON DELETE ALL);
GETBEFORECOLS(ON UPDATE KEYINCLUDING (CUST_MOBILE), ON DELETE ALL);
IGNOREUPDATEBEFORES
GETUPDATEBEFORES
TABLE osm repapi.policies, &
-- GETBEFORECOLS(ON UPDATE KEYINCLUDING (POL_FROM, POL_TO, POL_VALUE,
POL_SUB_TOTAL, PT_CODE, DT_CODE, POL_TOTAL, CUST_NO, DML_TS), ON DELETE ALL);
GETBEFORECOLS(ON UPDATE ALL, ON DELETE ALL);
IGNOREUPDATEBEFORES
```

```
EXTRACT PDBN201
SETENV (ORACLE SID = C1DS2A)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
-- USERID ogg_admin@dbn2.oracle.com, PASSWORD oracle
userid ogg admin@dbn2.oracle.com, password
AADAAAAAAAAAAAGAMGAJRIAJBJAITCWBJCNDFJOAUGBGBDJHNDPAFCGEJGVBPAPENANETBTIFEMHQAGD
, AES128, encryptkey dbn2
DECRYPTTRAIL AES128 KEYNAME c1ds2a
RMTHOST edbcn0.oracle.com, MGRPORT 7809, compress, compressthreshold 750
RMTHOSTOPTIONS ENCRYPT AES128 KEYNAME c1ds2a
ENCRYPTTRAIL AES128 KEYNAME c1ds2a
RMTTRAIL ./dirdat/t2
TABLE osm_repapi.*;
```

```
EXTRACT PDBN202
SETENV (ORACLE_SID = C1DS2A)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS LANG = AMERICAN AMERICA.AL32UTF8)
--USERID ogg admin@dbn2.oracle.com, PASSWORD oracle
userid ogg_admin@dbn2.oracle.com, password
AADAAAAAAAAAAAGAMGAJRIAJBJAITCWBJCNDFJOAUGBGBDJHNDPAFCGEJGVBPAPENANETBTIFEMHQAGD
, AES128, encryptkey dbn2
DECRYPTTRAIL AES128 KEYNAME c1ds2a
RMTHOST edbcn1.oracle.com, MGRPORT 7809, compress, compressthreshold 750
RMTHOSTOPTIONS ENCRYPT AES128 KEYNAME c1ds2a
ENCRYPTTRAIL AES128 KEYNAME c1ds2a
RMTTRAIL ./dirdat/t2
TABLE osm_repapi.*;
```

On the Primary node

```
replicat RDBN201
SETENV (ORACLE SID = DBN0)
SETENV (ORACLE_HOME = "/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG = AMERICAN_AMERICA.AL32UTF8)
--USERID ogg_admin@dbn0.oracle.com, PASSWORD oracle
userid ogg admin@dbn0.oracle.com, password
AADAAAAAAAAAAAAGAIDXCUIMEEARFOFYCUBREXFWEOAWALBOIADOGRAWGVDPAECZIKFLGBGNASHMHPFFG
, AES128, encryptkey dbn0
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rdbn201.dsc, PURGE, megabytes 2000
-- MAP osm_repapi.*, TARGET osm_repapi.*;
DECRYPTTRAIL AES128 KEYNAME c1ds2a
MAP osm_repapi.customers, TARGET osm_repapi.customers, &
-- Implement Conflict Detection and Resolution - CDR
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
--
MAP osm_repapi.policies, TARGET osm_repapi.policies, &
-- Implement Conflict Detection and Resolution - CDR
   RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX(DML_TS))), &
   RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX(DML TS))), &
   RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
   RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
```

Task 6: Configure Secure Manager

The required parameter for the Manager process is the port number. However; other parameters settings help generate statistic/report, autostartup processes and perform trail purge. Additionaly, configure secure manager process enhance the security of Oracle GoldenGate 12c instances.

On the Primary node

```
PORT 7809
ACCESSRULE, PROG *, IPADDR 192.168.1.51, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.61, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.71, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.31, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.41, ALLOW
PURGEOLDEXTRACTS ./dirdat/t*, USECHECKPOINTS, MINKEEPHOURS 2
```

On extended node 1

```
PORT 7809
ACCESSRULE, PROG *, IPADDR 192.168.1.51, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.61, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.71, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.31, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.41, ALLOW
PURGEOLDEXTRACTS ./dirdat/t*, USECHECKPOINTS, MINKEEPHOURS 2
```

On extended node 2

```
PORT 7811
ACCESSRULE, PROG *, IPADDR 192.168.1.51, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.61, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.71, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.31, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.41, ALLOW
PURGEOLDEXTRACTS ./dirdat/t*, USECHECKPOINTS, MINKEEPHOURS 2
```

On downstream system C<n>DS1A

```
PORT 7809
ACCESSRULE, PROG *, IPADDR 192.168.1.51, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.61, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.71, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.31, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.41, ALLOW
PURGEOLDEXTRACTS ./dirdat/s*, USECHECKPOINTS, MINKEEPHOURS 2
```

On downstream system C<n>DS1B

```
PORT 7810
ACCESSRULE, PROG *, IPADDR 192.168.1.51, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.61, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.71, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.31, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.41, ALLOW
PURGEOLDEXTRACTS ./dirdat/s*, USECHECKPOINTS, MINKEEPHOURS 2
```

On downstream system C<n>DS2A

```
PORT 7809
ACCESSRULE, PROG *, IPADDR 192.168.1.51, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.61, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.71, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.31, ALLOW
ACCESSRULE, PROG *, IPADDR 192.168.1.41, ALLOW
PURGEOLDEXTRACTS ./dirdat/s*, USECHECKPOINTS, MINKEEPHOURS 2
```

Task 7: Configure EDBC Encrypted Backup

This tasks re-develop RMAN backup by encrypting the backup set generated.

1. Configure RMAN for encrypted backup
2. Update SQLNET.ORA for wallet location
3. Create the wallet
4. Test encrypted RMAN Backup
5. Update stored script backup execute RMAN command

On the Primary node

Configure RMAN for encrypted backup

```
[oracle@edbcn0 ~]$ rman

Recovery Manager: Release 12.1.0.2.0 - Production on Mon Jan 23 00:58:55 2017

Copyright (c) 1982, 2014, Oracle and/or its affiliates.  All rights reserved.

RMAN> connect target

connected to target database: DBN0 (DBID=1344971290)

RMAN>
```

```
RMAN> show all;

RMAN configuration parameters for database with db_unique_name DBN0 are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP ON;
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; # default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE OFF;
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE 'DEFAULT' OPTIMIZE FOR
LOAD TRUE ; # default
CONFIGURE RMAN OUTPUT TO KEEP FOR 7 DAYS; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO
'/u01/app/oracle/product/12.1.0/dbhome_1/dbs/snapcf_DBN0.f'; # default
```

```
RMAN> CONFIGURE ENCRYPTION FOR DATABASE ON;

old RMAN configuration parameters:
CONFIGURE ENCRYPTION FOR DATABASE OFF;
new RMAN configuration parameters:
CONFIGURE ENCRYPTION FOR DATABASE ON;
new RMAN configuration parameters are successfully stored

RMAN>
```

Update/create SQLNET.ORA for wallet location

```
[oracle@edbcn0 ~]$ cd $ORACLE HOME/network/admin
[oracle@edbcn0 admin]$ cat sqlnet.ora
# setting the encryption parameters
sqlnet.crypto_seed = "kclabefmnoc"
sqlnet.encryption server = accepted
sqlnet.encryption_client = required
sqlnet.encryption types client = AES128
sqlnet.encryption_types_server = AES128
trace_directory_server=/u01/app/oracle/product/12.1.0/dbhome_1/network/log
trace_level_client=16
trace level server=16
#
ENCRYPTION WALLET LOCATION =
(SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
        (DIRECTORY = /u01/app/oracle/admin/DBN0/wallets)
    )
)
[oracle@edbcn0 admin]$
```

```
[oracle@edbcn0 admin]$ mkdir -p /u01/app/oracle/admin/DBN0/wallets
[oracle@edbcn0 admin]$
```

Create the wallet and add certificate, Save as to /u01/app/oracle/admin/DBN0/wallet

```
[oracle@edbcn0 admin]$ cd $ORACLE_HOME/bin
[oracle@edbcn0 bin]$ ./owm
```



Test encrypted RMAN Backup

```
RMAN> set encryption on identified by 'oracle 4U' only;

executing command: SET encryption

RMAN> backup tablespace users;

Starting backup at 23-JAN-17
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=47 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00006
name=/u01/app/oracle/oradata/DBN0/datafile/o1_mf_users_d0mopbdo_.dbf
channel ORA_DISK_1: starting piece 1 at 23-JAN-17
channel ORA_DISK_1: finished piece 1 at 23-JAN-17
piece
handle=/u01/app/oracle/fast_recovery_area/DBN0/backupset/2017_01_23/o1_mf_nnndf_
TAG20170123T022859_d8bj1vw2_.bkp tag=TAG20170123T022859 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:07
Finished backup at 23-JAN-17

Starting Control File and SPFILE Autobackup at 23-JAN-17
piece
handle=/u01/app/oracle/fast_recovery_area/DBN0/autobackup/2017_01_23/o1_mf_s_933
992947_d8bj244j_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 23-JAN-17

RMAN>
```

Update stored script backup execute RMAN command

```
[oracle@edbcn0 ~]$ pwd
/home/oracle
[oracle@edbcn0 ~]$ vi DBN0_BCKUP.rcv
```

```
set encryption on identified by 'oracle_4U' only;
run {execute script DBN0_BCK;}
```

On extended node 1

Configure RMAN for encrypted backup

```
[oracle@edbcn1 ~]$ rman

Recovery Manager: Release 12.1.0.2.0 - Production on Mon Jan 23 02:46:41 2017

Copyright (c) 1982, 2014, Oracle and/or its affiliates.  All rights reserved.

RMAN> connect target

connected to target database: DBN1 (DBID=1383723292)

RMAN>
```

```
RMAN> show all;

using target database control file instead of recovery catalog
RMAN configuration parameters for database with db unique name DBN1 are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP OFF; # default
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; # default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE OFF; # default
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE 'DEFAULT' OPTIMIZE FOR
LOAD TRUE ; # default
CONFIGURE RMAN OUTPUT TO KEEP FOR 7 DAYS; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO
'/u01/app/oracle/product/12.1.0/dbhome 1/dbs/snapcf DBN1.f'; # default

RMAN>
```

```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP ON;

new RMAN configuration parameters:
```

```
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters are successfully stored

RMAN> CONFIGURE ENCRYPTION FOR DATABASE ON;

new RMAN configuration parameters:
CONFIGURE ENCRYPTION FOR DATABASE ON;
new RMAN configuration parameters are successfully stored

RMAN>
```

Update/create SQLNET.ORA for wallet location

```
# setting the encryption parameters for redo-transport
sqlnet.crypto_seed = "kclabefmnoc"
sqlnet.encryption_server = accepted
sqlnet.encryption client = required
sqlnet.encryption_types_client = AES128
sqlnet.encryption types server = AES128
trace_directory_server=/u01/app/oracle/product/12.1.0/dbhome_1/network/log
trace_level_client=16
trace_level_server=16
# setting wallet location
ENCRYPTION_WALLET_LOCATION =
(SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
        (DIRECTORY = /u01/app/oracle/admin/DBN0/wallets/)
    )
)
```

```
[oracle@edbcn1 wallets]$ mkdir -p /u01/app/oracle/admin/DBN1/wallets
[oracle@edbcn1 wallets]$ cd /u01/app/oracle/admin/DBN1/wallets
[oracle@edbcn1 wallets]$ ls
[oracle@edbcn1 wallets]$ pwd
/u01/app/oracle/admin/DBN1/wallets
[oracle@edbcn1 wallets]$
```

Create the wallet and add certificate, Save as to /u01/app/oracle/admin/DBN1/wallet

```
[oracle@edbcn0 admin]$ export DISPLAY=192.168.1.200:0.0
[oracle@edbcn0 admin]$ cd $ORACLE_HOME/bin
[oracle@edbcn0 bin]$ ./owm
```



Test encrypted RMAN Backup

```
RMAN> connect target

connected to target database: DBN1 (DBID=1383723292)

RMAN> set encryption on identified by 'oracle_4U' only;

executing command: SET encryption
using target database control file instead of recovery catalog

RMAN> backup tablespace users;

Starting backup at 23-JAN-17
allocated channel: ORA DISK 1
channel ORA_DISK_1: SID=69 device type=DISK
channel ORA DISK 1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00006
name=/u01/app/oracle/oradata/DBN1/datafile/o1_mf_users_d0mq8hv2_.dbf
channel ORA DISK 1: starting piece 1 at 23-JAN-17
channel ORA_DISK_1: finished piece 1 at 23-JAN-17
piece
handle=/u01/app/oracle/fast_recovery_area/DBN1/backupset/2017_01_23/o1_mf_nnndf_
TAG20170123T025949_d8bkvpjp_.bkp tag=TAG20170123T025949 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:07
Finished backup at 23-JAN-17

Starting Control File and SPFILE Autobackup at 23-JAN-17
```

```
piece
handle=/u01/app/oracle/fast_recovery_area/DBN1/autobackup/2017_01_23/o1_mf_s_933
994797 d8bkvxz0 .bkp comment=NONE
Finished Control File and SPFILE Autobackup at 23-JAN-17

RMAN>
```

Update stored script backup execute RMAN command

```
[oracle@edbcn0 ~]$ vi DBN1_BCKUP.rcv
```

```
set encryption on identified by 'oracle 4U' only;
run {execute script DBN1_BCK;}
```

On extended node 2

Configure RMAN for encrypted backup

```
[oracle@edbcn2 ~]$ rman

Recovery Manager: Release 12.1.0.2.0 - Production on Mon Jan 23 03:02:35 2017

Copyright (c) 1982, 2014, Oracle and/or its affiliates.  All rights reserved.

RMAN> connect target

connected to target database: DBN2 (DBID=1602423061)

RMAN>
```

```
RMAN> show all;

using target database control file instead of recovery catalog
RMAN configuration parameters for database with db unique name DBN2 are:
CONFIGURE RETENTION POLICY TO REDUNDANCY 1; # default
CONFIGURE BACKUP OPTIMIZATION OFF; # default
CONFIGURE DEFAULT DEVICE TYPE TO DISK; # default
CONFIGURE CONTROLFILE AUTOBACKUP OFF; # default
CONFIGURE CONTROLFILE AUTOBACKUP FORMAT FOR DEVICE TYPE DISK TO '%F'; # default
CONFIGURE DEVICE TYPE DISK PARALLELISM 1 BACKUP TYPE TO BACKUPSET; # default
CONFIGURE DATAFILE BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE ARCHIVELOG BACKUP COPIES FOR DEVICE TYPE DISK TO 1; # default
CONFIGURE MAXSETSIZE TO UNLIMITED; # default
CONFIGURE ENCRYPTION FOR DATABASE OFF; # default
CONFIGURE ENCRYPTION ALGORITHM 'AES128'; # default
CONFIGURE COMPRESSION ALGORITHM 'BASIC' AS OF RELEASE 'DEFAULT' OPTIMIZE FOR
LOAD TRUE ; # default
CONFIGURE RMAN OUTPUT TO KEEP FOR 7 DAYS; # default
CONFIGURE ARCHIVELOG DELETION POLICY TO NONE; # default
CONFIGURE SNAPSHOT CONTROLFILE NAME TO
'/u01/app/oracle/product/12.1.0/dbhome 1/dbs/snapcf DBN2.f'; # default

RMAN>
```

```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP ON;

new RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters are successfully stored

RMAN> CONFIGURE ENCRYPTION FOR DATABASE ON;

new RMAN configuration parameters:
CONFIGURE ENCRYPTION FOR DATABASE ON;
new RMAN configuration parameters are successfully stored

RMAN>
```

Update/create SQLNET.ORA for wallet location

```
# setting the encryption parameters for redo-transport
sqlnet.crypto seed = "kclabefmnoc"
sqlnet.encryption_server = accepted
sqlnet.encryption client = required
sqlnet.encryption_types_client = AES128
sqlnet.encryption_types_server = AES128
trace_directory_server=/u01/app/oracle/product/12.1.0/dbhome_1/network/log
trace level client=16
trace level server=16
# setting wallet location
ENCRYPTION_WALLET_LOCATION =
(SOURCE =
    (METHOD = FILE)
    (METHOD DATA =
        (DIRECTORY = /u01/app/oracle/admin/DBN2/wallets/)
    )
)
```

```
[oracle@edbcn2 ~]$ mkdir -p /u01/app/oracle/admin/DBN2/wallets
[oracle@edbcn2 ~]$ cd /u01/app/oracle/admin/DBN2/wallets
[oracle@edbcn2 wallets]$ pwd
/u01/app/oracle/admin/DBN2/wallets
[oracle@edbcn2 wallets]$ ls
[oracle@edbcn2 wallets]$
```

Create the wallet and add certificate, Save as to /u01/app/oracle/admin/DBN2/wallet

```
[oracle@edbcn2 wallets]$ export DISPLAY=192.168.1.200:0.0
[oracle@edbcn2 wallets]$ cd $ORACLE_HOME/bin
[oracle@edbcn2 bin]$ ./owm
```



Test encrypted RMAN Backup

```
RMAN> connect target

connected to target database: DBN1 (DBID=1383723292)

RMAN> set encryption on identified by 'oracle_4U' only;

executing command: SET encryption

RMAN> backup tablespace users;

Starting backup at 23-JAN-17
using channel ORA_DISK_1

channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00004 name=/u01/app/oracle/oradata/DBN2/users01.dbf
channel ORA_DISK_1: starting piece 1 at 23-JAN-17
channel ORA_DISK_1: finished piece 1 at 23-JAN-17
piece
handle=/u01/app/oracle/fast_recovery_area/DBN2/backupset/2017_01_23/o1_mf_nnndf_
TAG20170123T055933 d8bwdrb7 .bkp tag=TAG20170123T055933 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:15
Finished backup at 23-JAN-17

Starting Control File and SPFILE Autobackup at 23-JAN-17
piece
handle=/u01/app/oracle/fast_recovery_area/DBN2/autobackup/2017_01_23/o1_mf_s_934
005591_d8bwf9ks_.bkp comment=NONE
```

```
Finished Control File and SPFILE Autobackup at 23-JAN-17

RMAN>
```

Update stored script backup execute RMAN command

```
[oracle@edbcn0 ~]$ vi DBN2_BCKUP.rcv
```

```
set encryption on identified by 'oracle_4U' only;
run {execute script DBN2
```

Peforming Database Encrypted RMAN Backup

```
[root@edbcn0 ~]# cat backupEDBC.sh
#!/bin/sh
echo "Backing up Database DBN1..."
su - oracle /home/oracle/DBN1 BCKUP.sh
echo "Backing up Database DBN2..."
su - oracle /home/oracle/DBN2_BCKUP.sh
echo "Backing up Database DBN0..."
su - oracle /home/oracle/DBN0_BCKUP.sh

[root@edbcn0 ~]# ./backupEDBC.sh
Backing up Database DBN1...
RMAN> 2> 3>
```

RMAN backup logfile for DBN1…

```
Recovery Manager: Release 12.1.0.2.0 - Production on Mon Jan 23 06:12:55 2017

Copyright (c) 1982, 2014, Oracle and/or its affiliates.  All rights reserved.

connected to target database: DBN1 (DBID=1383723292)
connected to recovery catalog database

RMAN> set encryption on identified by * only;
2> run {execute script DBN1_BCK;}
3>
executing command: SET encryption
starting full resync of recovery catalog
full resync complete

executing script: DBN1_BCK

allocated channel: c1
channel c1: SID=70 device type=DISK

allocated channel: c2
channel c2: SID=69 device type=DISK

allocated channel: c3
channel c3: SID=68 device type=DISK

Starting backup at 23-JAN-17
```

```
channel c1: starting full datafile backup set
channel c1: specifying datafile(s) in backup set
input datafile file number=00003
name=/u01/app/oracle/oradata/DBN1/datafile/o1_mf_sysaux_d0mq5ccn_.dbf
channel c1: starting piece 1 at 23-JAN-17
channel c2: starting full datafile backup set
channel c2: specifying datafile(s) in backup set
input datafile file number=00001
name=/u01/app/oracle/oradata/DBN1/datafile/o1 mf system d0mq6rnw .dbf
input datafile file number=00006
name=/u01/app/oracle/oradata/DBN1/datafile/o1 mf users d0mq8hv2 .dbf
channel c2: starting piece 1 at 23-JAN-17
channel c3: starting full datafile backup set
channel c3: specifying datafile(s) in backup set
input datafile file number=00004
name=/u01/app/oracle/oradata/DBN1/datafile/o1_mf_undotbs1_d0mq8jxv_.dbf
input datafile file number=00005
name=/u01/app/oracle/oradata/DBN1/datafile/ogg_dat01.dbf
channel c3: starting piece 1 at 23-JAN-17
channel c3: finished piece 1 at 23-JAN-17
piece handle=/home/oracle/BACKUP/dbf/DBN1/DBF 0hrqnjjq 1 1.dbf
tag=TAG20170123T061313 comment=NONE
channel c3: backup set complete, elapsed time: 00:00:07
channel c3: starting full datafile backup set
channel c3: specifying datafile(s) in backup set
including current control file in backup set
channel c3: starting piece 1 at 23-JAN-17
channel c3: finished piece 1 at 23-JAN-17
piece handle=/home/oracle/BACKUP/dbf/DBN1/DBF_0irqnjk1_1_1.dbf
tag=TAG20170123T061313 comment=NONE
channel c3: backup set complete, elapsed time: 00:00:01
channel c2: finished piece 1 at 23-JAN-17
piece handle=/home/oracle/BACKUP/dbf/DBN1/DBF_0grqnjjq_1_1.dbf
tag=TAG20170123T061313 comment=NONE
channel c2: backup set complete, elapsed time: 00:01:34
channel c1: finished piece 1 at 23-JAN-17
piece handle=/home/oracle/BACKUP/dbf/DBN1/DBF_0frqnjjq_1_1.dbf
tag=TAG20170123T061313 comment=NONE
channel c1: backup set complete, elapsed time: 00:01:46
Finished backup at 23-JAN-17

Starting Control File and SPFILE Autobackup at 23-JAN-17
piece
handle=/u01/app/oracle/fast recovery area/DBN1/autobackup/2017 01 23/o1 mf s 934
006501_d8bx9px0_.bkp comment=NONE
Finished Control File and SPFILE Autobackup at 23-JAN-17

sql statement: ALTER SYSTEM SWITCH LOGFILE

Starting backup at 23-JAN-17
current log archived
channel c1: starting archived log backup set
channel c1: specifying archived log(s) in backup set
input archived log thread=1 sequence=289 RECID=540 STAMP=933843064
input archived log thread=1 sequence=290 RECID=542 STAMP=933865238
input archived log thread=1 sequence=291 RECID=544 STAMP=933890106
input archived log thread=1 sequence=292 RECID=546 STAMP=933891613
input archived log thread=1 sequence=293 RECID=548 STAMP=933912058
input archived log thread=1 sequence=294 RECID=550 STAMP=933919248
```

```
input archived log thread=1 sequence=295 RECID=552 STAMP=933924012
input archived log thread=1 sequence=296 RECID=554 STAMP=933933797
input archived log thread=1 sequence=297 RECID=556 STAMP=933944861
input archived log thread=1 sequence=298 RECID=558 STAMP=933948225
input archived log thread=1 sequence=299 RECID=560 STAMP=933962649
input archived log thread=1 sequence=300 RECID=562 STAMP=933977074
input archived log thread=1 sequence=301 RECID=564 STAMP=933977092
input archived log thread=1 sequence=302 RECID=566 STAMP=933996044
channel c1: starting piece 1 at 23-JAN-17
channel c2: starting archived log backup set
channel c2: specifying archived log(s) in backup set
input archived log thread=1 sequence=262 RECID=487 STAMP=933641933
input archived log thread=1 sequence=263 RECID=489 STAMP=933641936
input archived log thread=1 sequence=264 RECID=490 STAMP=933641948
input archived log thread=1 sequence=265 RECID=492 STAMP=933642003
input archived log thread=1 sequence=266 RECID=494 STAMP=933644043
input archived log thread=1 sequence=267 RECID=496 STAMP=933667042
input archived log thread=1 sequence=268 RECID=498 STAMP=933689651
input archived log thread=1 sequence=269 RECID=500 STAMP=933705164
input archived log thread=1 sequence=270 RECID=502 STAMP=933705172
input archived log thread=1 sequence=271 RECID=504 STAMP=933705189
input archived log thread=1 sequence=272 RECID=506 STAMP=933705202
input archived log thread=1 sequence=273 RECID=508 STAMP=933717621
channel c2: starting piece 1 at 23-JAN-17
channel c3: starting archived log backup set
channel c3: specifying archived log(s) in backup set
input archived log thread=1 sequence=274 RECID=510 STAMP=933717654
input archived log thread=1 sequence=275 RECID=512 STAMP=933721234
input archived log thread=1 sequence=276 RECID=514 STAMP=933742805
input archived log thread=1 sequence=277 RECID=516 STAMP=933766664
input archived log thread=1 sequence=278 RECID=518 STAMP=933789640
input archived log thread=1 sequence=279 RECID=520 STAMP=933802307
input archived log thread=1 sequence=280 RECID=521 STAMP=933802564
input archived log thread=1 sequence=281 RECID=524 STAMP=933802646
input archived log thread=1 sequence=282 RECID=526 STAMP=933804051
input archived log thread=1 sequence=283 RECID=528 STAMP=933804365
input archived log thread=1 sequence=284 RECID=530 STAMP=933818909
input archived log thread=1 sequence=285 RECID=532 STAMP=933832827
input archived log thread=1 sequence=286 RECID=534 STAMP=933838413
input archived log thread=1 sequence=287 RECID=536 STAMP=933842626
input archived log thread=1 sequence=288 RECID=537 STAMP=933842996
channel c3: starting piece 1 at 23-JAN-17
channel c3: finished piece 1 at 23-JAN-17
piece handle=/home/oracle/BACKUP/arc/DBN1/ARC 0mrqnjni 1 1.dbf
tag=TAG20170123T061513 comment=NONE
channel c3: backup set complete, elapsed time: 00:01:26
channel c3: starting archived log backup set
channel c3: specifying archived log(s) in backup set
input archived log thread=1 sequence=303 RECID=568 STAMP=934002010
input archived log thread=1 sequence=304 RECID=570 STAMP=934006510
input archived log thread=1 sequence=305 RECID=572 STAMP=934006512
channel c3: starting piece 1 at 23-JAN-17
channel c3: finished piece 1 at 23-JAN-17
piece handle=/home/oracle/BACKUP/arc/DBN1/ARC_0nrqnjq8_1_1.dbf
tag=TAG20170123T061513 comment=NONE
channel c3: backup set complete, elapsed time: 00:00:15
channel c1: finished piece 1 at 23-JAN-17
piece handle=/home/oracle/BACKUP/arc/DBN1/ARC_0krqnjni_1_1.dbf
tag=TAG20170123T061513 comment=NONE
```

```
channel c1: backup set complete, elapsed time: 00:01:51
channel c2: finished piece 1 at 23-JAN-17
piece handle=/home/oracle/BACKUP/arc/DBN1/ARC 01rqnjni 1 1.dbf
tag=TAG20170123T061513 comment=NONE
channel c2: backup set complete, elapsed time: 00:01:51
Finished backup at 23-JAN-17

Starting Control File and SPFILE Autobackup at 23-JAN-17
piece
handle=/u01/app/oracle/fast_recovery_area/DBN1/autobackup/2017_01_23/o1_mf_s_934
006630 d8bxfq1r .bkp comment=NONE
Finished Control File and SPFILE Autobackup at 23-JAN-17

released channel: c1

released channel: c2

released channel: c3

Recovery Manager complete.
```

Task 8: Deploy Tablespace Encryption

There are several techniques to encrypt the database. The most common technique is Transparant Data Encryption (TDE), which can be applied at the tablespace, table and column. This task deploys tablespace encryption for the USERS (data) tablespace only. The steps to deploy TDE is by creating the keystore and master key.

1. Identified the location of the keystore by editing SQLNET.ORA file
2. Create the directory location
3. Restart the listener
4. Restart the database
5. Create the password based ketstore file
6. Open the keystore
7. Create the keystore master key
8. Convert the keystore to AUTO_LOGIN keystore from PASSWORD keystore
9. Restart the database
10. Create the encrtypted tablespace
11. Move data to the encrypted tablespace

On the primary node

Identify the location of the keystore by editing the instance SQLNET.ORA file. The keystore is identified by the parameter ENCRYPTION_WALLET_LOCATION. Add the parameter as follows:

```
ENCRYPTION_WALLET_LOCATION =
(SOURCE =
    (METHOD = FILE)
```

```
    (METHOD_DATA =
        (DIRECTORY=/u01/app/oracle/admin/DBN0/wallet)
    )
)
```

```
[oracle@edbcn0 admin]$ vi sqlnet.ora
```

```
[oracle@edbcn0 admin]$ cat sqlnet.ora
```

```
# setting the encryption parameters for redo-transport
sqlnet.crypto_seed = "kclabefmnoc"
sqlnet.encryption_server = accepted
sqlnet.encryption_client = required
sqlnet.encryption_types_client = AES128
sqlnet.encryption_types_server = AES128
trace directory server=/u01/app/oracle/product/12.1.0/dbhome 1/network/log
trace_level_client=16
trace_level_server=16
# setting wallet location
ENCRYPTION_WALLET_LOCATION =
(SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
        (DIRECTORY=/u01/app/oracle/admin/DBN0/wallet)
    )
)
```

Create the ketystore location as defined in SQLNET.ORA

```
[oracle@edbcn0 admin]$ mkdir -p /u01/app/oracle/admin/DBN0/wallet
```

Restart the listener

```
[oracle@edbcn0 admin]$ lsnrctl

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 28-JAN-2017 14:00:03

Copyright (c) 1991, 2014, Oracle.  All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> stop
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edbcn0.oracle.com)(PORT=1521)))
The command completed successfully
LSNRCTL> start
Starting /u01/app/oracle/product/12.1.0/dbhome_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.2.0 - Production
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome 1/network/admin/listener.ora
Log messages written to
/u01/app/oracle/diag/tnslsnr/edbcn0/listener/alert/log.xml
```

```
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn0.oracle.com)(PORT=1521)))

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edbcn0.oracle.com)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.2.0 - Production
Start Date                28-JAN-2017 14:00:06
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/edbcn0/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn0.oracle.com)(PORT=1521)))
Services Summary...
Service "DBN0.oracle.com" has 1 instance(s).
  Instance "DBN0", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
LSNRCTL> exit
```

Restart the database

```
[oracle@edbcn0 admin]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sat Jan 28 14:00:25 2017

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

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

Total System Global Area 1660944384 bytes
Fixed Size              2925072 bytes
Variable Size           1040190960 bytes
Database Buffers      603979776 bytes
Redo Buffers            13848576 bytes
Database mounted.
Database opened.
SQL>
```

Create the keystore file

```
SQL> administer key management create keystore
'/u01/app/oracle/admin/DBN0/wallet' identified by "oracle 4U";

keystore altered.

SQL>
```

Verify the keystore create, status is CLOSED now

```
SQL> select * from v$encryption_wallet;

WRL_TYPE
--------------------
WRL_PARAMETER
--------------------------------------------------------------------------------
STATUS                         WALLET_TYPE         WALLET_OR FULLY_BAC
------------------------------ -------------------- --------- ---------
    CON_ID
----------
FILE
/u01/app/oracle/admin/DBN0/wallet/
CLOSED                         UNKNOWN             SINGLE    UNDEFINED
      0


SQL>
```

Open the Keystore

```
SQL> administer key management set keystore open identified by "oracle_4U";

keystore altered.

SQL>
```

Verify keystore status (OPEN_NO_MASTER_KEY)

```
SQL> select * from v$encryption_wallet;

WRL_TYPE
--------------------
WRL_PARAMETER
--------------------------------------------------------------------------------
STATUS                         WALLET_TYPE         WALLET_OR FULLY_BAC
------------------------------ -------------------- --------- ---------
    CON_ID
----------
FILE
/u01/app/oracle/admin/DBN0/wallet/
OPEN_NO_MASTER_KEY         PASSWORD            SINGLE    UNDEFINED
      0


SQL>
```

Add the master key to the keystore

```
SQL> administer key management set key identified by "oracle 4U" with backup;
```

```
keystore altered.

SQL>
```

Confirm status (OPEN)

```
SQL> select * from v$encryption wallet;

WRL_TYPE
--------------------
WRL_PARAMETER
--------------------------------------------------------------------------------
STATUS                         WALLET_TYPE         WALLET_OR FULLY_BAC
------------------------------ -------------------- --------- ---------
    CON_ID
----------
FILE
/u01/app/oracle/admin/DBN0/wallet/
OPEN                           PASSWORD            SINGLE    NO
      0


SQL>
```

Convert the keystore to AUTO_LOGIN

```
SQL> administer key management create auto_login keystore from keystore
'/u01/app/oracle/admin/DBN0/wallet/' identified by "oracle_4U";

keystore altered.

SQL>
```

Restart the Database

```
SQL> startup force
ORACLE instance started.

Total System Global Area          1660944384 bytes
Fixed Size                           2925072 bytes
Variable Size                      989859312 bytes
Database Buffers                   654311424 bytes
Redo Buffers                        13848576 bytes
Database mounted.
Database opened.
SQL>
```

Verify AUTO_LOGIN keystore

```
SQL> select * from v$encryption_wallet;

WRL_TYPE
--------------------
```

```
WRL_PARAMETER
--------------------------------------------------------------------------------
STATUS                         WALLET TYPE         WALLET OR FULLY BAC
------------------------------ -------------------- --------- ---------
    CON_ID
----------
FILE
/u01/app/oracle/admin/DBN0/wallet/
OPEN                        AUTOLOGIN           SINGLE    NO
      0


SQL>
```

Create encrypted tablespace

```
SQL> create tablespace users enc
  2  datafile '/u01/app/oracle/oradata/DBN0/datafile/users_enc_01.dbf'
  3  size 1000M autoextend on
  4  encryption using 'AES128'
  5  default storage (ENCRYPT);

Tablespace created.

SQL>
```

Migrate the tables to the encrypted tablespace

```
SQL> alter table osm_repapi.customers move tablespace users_enc;

Table altered.

SQL> alter table osm_repapi.policies move tablespace users_enc;

Table altered.

SQL>
```

On extended node 1

Add keystore location in SQLNET.ORA file

```
[oracle@edbcn1 ~]$ cd $ORACLE HOME/network/admin
[oracle@edbcn1 admin]$ vi sqlnet.ora
```

```
# setting the encryption parameters for redo-transport
sqlnet.crypto_seed = "kclabefmnoc"
sqlnet.encryption server = accepted
sqlnet.encryption_client = required
sqlnet.encryption_types_client = AES128
sqlnet.encryption types server = AES128
trace_directory_server=/u01/app/oracle/product/12.1.0/dbhome_1/network/log
trace_level_client=16
```

```
trace_level_server=16
# setting wallet location
ENCRYPTION WALLET LOCATION =
(SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
        (DIRECTORY = /u01/app/oracle/admin/DBN1/wallet)
    )
)
```

Create the keystore directory

```
[oracle@edbcn1 admin]$ mkdir -p /u01/app/oracle/admin/DBN1/wallet
[oracle@edbcn1 admin]$ cd /u01/app/oracle/admin/DBN1/wallet
[oracle@edbcn1 wallet]$ ls
```

```
Stop and start listener and database

[oracle@edbcn1 wallet]$ lsnrctl

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 28-JAN-2017 14:05:57

Copyright (c) 1991, 2014, Oracle.  All rights reserved.

Welcome to LSNRCTL, type "help" for information.

LSNRCTL> stop
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edbcn1.oracle.com)(PORT=1521)))
The command completed successfully
LSNRCTL> start
Starting /u01/app/oracle/product/12.1.0/dbhome_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.2.0 - Production
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome 1/network/admin/listener.ora
Log messages written to
/u01/app/oracle/diag/tnslsnr/edbcn1/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn1.oracle.com)(PORT=1521)))

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edbcn1.oracle.com)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.2.0 - Production
Start Date                28-JAN-2017 14:06:00
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome 1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/edbcn1/listener/alert/log.xml
Listening Endpoints Summary...
```

```
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn1.oracle.com)(PORT=1521)))
Services Summary...
Service "DBN1.oracle.com" has 1 instance(s).
  Instance "DBN1", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
LSNRCTL> exit
```

```
[oracle@edbcn1 wallet]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sat Jan 28 14:10:22 2017

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

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

Total System Global Area          1660944384 bytes
Fixed Size                           2925072 bytes
Variable Size                      989859312 bytes
Database Buffers                   654311424 bytes
Redo Buffers                        13848576 bytes
Database mounted.
Database opened.
SQL>
```

Create the password type keystore

```
SQL> administer key management create keystore
'/u01/app/oracle/admin/DBN1/wallet' identified by "oracle_4U";

keystore altered.

SQL>
```

Display keystore details, CLOSED and not yet known.

```
SQL> select * from v$encryption wallet;

WRL TYPE
--------------------
WRL_PARAMETER
--------------------------------------------------------------------------------
STATUS                         WALLET TYPE         WALLET OR FULLY BAC
------------------------------ -------------------- --------- ---------
    CON ID
----------
FILE
/u01/app/oracle/admin/DBN1/wallet/
CLOSED                         UNKNOWN             SINGLE    UNDEFINED
      0


SQL>
```

Open the ketstore

```
SQL> administer key management set keystore open identified by "oracle_4U";

keystore altered.

SQL>
```

Display keystore details, it shows it's of type PASSWORD and status OPEN_NO_MASTER_KEY. Perfect.

```
SQL> select * from v$encryption_wallet;

WRL TYPE
--------------------
WRL_PARAMETER
------------------------------------------------------------------------------
STATUS                           WALLET_TYPE         WALLET_OR FULLY_BAC
-------------------------------- -------------------- --------- ---------
    CON_ID
----------
FILE
/u01/app/oracle/admin/DBN1/wallet/
OPEN_NO_MASTER_KEY         PASSWORD            SINGLE    UNDEFINED
     0


SQL>
```

Add the master key ot the keystore

```
SQL> administer key management set key identified by "oracle_4U" with backup;

keystore altered.

SQL>
```

Query current status

```
SQL> select * from v$encryption_wallet;

WRL TYPE
--------------------
WRL PARAMETER
------------------------------------------------------------------------------
STATUS                           WALLET_TYPE         WALLET_OR FULLY_BAC
-------------------------------- -------------------- --------- ---------
    CON ID
----------
FILE
/u01/app/oracle/admin/DBN1/wallet/
OPEN                         PASSWORD            SINGLE    NO
```

```
      0


SQL>
```

Convert the keystore to AUTO_LOGIN type

```
SQL> administer key management create auto login keystore from keystore
'/u01/app/oracle/admin/DBN1/wallet/' identified by "oracle_4U";

keystore altered.

SQL>
```

Restart the database

```
SQL> startup force
ORACLE instance started.

Total System Global Area          1660944384 bytes
Fixed Size                           2925072 bytes
Variable Size                      989859312 bytes
Database Buffers                   654311424 bytes
Redo Buffers                        13848576 bytes
Database mounted.
Database opened.
SQL>
```

Re-query the keystore and verify it's open and of type AUTO_LOGIN

```
SQL> select * from v$encryption wallet;

WRL_TYPE
--------------------
WRL_PARAMETER
--------------------------------------------------------------------------------
STATUS                           WALLET_TYPE         WALLET_OR FULLY_BAC
------------------------------ -------------------- --------- ---------
    CON_ID
----------
FILE
/u01/app/oracle/admin/DBN1/wallet/
OPEN                         AUTOLOGIN           SINGLE    NO
      0


SQL>
```

Create the encrypted Tablespace

```
SQL> create tablespace users_enc
  2  datafile '/u01/app/oracle/oradata/DBN1/datafile/users_enc_01.dbf'
  3  size 1000M autoextend on
```

```
  4  encryption using 'AES128'
  5  default storage (ENCRYPT);

Tablespace created.

SQL>
```

Move the data from non-encrtypted tablespace to encrypted tablespace

```
SQL> alter table osm repapi.customers move tablespace users enc;

Table altered.

SQL> alter table osm repapi.policies move tablespace users enc;

Table altered.

SQL>
```

On extended node 2

```
[oracle@edbcn2 ~]$ cd $ORACLE HOME/network/admin
[oracle@edbcn2 admin]$ vi sqlnet.ora
```

```
[oracle@edbcn2 admin]$ cat sqlnet.ora
# setting the encryption parameters for redo-transport
sqlnet.crypto_seed = "kclabefmnoc"
sqlnet.encryption_server = accepted
sqlnet.encryption_client = required
sqlnet.encryption types client = AES128
sqlnet.encryption_types_server = AES128
trace directory server=/u01/app/oracle/product/12.1.0/dbhome 1/network/log
trace_level_client=16
trace_level_server=16
# setting wallet location
ENCRYPTION_WALLET_LOCATION =
(SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
        (DIRECTORY = /u01/app/oracle/admin/DBN2/wallet)
    )
)
```

```
[oracle@edbcn2 admin]$ mkdir -p /u01/app/oracle/admin/DBN2/wallet
[oracle@edbcn2 admin]$ cd /u01/app/oracle/admin/DBN2/wallet
[oracle@edbcn2 wallet]$ ls
```

```
[oracle@edbcn2 wallet]$ lsnrctl

LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 28-JAN-2017 14:28:31

Copyright (c) 1991, 2014, Oracle.  All rights reserved.

Welcome to LSNRCTL, type "help" for information.
```

```
LSNRCTL> stop
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edbcn2.oracle.com)(PORT=1521)))
The command completed successfully
LSNRCTL> start
Starting /u01/app/oracle/product/12.1.0/dbhome_1/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.2.0 - Production
System parameter file is
/u01/app/oracle/product/12.1.0/dbhome 1/network/admin/listener.ora
Log messages written to
/u01/app/oracle/diag/tnslsnr/edbcn2/listener/alert/log.xml
Listening on:
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn2.oracle.com)(PORT=1521)))
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))

Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=edbcn2.oracle.com)(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 12.1.0.2.0 - Production
Start Date                28-JAN-2017 14:28:34
Uptime                    0 days 0 hr. 0 min. 0 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File
/u01/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/edbcn2/listener/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=edbcn2.oracle.com)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
Services Summary...
Service "DBN2.oracle.com" has 1 instance(s).
  Instance "DBN2", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
LSNRCTL> exit
```

```
[oracle@edbcn2 wallet]$ sqlplus /nolog

SQL*Plus: Release 12.1.0.2.0 Production on Sat Jan 28 14:28:47 2017

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

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

Total System Global Area          1660944384 bytes
Fixed Size                           2925072 bytes
Variable Size                      989859312 bytes
Database Buffers                   654311424 bytes
Redo Buffers                        13848576 bytes
Database mounted.
```

```
Database opened.
SQL>
```

```
SQL> administer key management create keystore
'/u01/app/oracle/admin/DBN2/wallet' identified by "oracle_4U";

keystore altered.

SQL>
```

```
SQL> select * from v$encryption_wallet;

WRL TYPE
--------------------
WRL PARAMETER
--------------------------------------------------------------------------------
STATUS                           WALLET_TYPE         WALLET_OR FULLY_BAC
-------------------------------- -------------------- --------- ---------
    CON ID
----------
FILE
/u01/app/oracle/admin/DBN2/wallet/
CLOSED                           UNKNOWN             SINGLE    UNDEFINED
      0


SQL>
```

```
SQL> administer key management set keystore open identified by "oracle 4U";

keystore altered.

SQL>
```

```
SQL> select * from v$encryption_wallet;

WRL TYPE
--------------------
WRL_PARAMETER
--------------------------------------------------------------------------------
STATUS                           WALLET_TYPE         WALLET_OR FULLY_BAC
-------------------------------- -------------------- --------- ---------
    CON ID
----------
FILE
/u01/app/oracle/admin/DBN2/wallet/
OPEN_NO_MASTER_KEY         PASSWORD            SINGLE    UNDEFINED
      0


SQL>
```

```
SQL> administer key management set key identified by "oracle_4U" with backup;

keystore altered.
```

```
SQL>
```

```
SQL>  select * from v$encryption_wallet;

WRL_TYPE
--------------------
WRL_PARAMETER
--------------------------------------------------------------------------------
STATUS                           WALLET_TYPE         WALLET_OR FULLY_BAC
------------------------------ -------------------- --------- ---------
    CON_ID
----------
FILE
/u01/app/oracle/admin/DBN2/wallet/
OPEN                         PASSWORD            SINGLE    NO
      0


SQL>
```

```
SQL> administer key management create auto_login keystore from keystore
'/u01/app/oracle/admin/DBN2/wallet/' identified by "oracle_4U";

keystore altered.

SQL>
```

```
SQL> startup force
ORACLE instance started.

Total System Global Area          1660944384 bytes
Fixed Size                           2925072 bytes
Variable Size                      989859312 bytes
Database Buffers                   654311424 bytes
Redo Buffers                        13848576 bytes
Database mounted.
Database opened.
SQL>
```

```
SQL> select * from v$encryption_wallet;

WRL TYPE
--------------------
WRL_PARAMETER
--------------------------------------------------------------------------------
STATUS                           WALLET_TYPE         WALLET_OR FULLY_BAC
------------------------------ -------------------- --------- ---------
    CON_ID
----------
FILE
/u01/app/oracle/admin/DBN2/wallet/
OPEN                         AUTOLOGIN           SINGLE    NO
      0
```

```
SQL>
```

```
SQL> create tablespace users_enc
  2  datafile '/u01/app/oracle/oradata/DBN2/users_enc_01.dbf'
  3  size 1000M autoextend on
  4  encryption using 'AES128'
  5  default storage (ENCRYPT);

Tablespace created.

SQL>
```

```
SQL> create tablespace users enc
  2  datafile '/u01/app/oracle/oradata/DBN2/users_enc_01.dbf'
  3  size 1000M autoextend on
  4  encryption using 'AES128'
  5  default storage (ENCRYPT);

Tablespace created.

SQL>
```

Move the data from non-encrtypted tablespace to encrypted tablespace

```
SQL> alter table osm_repapi.customers move tablespace users_enc;

Table altered.

SQL> alter table osm_repapi.policies move tablespace users_enc;

Table altered.

SQL>
```

Summary

Congratulation and well done…

You've completed the design, build and deploy of Extended Database Cluster 12c (EDBC) workshop. Successfully completing the workshop prepares you for building an enterprise Extended Database Cluster 12c (EDBC) with compute capacity of N nodes. Figure 17 shows the workshop outline and workflow.

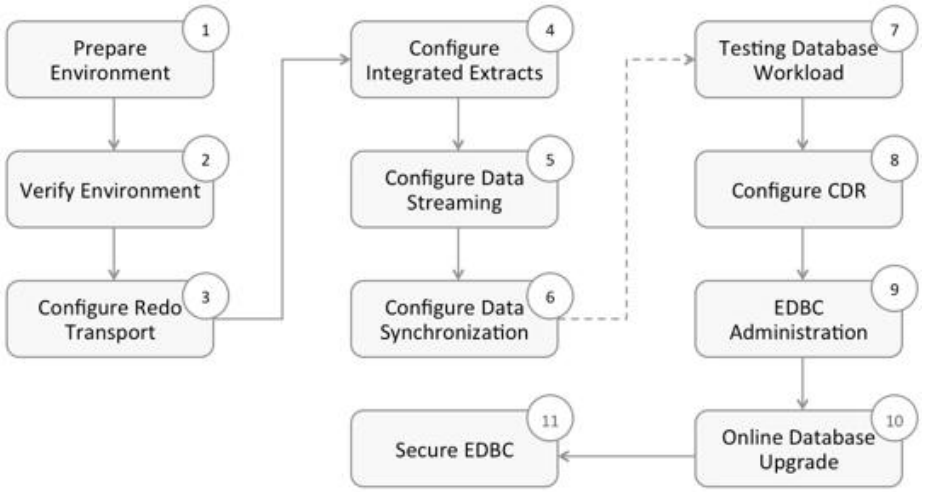


Figure 17: Workshop Outlines

In order to complete the 2-day workshop, it's required you've the basic Oracle Database 12c and Oracle GoldenGate 12c. If not, refer to APPENDIX A for Oracle GoldenGate 12c Fundamentals. Part of the workshop is mandatory and must be completed successfully, hence Workshop 1 through Workshop 6 are mandatory, and Workshop 7 through Workshop 11 is optional. However; It's highly recommended to complete Workshop 1-11 on the sequence developed. The workshop is deliverable only using the supplied Oracle VirtualBox for the primary node, extended nodes and downstream systems running Oracle Database 11g/12c Enterprise Edition and Oracle GoldenGate 12c.

If you have questions regarding Extended Database Cluster 12c (EDBC), please contact the EDBC developer directly:

EDBC Developer: Fawzi A. Alswaimil
E-Mail: Fawzi.alswaimil@oracle.com | falswaimil@gmail.com
Mobile: +966 50 345 5094

Oracle GoldenGate Fundamentals

APPENDIX

A

Overview

The objective of this appendix 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.learn-ogg.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.

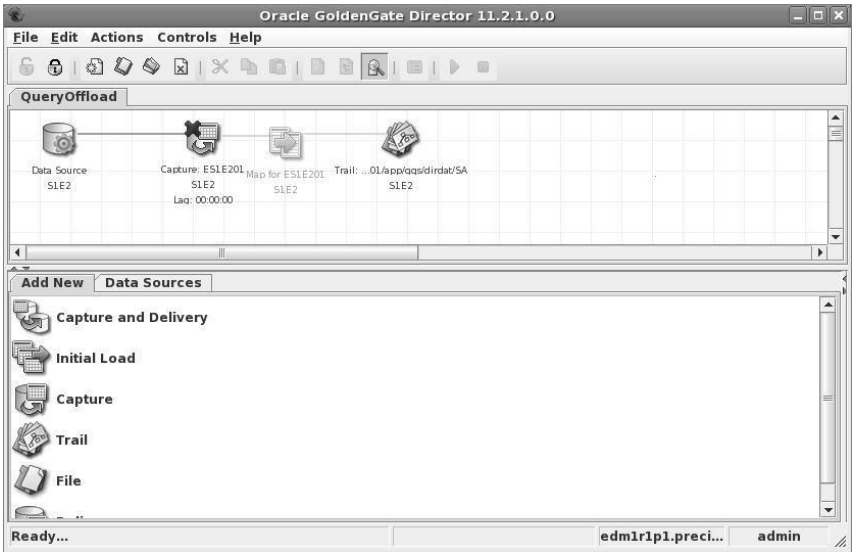


Figure 1-1: Oracle GoldenGate director visual diagramming tool

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.

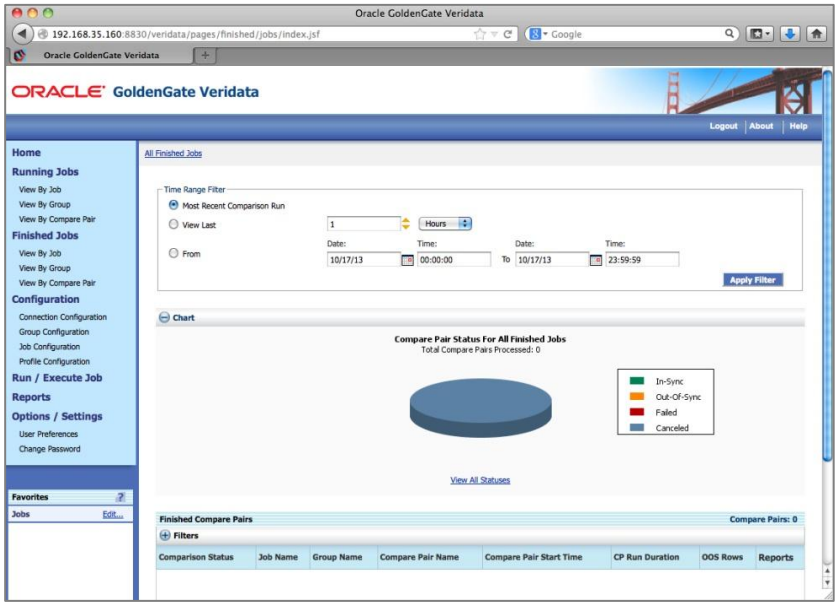


Figure 1-2: Oracle GoldenGate Veridata

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

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.

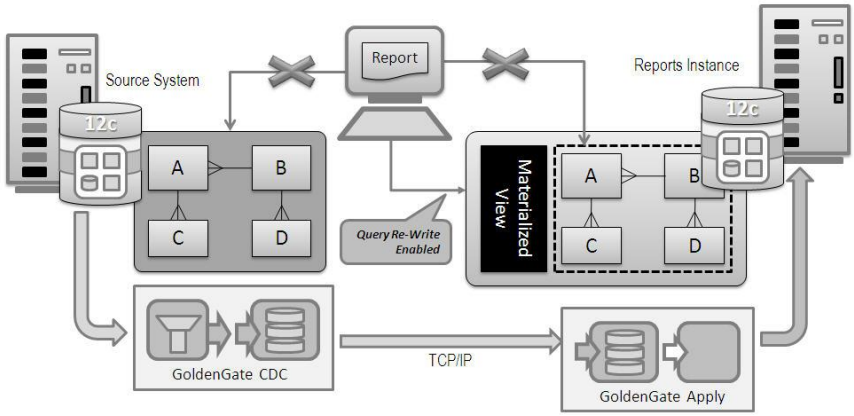


Figure 1-3: Query offloading technique

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.

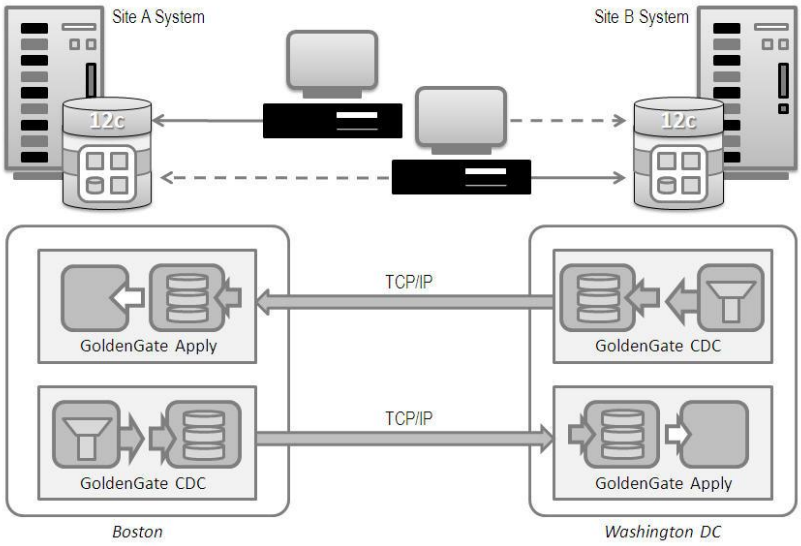


Figure 1-4: Oracle GoldenGate for high-availability

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.

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.

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.

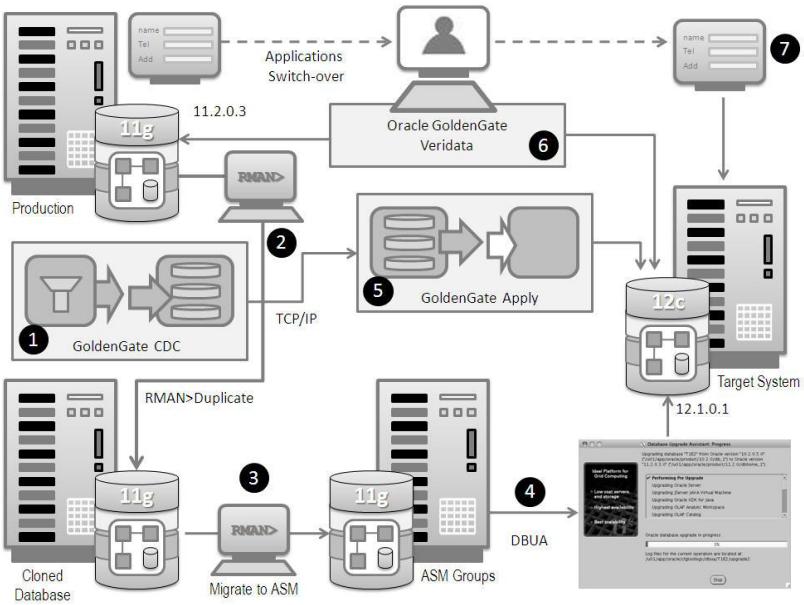


Figure 1-5: Zero downtime migration and upgrade workflow

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.

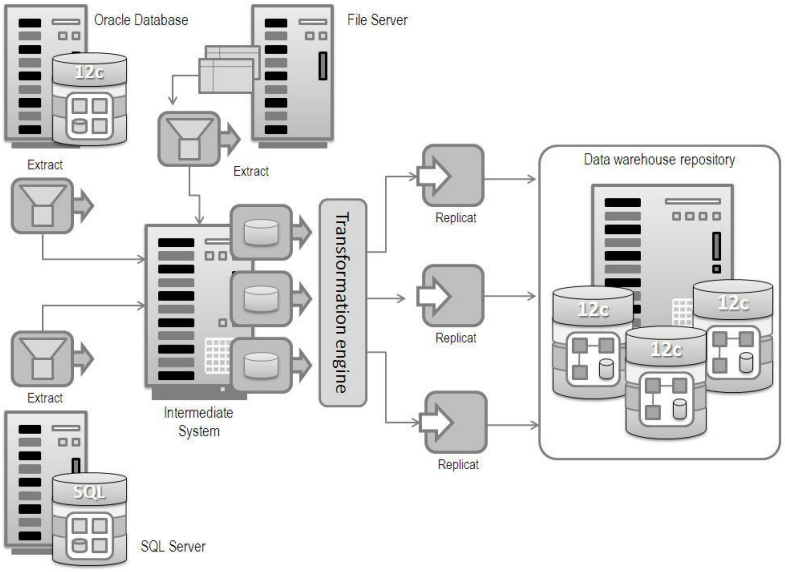


Figure 1-6: Data warehouse feed from heterogamous systems

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.

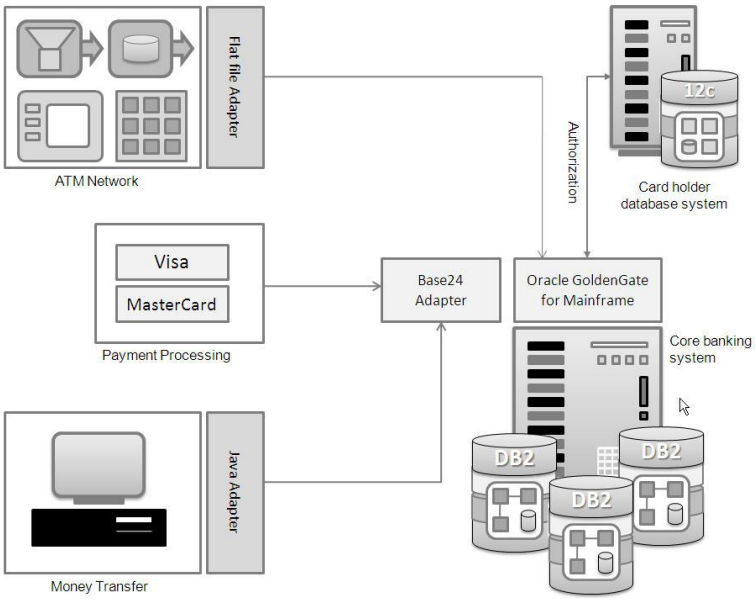


Figure 1-7: payment processing systems

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

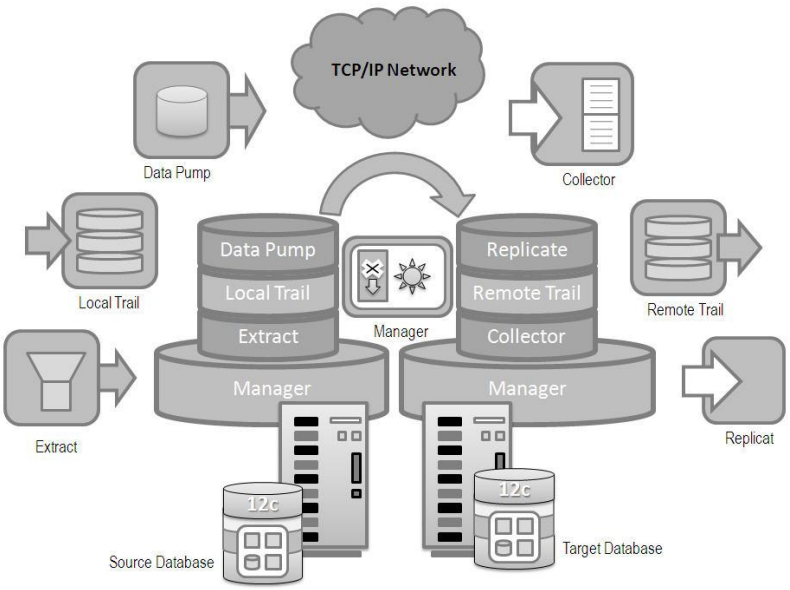


Figure 1-8: Oracle GoldenGate building blocks

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.

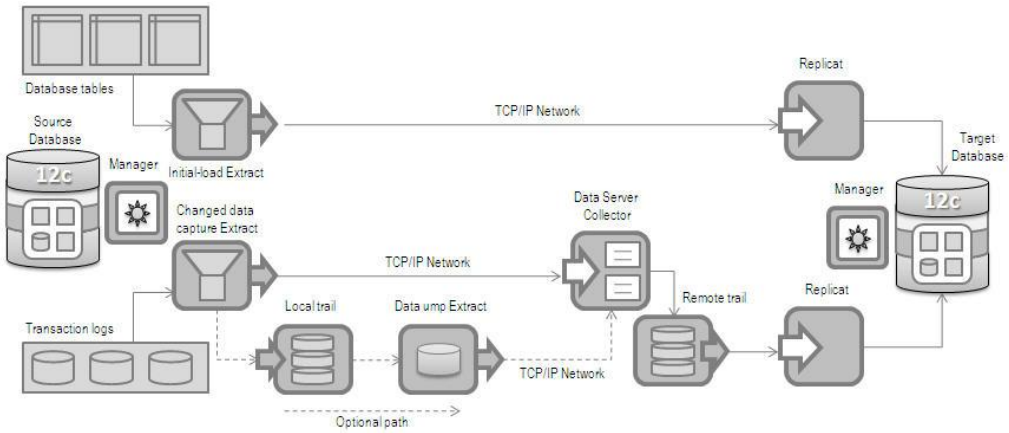


Figure 1-9: Building blocks connected

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 shows 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.

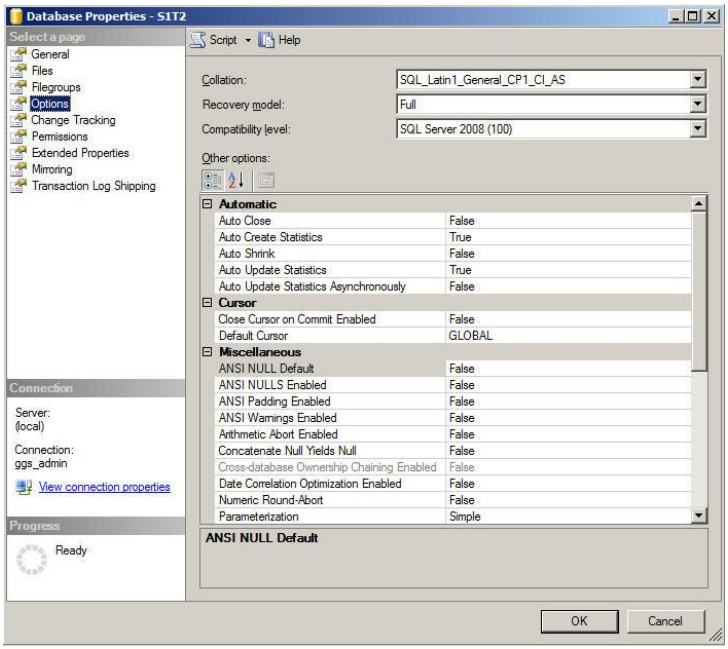


Figure 1-10: MS SQL Server recovery model

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.

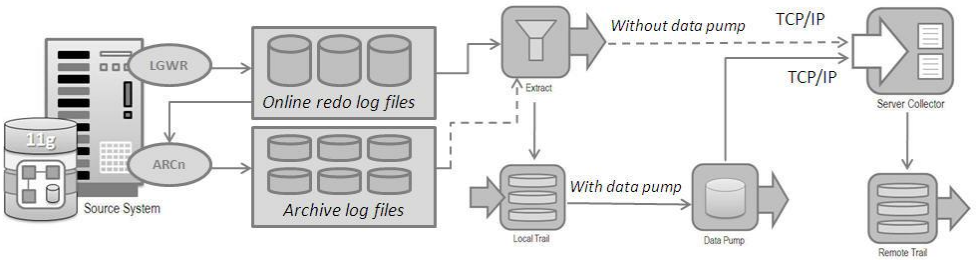


Figure 1-11: Online extract process options

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, ensuing the transaction committed order of the source database for data consistency and integrity. The online replicate process expect 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 it's read and write progress using either of two options:

- Concurrently to checkpoint file and database table, know 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 up on 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 invoking 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.

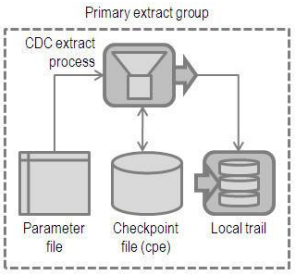
Remote trail files

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

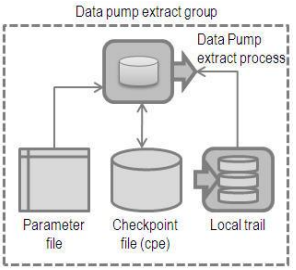
The real-time data movement from source to target systems handled by logical groups. The logical grouping allows an effective allocation and management of physical resource associated with each group. Data down-streaming starts at the capture group, optionally through the secondary group and ends at 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



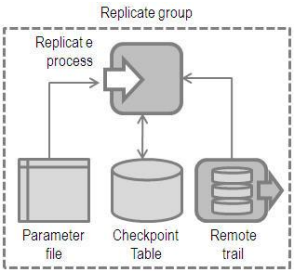
This is the capture group, it consists of an extract process for performing change data capture (CDC), a parameter file with the name <group name.prm>, a checkpoint file with the name <group name.cpe>, output local trail or remote trail files. The group is associated with its own discard and report files. 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> checkpoint file with the name <group name.cpe) and input local trail. The group is also associated with its discard and report files. The data pump uses TCP/IP communicate with a dynamic dedicated data server collector, which re-assemble and write to the remote 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>.

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.

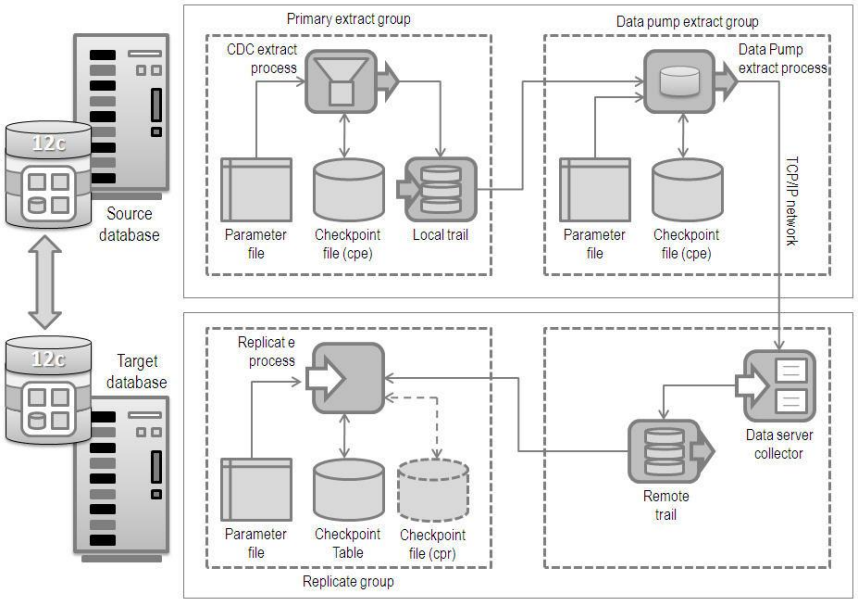


Figure 1-12: Process grouping and workflow

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.

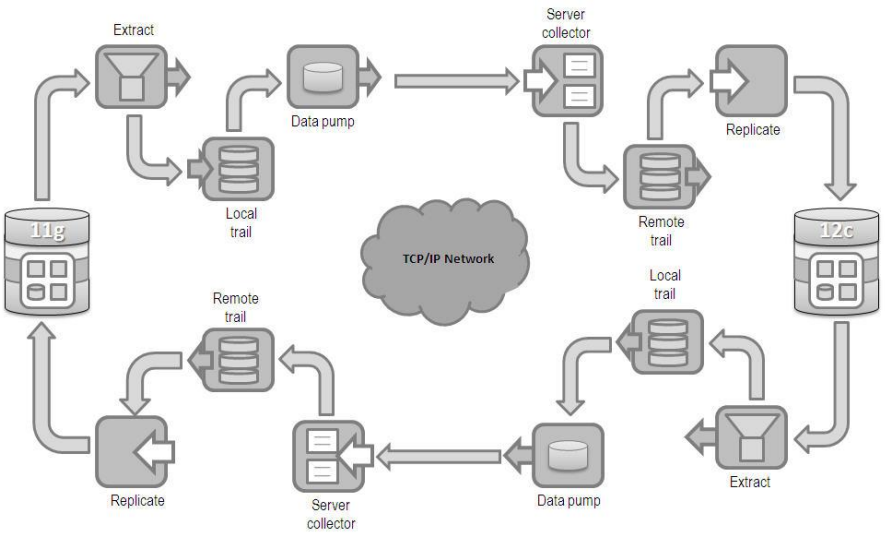


Figure 1-13: GoldenGate processes architecture

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.

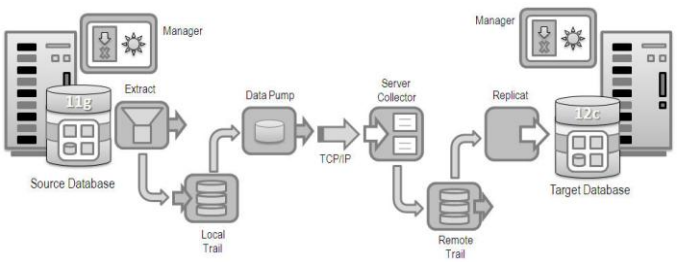


Figure 1-14: Unidirectional Oracle GoldenGate components

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 follow 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.

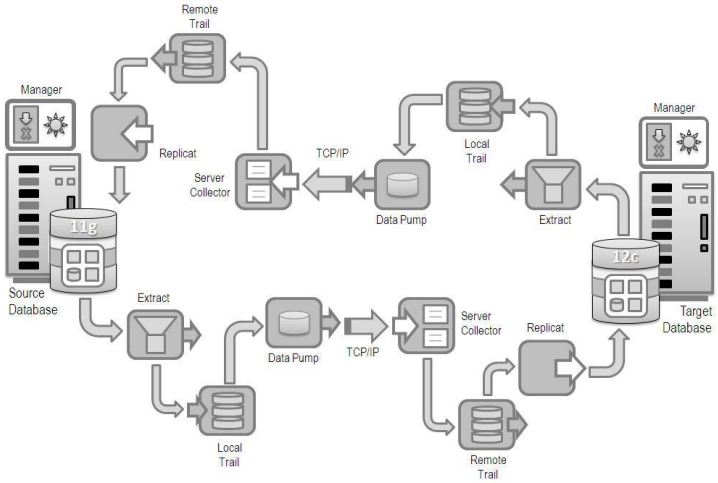


Figure 1-15: Bidirectional configuration

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.

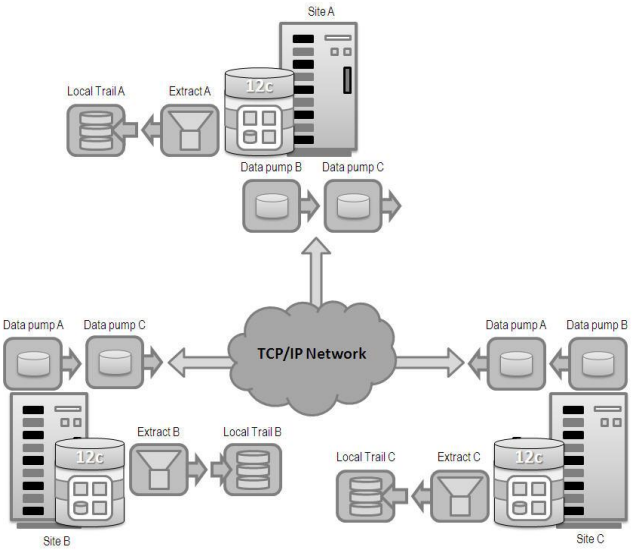


Figure 1-16: Master-to-master replication

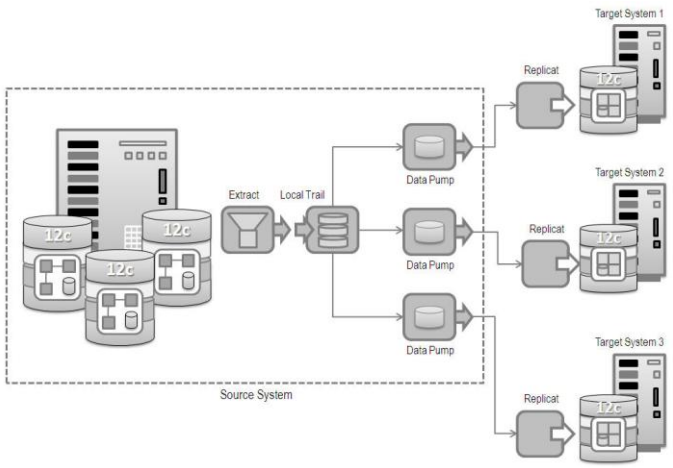
Figure 1-16 show master-to-master for three site. Extending this topology by streaming data from a regional master to a local site achieves efficient transaction processing and higher data availability.

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.



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.

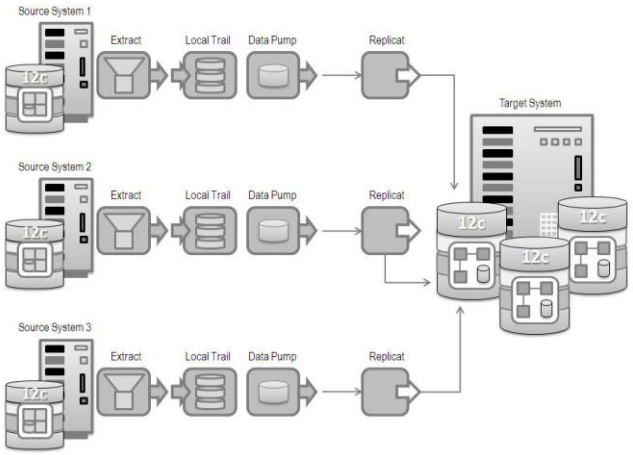


Figure 1-18: Consolidate for data warehouse repository

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.

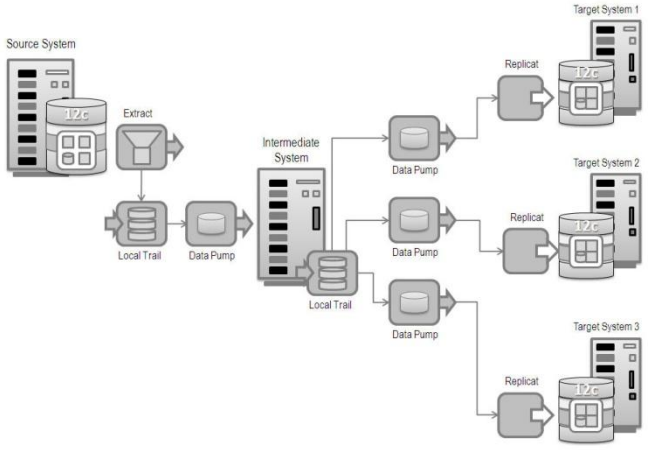


Figure 1-19: Cascade using intermediate system

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.

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.

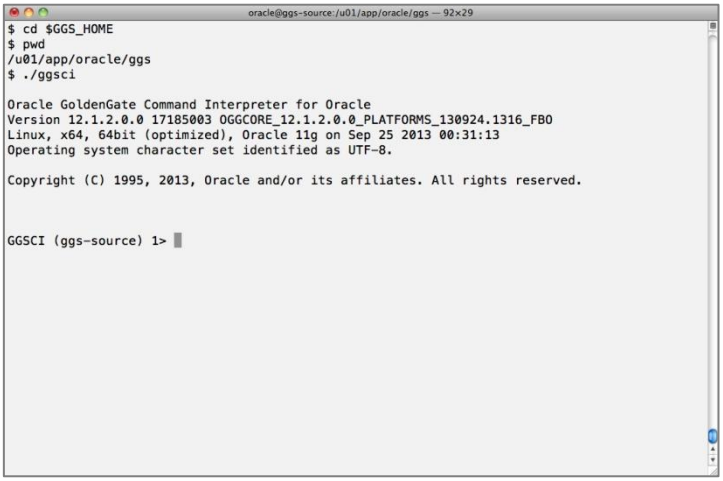


Figure 1-20: Oracle GoldenGate software command interface (GGSCI)

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

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
```

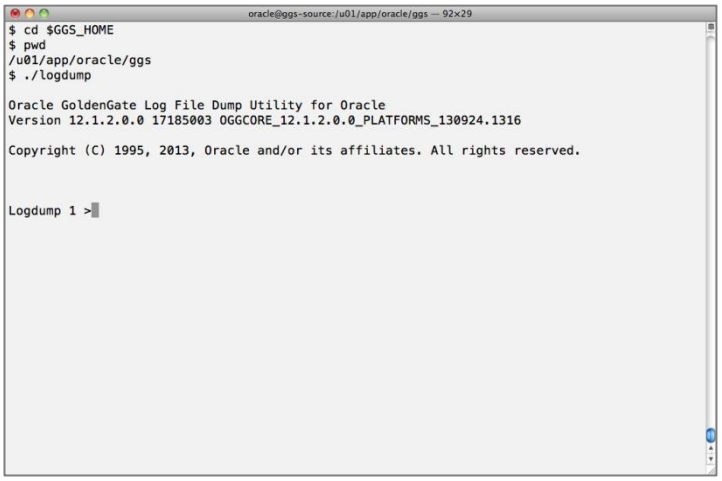


Figure 1-21: Oracle GoldenGate log file dump utility

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 has 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 specify a static TCP/IP listening port number.
- There can be one-and-only-one manager process per Oracle GoldenGate instance.
- The manager process status 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 effecting currently running processes.

The following is an example of common manager process parameters. These parameters' values settings 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 PUREGEOLDEXTRACTS allows the manager to perform trail files 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
RMTTRAIL ./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 RMTTRAIL 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 RMTTRAIL command.

```
GGSCI (ggs-source) 28> ADD RMTTRAIL ./dirdat/ta, EXTRACT epmp01,
MEGABYTES 32
RMTTRAIL added.
```

Use the command START EXTRACT * for staring 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.

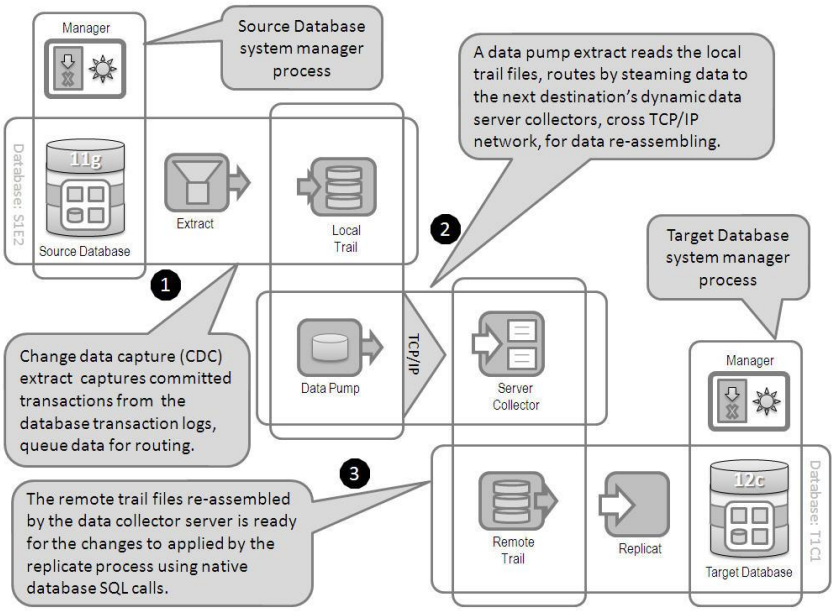
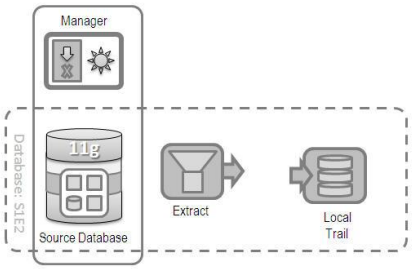


Figure 1-22: Oracle GoldenGate modular architecture

Figure 1-22 illustrates the building block for the three modular blocks:

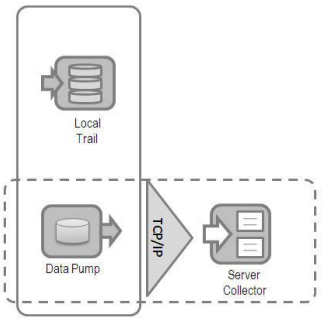
- Change data capture phase using primary extract group
- Routing phase using secondary extract group
- Apply phase using change data delivery group

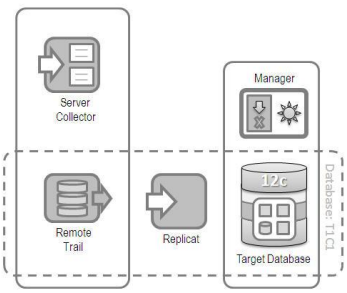
The capture phase, it involves one or more primary extract capturing *only committed* transactions as they becomes available on the transaction logs. This is occurring outside of the database, hence minimum impact sustained by the source database. For achieving a fault-tolerance non-interruptible data 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.

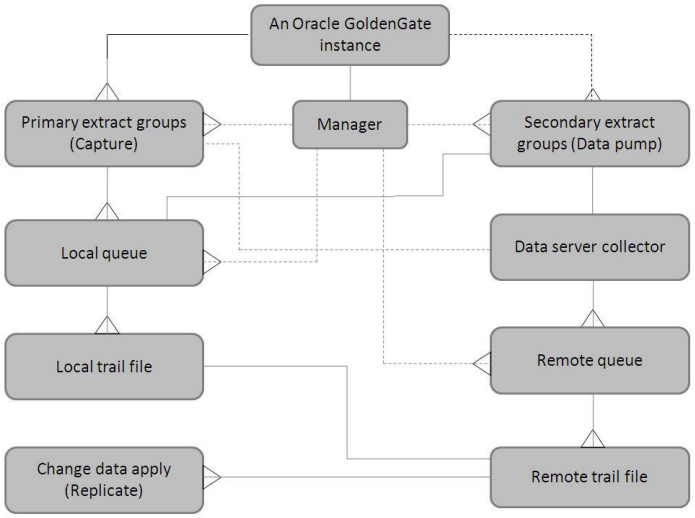


Figure 1-23: GoldenGate components relationship

An Oracle GoldenGate instance must has one-and-only-one manager. An *operational* Oracle GoldenGate instance must be composed of one-or-more primary extract groups (capture), but may has one-or-more secondary extract groups (data pump).

Each primary extract must creates 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 files applied using the target database SQL native interface.

The manager process may manages 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.

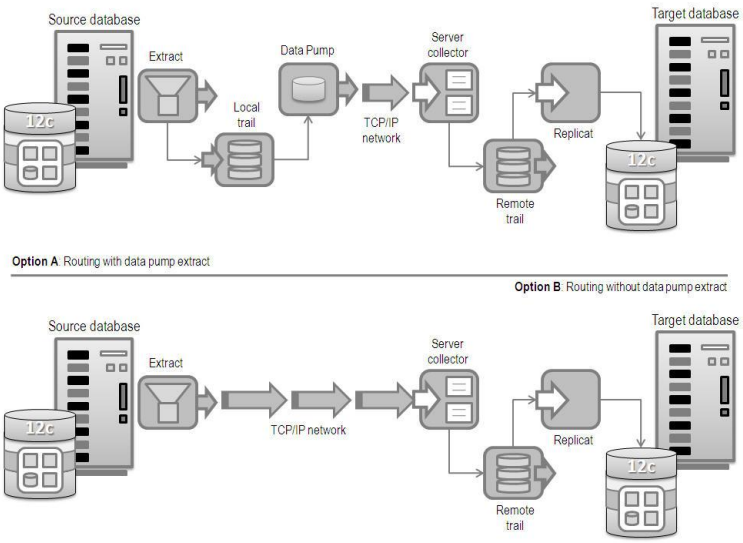


Figure 1-24: End-to-end configuration options

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 REPLCIATE. 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 REPLCIATE.

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.

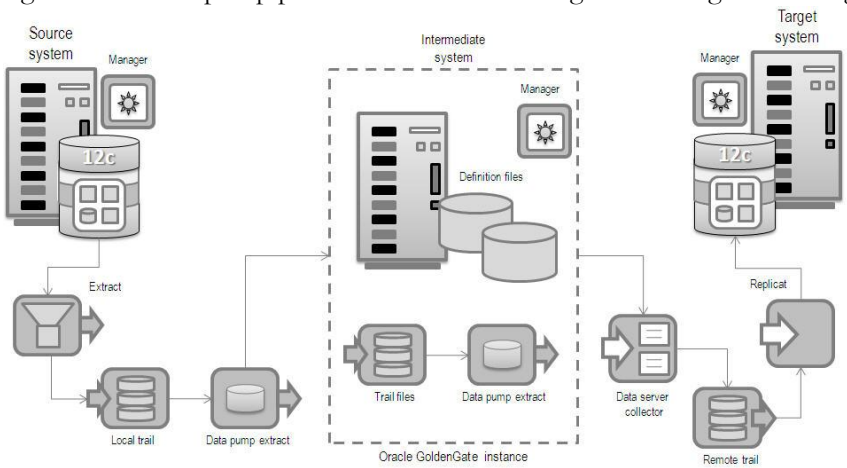


Figure 1-25: Intermediate system for scalability

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.

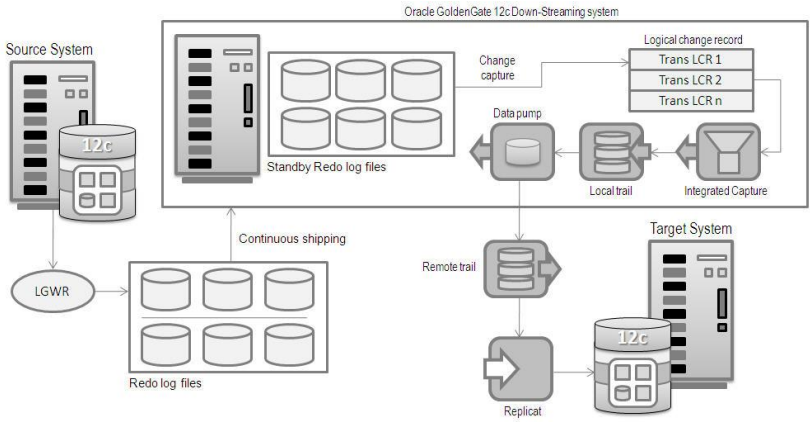


Figure 1-26: Integrated capture architecture

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
DEFSFILE ./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>
```

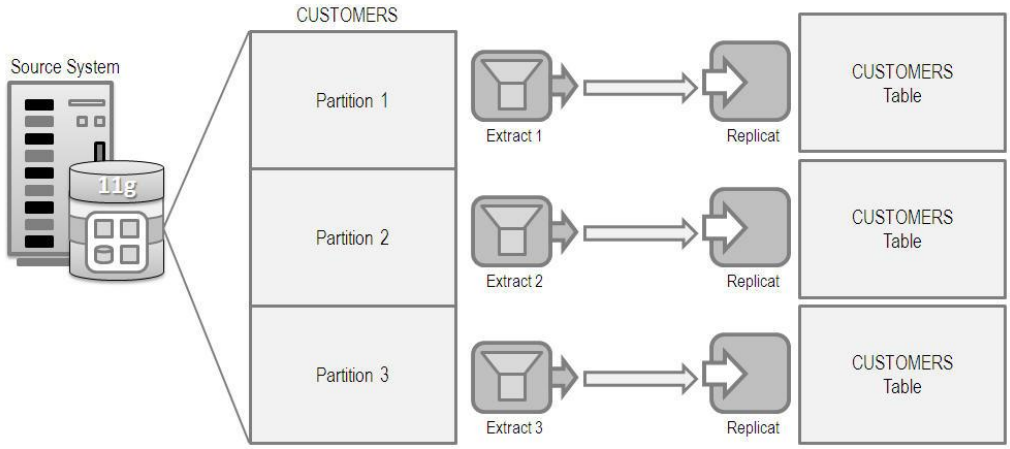


Figure 1-27: Physical private database (PPD)

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)
COMMIT_TS                                          VARCHAR2(128)
POS                                                VARCHAR2(128)
RBA                                                VARCHAR2(128)
TABLENAME                                          VARCHAR2(128)
OPTYPE                                             VARCHAR2(128)

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:  AACAAAAAAAAAAAGAIFAAUDVHCFUGFIYF
Algorithm used:  BLOWFISH

GGSCI (ggs-source) 35>
```

- Test the encrypted password

```
GGSCI (ggs-source) 39> DBLOGIN USERID ggs_admin@S1E2,
PASSWORD AACAAAAAAAAAAAGAIFAAUDVHCFUGFIYF, 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
AACAAAAAAAAAAAGAIFAAUDVHCFUGFIYF, 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
RMTTRAIL /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.

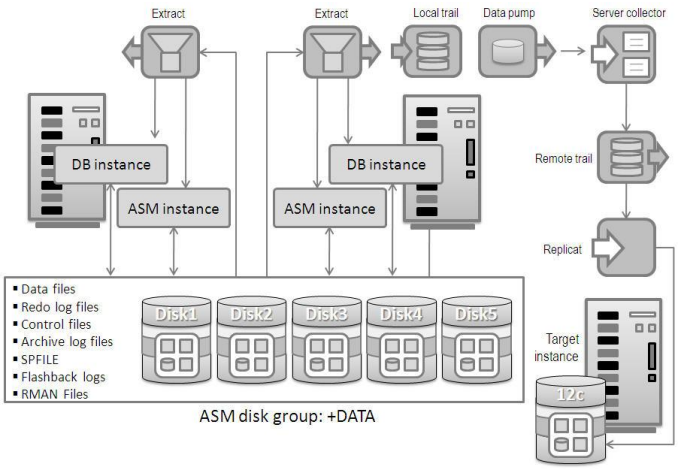


Figure 1-28: Oracle ASM for GoldenGate

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 AACAAAAAAAAAAAGAIFAAUDVHCFUGFIYF,
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 into turn re-assemble the data and write to its own remote trail. Ultimately, using parallel processing maintains stable sub-second latency and lagging.

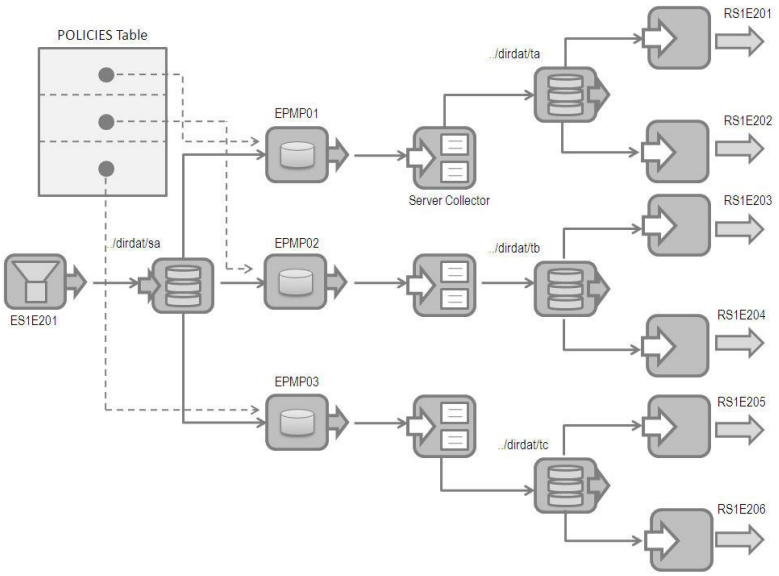


Figure 1-29: Oracle GoldenGate Parallel processing

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 Name	GoldenGate process Type	Memory 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
	Total (GB)	**1.27**

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   AACAAAAAAAAAAAGAIFAAUDVHCFUGFIYF,
ENCRYPTKEY DEFAULT
EXTTRAIL ./dirdat/sa
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
```

-- epmp01.prm

```
EXTRACT epmp01
USERID   ggs_admin@S1E2,   PASSWORD   AACAAAAAAAAAAAGAIFAAUDVHCFUGFIYF,
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   AACAAAAAAAAAAAGAIFAAUDVHCFUGFIYF,
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   AACAAAAAAAAAAAGAIFAAUDVHCFUGFIYF,
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 EXTTRAIL and ADD RMTTRAIL 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@t1c1, 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@t1c1, 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@t1c1, 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

GGSCI (ggs-target) 13>
```

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 ggsmsg.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.

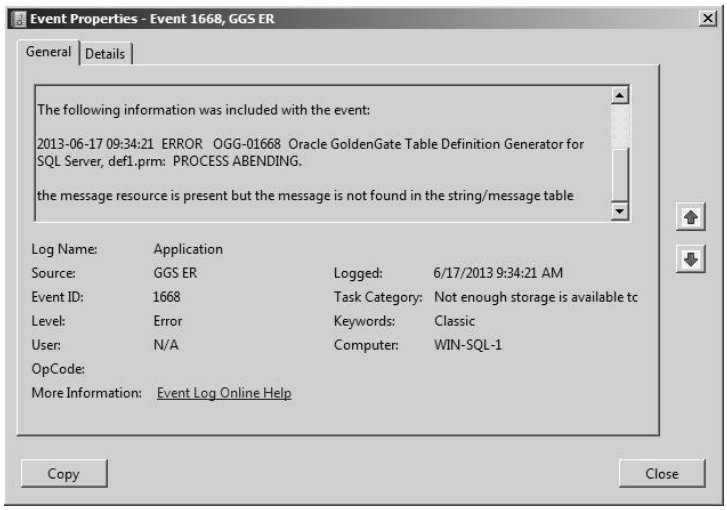


Figure 1-30: MS Windows event viewer for Oracle GoldenGate

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 computername 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

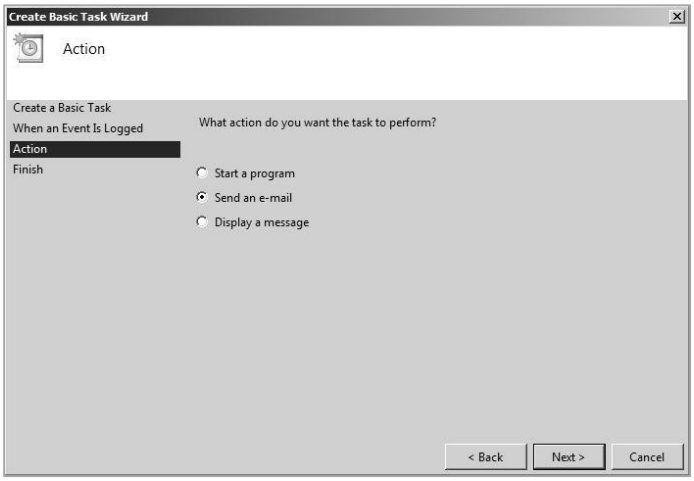


Figure 1-31: Managing and associating an event using an action

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 tcperr, and test using the same network throughput before deploying on production. As unexpected behavior may result due to incorrect settings of delay and retires 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 is 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.

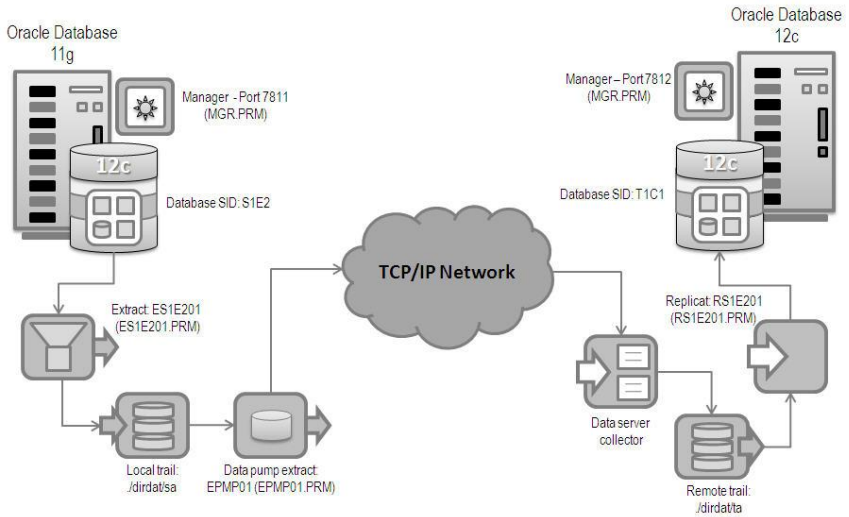


Figure 1-32: Fundamental setup components

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 AACAAAAAAAAAAAGAIFAAUDVHCFUGFIYF,
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 AACAAAAAAAAAAAGAIFAAUDVHCFUGFIYF,
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.

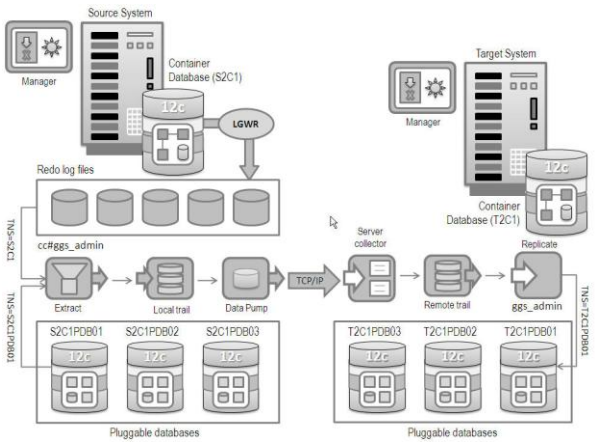


Figure 1-33: Pluggable database Support

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 GLBOALS 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
assumetargetdefs
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.learn-ogg.com to support the workshops' examples.

Good luck, and let's get started with Oracle GoldenGate 12c, *the Oracle bridge.*

Bi-Directional Topology for Business Continuity

APPENDIX B

Overview

The objective of this chapter is to use Oracle GoldenGate 12c two-node configuration for building high availability infrastructure to support business applications maximum availability by minimizing disruption due to system failure. Oracle customers have been implementing Oracle real application clusters (RAC) for high availability and Oracle Data Guard (DG) for disaster recovery. The two solutions co-exist for best continuity. RAC and DG require an advanced level of skills to implement and manage together. RAC and DG are the major components of Oracle Maximum Availability Architecture (MAA) for sites running mission critical systems and requiring a maximum level of scalability, data availability, and data protection. For applications that may compromise RAC embedded scalability and DG fast start failover, Oracle GoldenGate achieves the requirements for business continuity with a sustainable budget, operational cost, and reasonable infrastructure complexity.

The flexibility of Oracle GoldenGate advanced configuration enables the software to provide business continuity similar to RAC and DG combined using active-passive and active-active bi-directional topology architecture. However, this specific implementation is not out-of-the-box and requires a considerable level of Oracle GoldenGate and database experience in respect to the database design and applications coding.

This chapter implements Oracle GoldenGate for business continuity using similarly designed architecture for deploying:

- Active-Passive Model for Disaster Recovery (DR)
- Active-Active Model for High Availability (HA)

Unlike Oracle Real Application Clusters (RAC) in terms of scalability, Oracle GoldenGate does not promote the built-in scalability provided by RAC, which is the core function of RAC to support linearly increasing transactions volume. When scalability is an important element of the application, sufficient evaluation is necessary before deploying Oracle GoldenGate for high availability. Also, with Oracle Real Application Clusters (RAC), increasing the number of nodes delivers higher scalability.

However, Oracle GoldenGate encourages the use of two-node architecture only for high-availability. This is recommended to avoid complex data conflict detection and resolutions in an active-active bi-directional topology.

For an enterprise, Oracle GoldenGate does not replace RAC or DG, rather it is implemented to complement and integrate with the Oracle clusterware and Oracle physical standby database. Refer to Chapter 12 to learn more about integrating Oracle GoldenGate with RAC and DG.

For sites running the Oracle Database Standard Edition, Oracle GoldenGate is an attractive option for implementing high availability (HA) and disaster recovery (DR). The option is cost-effective and highly suitable for in-house developed applications, but needs to be developed from the group-up so that the applications are replication-aware. In deploying Oracle GoldenGate on bi-directional active-active topology, the challenge is to handle data conflicts which are discussed using the following:

- Data and Applications Segmentation
- Oracle GoldenGate Built-In Conflict Detection and Resolution (CDR)
- Custom Conflict Detection and Resolution using PL/SQL Sub-Programs Development
- Oracle Virtual Private Database (VPD)

GoldenGate Business Continuity Architecture

Oracle GoldenGate bi-directional architecture achieves better performance by splitting the load across two instances. This sustains fault tolerance when one of the two instances fail by moving all processing to the surviving system. However, the top advantage of using Oracle GoldenGate bi-directional architecture is the geographically separated data centers that enable cost-effective business continuity.

Figure 5-1 depicts bi-directional topology components needed for business continuity. The data flow is from the primary system (the source) to the live standby system (the target) and vice versa. The primary system database S1E2 is on Oracle Database 11g and the live standby system database T1E2 is on Oracle Database 11g. The Oracle GoldenGate building blocks of extract groups, delivery groups, and related components for data movement are outlined below.

The Primary System (The Source) Components

- Extract groups, the primary extract ES1E203, the local trail ./dirdat/sc and the secondary extract PS1E203.

- The delivery group, the dynamic collector server, the remote trail ./dirdat/tc and the delivery process RS1E203.

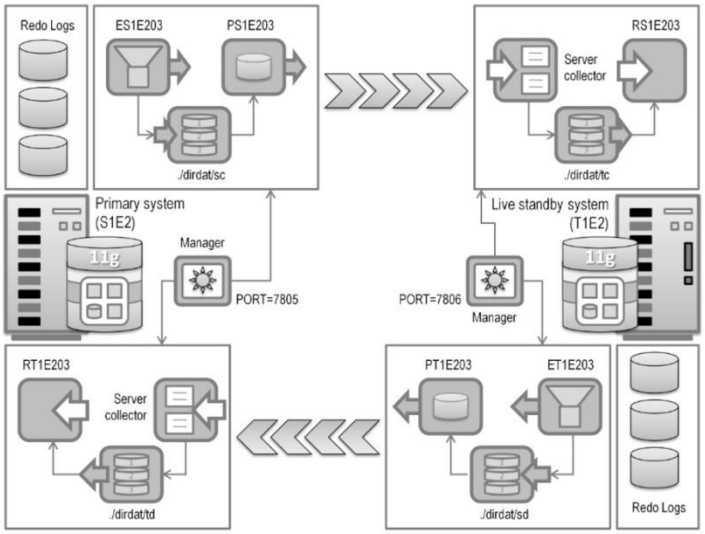


Figure 5-1: Bi-Directional Topology Building Blocks

The Live Standby System (The Target) Components

- Extract groups, the primary extract ET1E203, the local trail ./dirdat/sd and the secondary extract PT1E203.

- The delivery group, the dynamic collector server, the remote trail ./dirdat/td and the delivery process RT1E203.

Database Configuration

The primary system database and live standby database are operating on ARCHIVELOG mode. The transaction logs (online redo log files) structure of both databases are identical. Each has 3 groups used by the primary extract for performing change data capture (CDC). Online redo log files are archived and stored in the fast recovery area (FRA) for a recovery window of (7) days in case of being required by the primary extract or the recovery manager (RMAN) recover operation. The below RUN command backup the archived log files while maintaining a recovery window of (7) days of archived files in the FRA. Deleting files that are not required by the recovery window of (7) days.

```
RMAN> RUN {
2> BACKUP ARCHIVELOG ALL;
```

```
3> DELETE COPY OF ARCHIVELOG ALL COMPLETED BEFORE 'SYSDATE-7';
4> }
```

At minimum, set the size of the recovery window equal to or greater than the value of the database parameter CONTROL_FILE_RECORD_KEEP_TIME. The default is (7) days, unless a recovery catalog is configured. Also, ensuring FRA storage capacity allows retaining (7) days of archived log files by setting the database parameter DB_RECOVERY_FILE_DEST_SIZE as needed.

Extending Oracle GoldenGate for High Availability

When using Oracle GoldenGate for business continuity, the task of migrating users from the primary system to standby system and vice versa requires the database administrator (DBA) intervention and users reconnect to the surviving instance. However, for an active-active environment, the application needs the load balancing technique for connecting sessions to the database with minimal workload. Figure 5-2 shows the load balancer workflow that dynamically routes the application to the database with minimal workload.

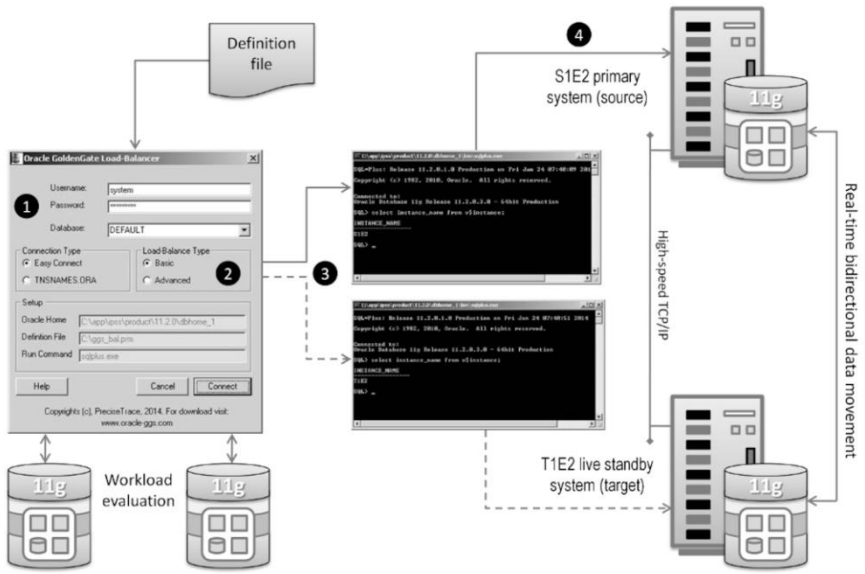


Figure 5-2: High Availability Load Balancer

The load balancer negotiates the primary and standby database to determine the preferred instance based on current workload. Then, it uses the selected connection type to establish the connection to the primary or standby systems – the database

with lesser workload. The load balancer uses a definition file that defines the primary and standby systems. Each line is composed of the host name or IP address, the listener port, the database service name, and the run command. The definition file c:\ggs-bal.prm is mandatory and the database credential must be identical for the primary system and live standby system database. The following lines define the primary and live standby systems configuration.

```
ggs-source   1521   s1e2.precisetrace.com    sqlplus.exe
ggs-target   1521   t1e2.precisetrace.com    sqlplus.exe
```

The above load balancer maintains the workload between the primary and the live-standby systems by performing the following:

1. Connecting to the primary and standby systems for evaluating the current database workload.
2. Comparing the two workloads to determine the preferred database instance – the one with lesser workload.
3. Using the selected connection type to compose the database connect string.
4. Routing and spawning the application to the preferred system.

When using Oracle GoldenGate for high-availability (active-active), the number of Oracle GoldenGate database instances should be limited to a maximum of (3) database instances to avoid complex conflicts resolution and configuration. To download the load balancer utility, visit http://www.learn-ogg.com.

Disaster Recovery

When using Oracle GoldenGate to implement a disaster recovery (DR) site, it is developed as *active-passive*, bi-directional topology architecture. The source database is designated as the primary system and the target database is designated as the live-standby system. The live-standby system database remains on read-write mode. Applications users are connected to either the primary or the live-standby system, but not both. The live-standby system remains ready for the failover and switchover situations. Routing connection to one-and-only-one system is mostly handled from within the application servers as depicted on Figure 5-3 where sessions are established from either the primary or the live-standby system database.

The tremendous advanced features, configurable components flexibility, communication robustness and extendibility of Oracle GoldenGate have technically established a distinctive architecture for designing an enterprise heterogeneous platform suitable for protection and integration the enterprise data. Having a live-standby database continuously synchronized in real-time maximizes the usability of the database for other applications. Another distinctive advantage of Oracle GoldenGate is heterogeneity.

The capability of the live-standby system to run on a dissimilar platform from the primary system attracts decision makers and technical team members to build a sophisticated, one-of-a-kind disaster recovery site. Mostly designed to run on a mid-range Intel-based system powered by Oracle Linux. It delivers high-quality service and low-cost per transaction with best of class stability and reliability.

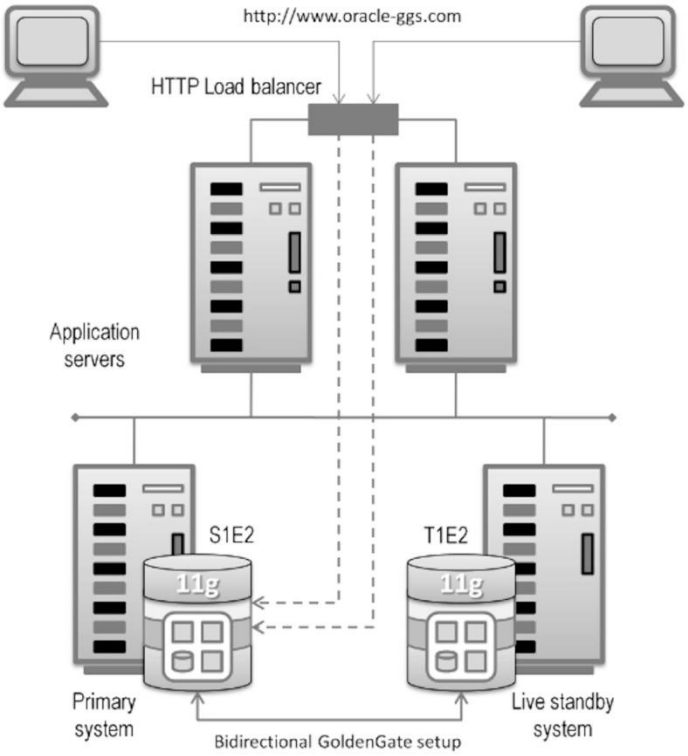


Figure 5-3: Disaster Recovery Connections

Oracle GoldenGate Value Proposition

A value proposition of Oracle GoldenGate is the certification with Oracle Database Standard Edition (SE). Small to mid-range server sites running Oracle Database SE can develop their disaster recovery solution using Oracle GoldenGate only. The demonstration of this chapter was developed using Oracle Database 11g SE running on Oracle Linux. This value proposition should not be underestimated!

```
SQL> SELECT * FROM v$version;

BANNER
----------------------------------------------------------------
Oracle Database 11g Release 11.2.0.3.0 - 64bit Production
```

```
PL/SQL Release 11.2.0.3.0 - Production
CORE    11.2.0.3.0    Production
TNS for Linux: Version 11.2.0.3.0 - Production
NLSRTL Version 11.2.0.3.0 - Production

SQL>
```

Disaster Recovery Considerations

The concept of disaster recovery (DR) imposes the standby database to operate on read-only mode at all times. However, because Oracle GoldenGate requires that the database operates on read-write, it is only the synchronized tables that are restricted from DML operation outside of the Oracle GoldenGate apply session (Replicat). Also, even though the bi-directional setup is already in place, it is active from only one direction at any time. Other guidelines to consider when using Oracle GoldenGate for disaster recovery are listed below:

Live-Standby System

At all times, the data flow is active from only one direction. The live-standby database (target) remains on read-write mode but fundamentally inactive – used only by the Replicat sessions. Under normal situations, OLTP users are connected to the primary system database (source) only. Utilizing the live-standby system for other applications should avoid interaction with mapped tables used by Oracle GoldenGate delivery processes (Replicats).

Operating Mode and Failover

The live-standby system database should be operated on ARCHIVELOG mode. This prepares the live-standby system for a planned switchover and unplanned failover. Oracle GoldenGate switchover and failover is an interactive operation and handled manually. Using tools such as Oracle GoldenGate Veridata and the Oracle GoldenGate Monitor are highly recommended to proactively avoid data divergence and minimizing latency.

Query Offloading

OLAP users are normally directed to the live-standby system database (target). It is the ideal platform for running long query reports, but the database should not be modified by creating additional objects such as indexes and materialized views. Otherwise, these created objects must be dropped as part of a failover/switchover operation.

Unsupported Data Types

The unsupported data types by Oracle GoldenGate is minimum. The most common applications data types not supported by Oracle GoldenGate is BFILE, which handled using the PL/SQL supplied API DBMS_LOB and DBMS_FILE_TRANSFER.
Refer to Chapter 11 to learn more about using these packages with Oracle GoldenGate.
Evaluating the extent of unsupported data types is necessary to qualify using Oracle GoldenGate for disaster recovery solution. Refer to Chapter 1 to learn more about unsupported data types.

Disaster Recovery Implementation

A disaster recovery implementation using Oracle GoldenGate consists of the tasks listed in Table 5-1. Testing switchover and failover for moving users from the primary system to the live-standby system and then moving users back to the primary system is the last step and executed for planned & unplanned scenarios.

Task	
1.	Install software, configure manager process, and start Oracle GoldenGate instance.
2.	Enable database supplemental logging and archive log mode.
3.	Create Oracle GoldenGate database user and grant privileges.
4.	Login to the database from GGSCI and Add table-level logging.
5.	Prepare GLOBALS parameters and the checkpoint table.
6.	Perform initial load from primary (source) to live standby (target) systems.
7.	Configure, create, and deploy Oracle GoldenGate groups for bi-directional topology.
8.	Enable data capture and delivery from primary to live standby system.
9.	Perform moving users for planned & unplanned switchover and failover.

Table 5-1: Bi-Directional Configuration for High Availability and Disaster Recovery Solutions

The destination for each of the above tasks is shown in Figure 5-4. Since this is a bi-directional topology architecture, the tasks are performed on the primary system/source database and on the live-standby system/target database.

The live-standby system should have adequate resources to support the post-failover/ switchover operation timeframe. Having matched resources is recommended.
Even though Oracle GoldenGate supports a heterogeneous database environment, it is absolutely not recommended when implementing high availability and disaster recovery solutions using Oracle GoldenGate. The primary and live-standby platform should have identical architecture. The planned switchover and the unplanned failover is a manual process. Automation is developed using Linux shell script and Oracle GoldenGate obey files. The use of Oracle GoldenGate Veridata for data consistency and the Oracle GoldenGate Monitor are highly recommended for such an environment.

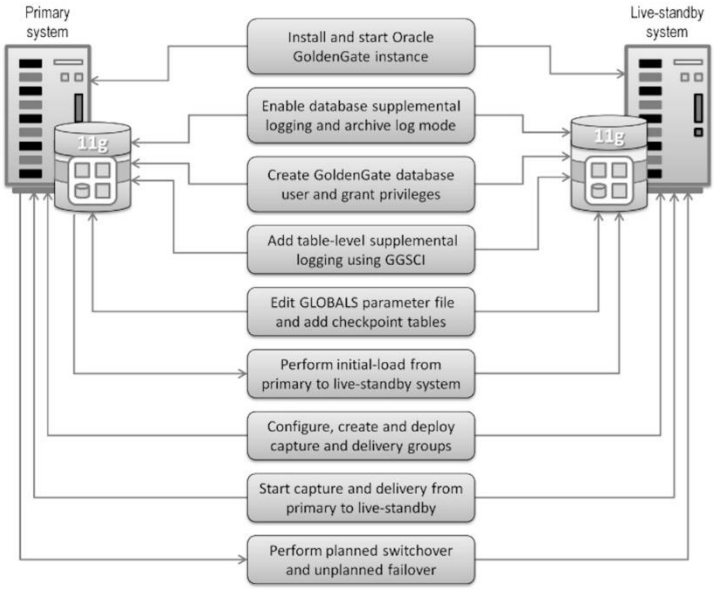


Figure 5-4: Bi-Directional Topology Architecture Tasks vs. Destinations

Software Installation

Successful installation of Oracle GoldenGate on the primary and live-standby systems must be successfully completed to proceed further. Consider the platform architecture. The chapter uses Oracle GoldenGate for Oracle Database 11g 64-bit. The same bundle of Oracle GoldenGate 12c provides support for Oracle database enterprise and standard editions. Refer to Chapter 2 to learn about installing Oracle GoldenGate.

Supplemental Logging

Enable supplemental logging on the primary and live-standby databases. Also, ensure the databases are operating on ARCHIVELOG mode.

```
SQL> CONN / AS SYSDBA
Connected.
SQL> SELECT log_mode FROM v$database;

LOG_MODE
------------
ARCHIVELOG

SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;

Database altered.
```

```
SQL>
```

Refer to Chapter 2 to learn more about database supplemental logging settings. Even though the data flow is in one direction at any time, ensuring the database supplemental logging and ARCHIVELOG mode are enabled on the primary and Live-standby systems minimizes the meantime to migrate users and reverse the processing direction.

Database Users

Create Oracle GoldenGate database user on the primary and live-standby databases. The use of dedicated Oracle GoldenGate is required for active-active bi-directional topology. It is used for an endless loop detection purpose as explained in this chapter.

```
SQL> CREATE USER ggs_admin identified by oracle;

User created.

SQL> GRANT DBA TO ggs_admin;

Grant succeeded.

SQL> CONN ggs_admin/oracle
Connected.
SQL>
```

Refer to Chapter 2 to learn more about the precise database privileges granted to Oracle GoldenGate users on the primary (source) and live-standby (target) database. The above command grants DBA which is considered an over grant and not recommended or even possible for some sites.

Start Oracle GoldenGate Instances

The primary system and live-standby system are running on Oracle Database 11g Standard Edition 64-bit. The primary system Oracle GoldenGate instance manager process is listening on port 7805 and the live-standby Oracle GoldenGate manager process is listening on port 7806. Using the GoldenGate Software Command Interface (GGSCI), configure and start the manager process on the primary and live-standby systems.

```
GGSCI (apps-srv1) 6> START MGR
Manager started.

GGSCI (apps-srv1) 7> INFO MGR

Manager is running (IP port apps-srv1.7805, Process ID 13632).

GGSCI (apps-srv1) 8>
```

Repeat the above GGSCI commands for the live-standby system.

```
GGSCI (apps-srv2) 2> START MGR
Manager started.

GGSCI (apps-srv2) 3> INFO MGR

Manager is running (IP port apps-srv2.7806, Process ID 30133).

GGSCI (apps-srv2) 4>
```

Refer to Chapter 2 to learn more about the manager process common parameters settings for managing trails and unattended restart.

Checkpoint Table

The database checkpoint table is the default for the Replicat process checkpoint. Prepare the GLOBALS parameter file, then use GGSCI to create the checkpoint table. The GLOBALS parameter file contains the following one line only. Where GGS-ADMIN is the Oracle GoldenGate database schema.

```
CHECKPOINTTABLE ggs-admin.chkpttab

GGSCI (apps-srv1) 1> DBLOGIN USERID ggs_admin, PASSWORD oracle
Successfully logged into database.

GGSCI (apps-srv1) 2> ADD CHECKPOINTTABLE

No checkpoint table specified. Using GLOBALS specification
(ggs_admin.chkpttab)...

Successfully created checkpoint table ggs_admin.chkpttab.

GGSCI (apps-srv1) 3>
```

Create the checkpoint table on primary system and live-standby systems. Remember to exit and restart GGSCI to activate parameters added to GLOBALS parameter file.

Performing Initial-Data Load

This is a major task and must be delivered 100% successfully. The live-standby system database must be established prior to beginning the configuration tasks. Even though this demonstration is from Oracle-to-Oracle database (homogenous), file to Replicat Oracle GoldenGate initial-data load method is used to establish the live-standby (target) database. Since we consider the initial load from an inactive database, the initial load is performed using the following steps:

- Configure and start the initial load Extract (ELOAD03)

- Configure the initial load Replicat (RLOAD03)
- Start the initial load Replicat (RLOAD03)

Initial Load Extract

Before starting the initial load extract, some analysis must be conducted to determine any unsupported column data types. Since this configuration is for developing a disaster recovery (DR), all tables must be included in the capture groups. For a very large list of tables, use the following PL/SQL code or the Extract and Replicat wild-card syntax format with exclude option to prepare the ELOAD03.PRM file.

```
SQL> CREATE DIRECTORY ggs_dir as '/d01/oracle/ggs/dirprm';

Directory created.

SQL> GRANT ALL ON DIRECTORY ggs_dir to public;

Grant succeeded.

SQL> DECLARE
  2      filehandler UTL_FILE.FILE_TYPE;
  3      CURSOR c1 is SELECT table_name
  4                   FROM   dba_tables
  5                   WHERE  owner='OSM$REPAPI';
  6      vtable_name VARCHAR2(30);
  7  BEGIN
  8      fileHandler := UTL_FILE.FOPEN('GGS_DIR', 'eload03.prm', 'W');
  9      UTL_FILE.PUTF(fileHandler, 'SOURCEISTABLE\n');
 10      UTL_FILE.PUTF(fileHandler, 'USERID ggs_admin, PASSWORD oracle\n');
 11      UTL_FILE.PUTF(fileHandler, 'RMTHOST apps-srv2, MGRPORT 7806\n');
 12      UTL_FILE.PUTF(fileHandler, 'RMTFILE ./dirdat/initload01.dat, PURGE\n');
 13      OPEN c1;
 14      LOOP
 15         FETCH c1 into vtable_name;
 16         EXIT WHEN c1%NOTFOUND;
 17         UTL_FILE.PUTF(fileHandler, 'TABLE osm$repapi.'||vtable_name||';\n');
 18      END LOOP;
 19      CLOSE c1;
 20      UTL_FILE.FCLOSE(fileHandler);
 21  EXCEPTION
 22      WHEN utl_file.invalid_path THEN
 23        raise_application_error(-20000, 'ERROR: Invalid GoldenGate path.');
 24  END;
 25  /

PL/SQL procedure successfully completed.

SQL>
```

```
GGSCI (apps-srv1) 1> EDIT PARAMS eload03
```

```
SOURCEISTABLE
USERID ggs_admin, PASSWORD oracle
```

```
RMTHOST apps-srv2, MGRPORT 7806
RMTFILE ./dirdat/initload03.dat, PURGE
TABLE osm$repapi.POLICY TYPES;
TABLE osm$repapi.DISCOUNT_TYPES;
TABLE osm$repapi.CUSTOMERS;
TABLE osm$repapi.POLICIES;
```

```
GGSCI (apps-srv1) 1> SHELL ./extract pf ./dirprm/eload03.prm rf
./dirrpt/eload03.rpt
```

The GGSCI SHELL command allows you to invoke programs and operating system shell commands from within Oracle GoldenGate. The parameter SOURCEISTABLE directly reads records from the database tables, and must be the first parameter in the configuration file.

Initial Load Replicat

The initial load data file has been created on the live-standby system. Verify the existence of ./dirdat/initload03.dat on the live standby. Before starting the Replicat program, create a special run Replicat process (RLOAD03) as shown below.

```
GGSCI (apps-srv2) 1> ADD REPLICAT rload03, SPECIALRUN
REPLICAT added.

GGSCI (apps-srv2) 2>
```

```
GGSCI (apps-srv2) 2> EDIT PARAMS rload03
```

```
REPLICAT rload03
SPECIALRUN
ASSUMETARGETDEFS
HANDLECOLLISIONS
USERID ggs_admin, PASSWORD oracle
EXTFILE ./dirdat/initload03.dat
DISCARDFILE ./dirrpt/rload03.dsc, PURGE
MAP osm$repapi.*, TARGET osm$repapi.*;
END RUNTIME
```

```
GGSCI (apps-srv2) 1> SHELL ./replicat pf ./dirprm/rload03.prm rf
./dirrpt/rload03.rpt
```

Now, the live-standby database is established and identical to the primary database. Before proceeding further, verify ascertain the data using Oracle GoldenGate Veridata. Also, because the initial-load is done from an inactive database, review Oracle GoldenGate report generated from the initial-load Replicat process.

```
GGSCI (apps-srv2) 4> VIEW REPORT ./dirrpt/rload03.rpt
```

```
Reading ./dirdat/initloa3.dat, current RBA 37987353, 200007 records

Report at 2014-01-04 17:27:04 (activity since 2014-01-04 17:26:08)

From Table OSM$REPAPI.POLICY_TYPES to OSM$REPAPI.POLICY_TYPES:
       #                    inserts:         4
       #                    updates:         0
       #                    deletes:         0
       #                   discards:         0
From Table OSM$REPAPI.DISCOUNT_TYPES to OSM$REPAPI.DISCOUNT_TYPES:
       #                    inserts:         3
       #                    updates:         0
       #                    deletes:         0
       #                   discards:         0
From Table OSM$REPAPI.CUSTOMERS to OSM$REPAPI.CUSTOMERS:
       #                    inserts:    100000
       #                    updates:         0
       #                    deletes:         0
       #                   discards:         0
From Table OSM$REPAPI.POLICIES to OSM$REPAPI.POLICIES:
       #                    inserts:    100000
       #                    updates:         0
       #                    deletes:         0
       #                   discards:         0

Last log location read:
     FILE:      ./dirdat/initload03.dat
     RBA:       37987353
     TIMESTAMP: 2014-01-04 14:24:58.801283
     EOF:       NO
     READERR:   400
```

As per the report details, the initial load has been completed successfully. Refer to Chapter 3 for more details regarding initial data load techniques and options.

Bi-Directional Topology Configuration

The basic task of configuring Oracle GoldenGate to support a disaster recovery site is to have a working bi-directional data flow, from the primary system to the live-standby system and vice versa. The purpose of the basic configuration is to test the capture, route, and delivery groups on a bi-directional environment using automated database transactions developed in PL/SQL and run from the command line. Figure 5-5 depicts Oracle GoldenGate groups that are configured next.

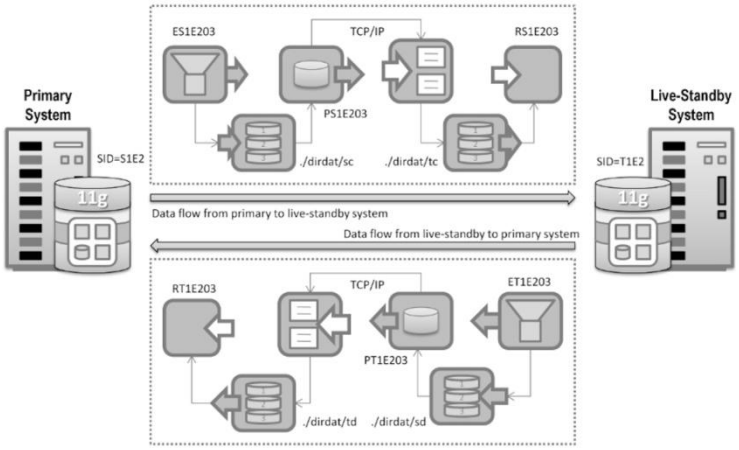


Figure 5-5: Bi-Directional Capture and Delivery Groups

After completing the bi-directional configuration, run and verify the transactional testing from both sides. The following procedure TRANS_LOAD is run as an anonymous PL/SQL block. To download the TRANS_LOAD source code, visit http://www.learn-ogg.com.

```
SQL> BEGIN
  2     FOR i IN 100901..101000 LOOP
  3         trans_load(1, i);
  4     END LOOP;
  5  END;
  6  /

PL/SQL procedure successfully completed.

SQL>
```

Since the procedure TRANS_LOAD does not handle primary key collision, ensure the lower and upper range of the loop statement is correctly bounded to avoid primary key violation error.

Intermediate Systems

The use of an intermediate system between the primary and live-standby systems is an option. This configuration is only recommended when the current active system needs to ship trail files to a third system, not really part of this high availability bi-directional topology. Other reasons for using an intermediate system is to offload the current active system (primary system) from routing trail records to the remote system, or network security prevents either system from sending trail records, and the need of a

dedicated system for sending trail to the third system regardless of which system (primary or live-standby) is active. The disadvantage is increasing the latency, which increases the likelihood of data conflicts. Refer to Chapter 12 for details regarding intermediate systems.

Primary to Standby Data Flow

The configuration of the primary (source) to the live-standby (target) systems data flow is composed of deploying the primary & secondary extract groups and a delivery group. Figure 5-6 shows the components configured to move data from the primary to the live-standby system.

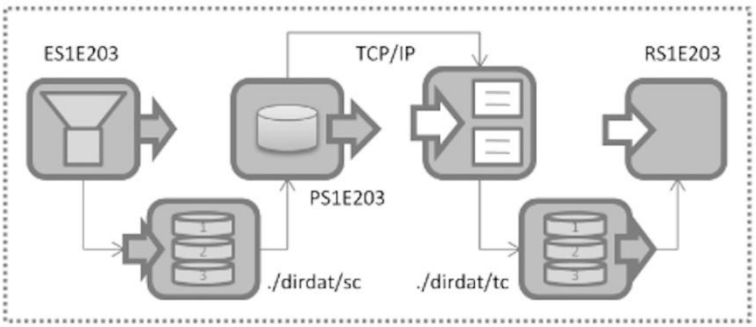


Figure 5-6: Primary to Standby Data Flow

Primary Extract Group

The primary group is composed of the parameter file ES1E203.PRM, the primary extract process ES1E203, the local trail ./dirdat/sc, and the checkpoint file ./dirchk/ES1E203.CPE. The primary extract ES1E203 is configured to capture from the transaction logs and writes to the local trail. The next two commands create the primary extract and its local trail. The local trail files is sized 128MB.

```
GGSCI (apps-srv1) 2> EDIT PARAMS es1e203
```

```
EXTRACT es1e203
SETENV  (ORACLE SID=S1E2)
SETENV  (ORACLE_HOME="/d01/app/oracleSE")
SETENV  (NLS LANG=AMERICAN AMERICA.AR8ISO8859P6)
USERID  ggs_admin, password oracle
EXTTRAIL ./dirdat/sc
TABLE osm$repapi.CUSTOMERS;
TABLE osm$repapi.POLICIES;
```

```
GGSCI (apps-srv1) 3> ADD EXTRACT es1e203, TRANLOG, BEGIN now, THREADS 1
EXTRACT added.
```

```
GGSCI (apps-srv1) 5> ADD EXTTRAIL ./dirdat/sc, EXTRACT es1e203, MEGABYTES 128
EXTTRAIL added.

GGSCI (apps-srv1) 6>
```

Secondary Extract Group

The secondary group is composed of the parameter file PS1E203.PRM, the data pump extract process PS1E203, the remote trail ./dirdat/tc, and the checkpoint file ./dirchk/PS1E203.CPE. The data pump extract PS1E203 sends data over TCP/IP to the live-standby system. The next commands create the secondary group (data pump) and its remote trail. The remote trail files is sized 128MB.

```
GGSCI (apps-srv1) 6> EDIT PARAMS ps1e203
```

```
GGSCI (apps-srv1) 6> EDIT PARAMS ps1e203 EXTRACT ps1e203
USERID ggs_admin, PASSWORD oracle
PASSTHRU
RMTHOST apps-srv2, MGRPORT 7806
RMTTRAIL ./dirdat/tc
TABLE osm$repapi.CUSTOMERS;
TABLE osm$repapi.POLICIES;
```

```
GGSCI (apps-srv1) 7> ADD EXTRACT ps1e203, EXTTRAILSOURCE ./dirdat/sc
EXTRACT added.
```

```
GGSCI (apps-srv1) 11> ADD RMTTRAIL ./dirdat/tc, EXTRACT ps1e203, MEGABYTES 128
RMTTRAIL added.

GGSCI (apps-srv1) 12>
```

Delivery Group

The delivery group is composed of the parameter file RS1E203.PRM, the Replicat process PS1E203, the remote trail ./dirdat/tc, and the checkpoint file ./dirchk/ES1E203.CPR. The delivery process uses the live-standby database SQL Native Call to apply transactions to the database.

```
GGSCI (apps-srv2) 2> ADD REPLICAT rs1e203, EXTTRAIL ./dirdat/tc
REPLICAT added.

GGSCI (apps-srv2) 3>
```

Standby to Primary Data Flow

To support switchover and failover operations for planned and unplanned modes,

the data flow from the standby to the primary systems must be in place. Figure 5-7 shows the data flow from the live-standby to the primary system.

The live-standby to primary data flow configuration remains inactive. It is activated only when users are moved from the primary system to the live-standby system. The application servers, where the applications are running, are configured to connect to one and only one specific system – either the primary system or the live-standby system, but not both.

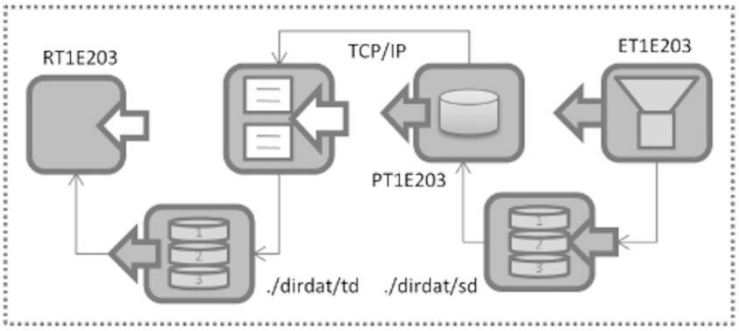


Figure 5-7: Standby to Primary Data Flow

Primary Extract Group

The primary group is composed of the parameter file ET1E203.PRM, the primary extract process ET1E203, the local trail ./dirdat/sd and the checkpoint file ./dirchk/ET1E203.CPE. The next commands create the primary extract and its local trail. The local trail files sized 128MB.

```
GGSCI (apps-srv2) 1> EDIT PARAMS et1e203
```

```
EXTRACT et1e203
SETENV (ORACLE SID="T1E2")
SETENV (ORACLE_HOME="/d01/app/oracleSE/product/11.2.0/dbhome_1")
SETENV (NLS_LANG="AMERICAN_AMERICA.AL32UTF8")
USERID ggs_admin, PASSWORD oracle
EXTTRAIL ./dirdat/sd
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
```

```
GGSCI (apps-srv2) 5> ADD EXTRACT et1e203, TRANLOG, BEGIN now, THREADS 1
EXTRACT added.
```

```
GGSCI (apps-srv2) 6> ADD EXTTRAIL ./dirdat/sd, EXTRACT et1e203, MEGABYTES 128
EXTTRAIL added.

GGSCI (apps-srv2) 7>
```

Secondary Extract Group

The secondary group is composed of the parameter file PT1E203.PRM, the data pump extract process PT1E203, the remote trail ./dirdat/td and the checkpoint file ./dirchk/PT1E203.CPE. The next commands create the secondary group (data pump) and its remote trail. The remote trail files sized 128MB.

```
GGSCI (apps-srv2) 7> EDIT PARAMS pt1e203
```

```
EXTRACT pt1e203
USERID ggs_admin, PASSWORD oracle
PASSTHRU
RMTHOST apps-srv1, MGRPORT 7805
RMTTRAIL ./dirdat/td
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
```

```
GGSCI (apps-srv2) 9> ADD EXTRACT pt1e203, EXTTRAILSOURCE ./dirdat/sd
EXTRACT added.
```

```
GGSCI (apps-srv2) 11> ADD RMTTRAIL ./dirdat/td, EXTRACT pt1e203, MEGABYTES 128
RMTTRAIL added.

GGSCI (apps-srv2) 12>
```

Delivery Group

The delivery group is composed of the parameter file RT1E203.PRM, the Replicat process PT1E203, the local trail ./dirdat/sd and the checkpoint file ./dirchk/ET1E203.CPR for reporting. The Replicat process ES1E203 uses a checkpoint table, which is the default location for checkpoint. The next command creates the delivery group (Replicat).

```
GGSCI (apps-srv1) 3> EDIT PARAMS rt1e203
```

```
REPLICAT rt1e203
SETENV (ORACLE_SID=S1E2)
SETENV (NSL_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rt1e203.dsc, PURGE
MAP osm$repapi.customers, TARGET osm$repapi.customers;
MAP osm$repapi.policies, TARGET osm$repapi.policies;
```

```
GGSCI (apps-srv1) 4> ADD REPLICAT rt1e203, EXTTRAIL ./dirdat/td
REPLICAT added.

GGSCI (apps-srv1) 5>
```

This concludes Oracle GoldenGate bi-directional configuration. Under normal business operation, the data flow is always from the primary system (source) to the live-standby system (target). Users are temporary migrated to the live-standby system either due to switchover or failover operation.

Switchover and Failover

Moving users from the primary to the standby site is performed as a planned (switchover) or unplanned (failover) situation. A switchover is for well-planned specific system maintenance tasks such as primary site database upgrade, standby operating system upgrade, etc. The following sections describe the process workflow for a primary site protected by one live standby site, which may need to perform either a switchover or failover.

A switchover consists of the following tasks:

- Switchover for planned primary to standby site
- Moving users back to the primary site

A failover consists of the following tasks:

- Failover for unplanned primary to standby site
- Re-establish the primary site
- Moving users back to the primary site

During the switchover and failover operations, the primary system is referred to by (P) and the live standby system is referred to by (S). This distinction avoids executing commands on the wrong Oracle GoldenGate instance or database.

Planned Primary to Live-Standby Switchover Workflow

Database and system administrators plan and perform system maintenance tasks. For systems operating 24/7, temporarily moving users from the primary system to the live-standby system enables business continuity by sustaining the minimum downtime while ensuring data integrity. Figure 5-8 depicts the steps to move users from the primary to the live-standby system.

The two common maintenance task benefits from a planned switchover are:

- Database Upgrades
- Operating System Upgrades

Due to less resource availability on the live-standby system, the switchover is temporary. Users are moved back to the primary system after successfully performing the maintenance tasks.

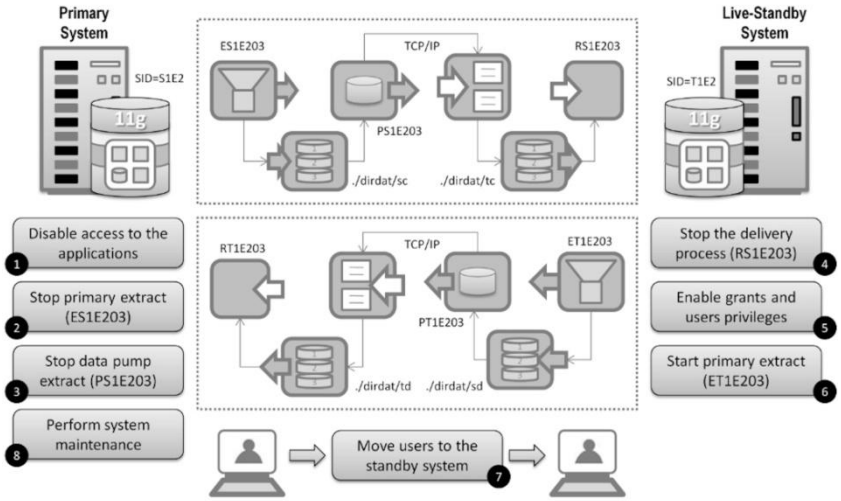


Figure 5-8: Planned Switchover from Primary to Standby System

Tasks performed on the primary system are referred to as (P), whereas tasks performed on the live-standby system referred to as (S).

Disable Access to the Applications (P)

For web-based applications accessing the database, shutdown of the application servers disables access to the database. To ascertain that no transactions were applied to the database during the switchover, after shutdown of the application servers, place the primary database on restricted sessions as shown below.

```
SQL> CONN / AS SYSDBA
Connected.
SQL> ALTER SYSTEM ENABLE RESTRICTED SESSION;

System altered.

SQL> GRANT RESTRICTED SESSION to ggs_admin;

Grant succeeded.

SQL> SELECT logins
  2  FROM   v$instance;

LOGINS
----------
RESTRICTED
```

The primary and secondary extracts are still running to complete capture and routing of the transaction after the shutdown of the application servers. After finishing Step 8 of Figure 5-8, revoke the restricted session privilege from the Oracle GoldenGate database user and disable the database restricted session.

```
SQL> REVOKE RESTRICTED SESSION FROM ggs_admin;

Revoke succeeded.

SQL> ALTER SYSTEM DISABLE RESTRICTED SESSION;

System altered.

SQL>
```

Stop the Primary Extract (P)

Before stopping the primary extract (ES1E203), issue the LAG EXTRACT command until it returns the message "At EOF, no more records to process." This indicates the primary extract has processed the transactions logs and is ready to issue the STOP EXTRACT command.

```
GGSCI (apps-srv1) 2> LAG EXTRACT es1e203

Sending GETLAG request to EXTRACT ES1E203 ...
Last record lag 1 seconds.
At EOF, no more records to process.

GGSCI (apps-srv1) 3> STOP EXTRACT es1e203

Sending STOP request to EXTRACT ES1E203 ...
Request processed.

GGSCI (apps-srv1) 4>
```

Stop the Data Pump Extract (P)

Run the LAG EXTRACT command until it returns "At EOF, no more records to process." This indicates the data pump has sent all transactions to the live-standby system. Now, issue the STOP EXTRACT command.

```
GGSCI (apps-srv1) 5> LAG EXTRACT ps1e203

Sending GETLAG request to EXTRACT PS1E203 ...
Last record lag 4 seconds.
At EOF, no more records to process.

GGSCI (apps-srv1) 6> STOP EXTRACT ps1e203
```

```
Sending STOP request to EXTRACT PS1E203 ...
Request processed.

GGSCI (apps-srv1) 7>
```

Stop the Delivery Process (S)

Run the command LAG REPLICAT until it returns the message "At EOF, no more records to process." This indicates that the delivery process has applied all of the transaction to the live-standby database. Now, issue the STOP REPLICAT command.

```
GGSCI (apps-srv2) 2> LAG REPLICAT rs1e203

Sending GETLAG request to REPLICAT RS1E203 ...
Last record lag 7 seconds.
At EOF, no more records to process.

GGSCI (apps-srv2) 3> STOP REPLICAT rs1e203

Sending STOP request to REPLICAT RS1E203 ...
Request processed.

GGSCI (apps-srv2) 4>
```

Enable Grants and Constraints (S)

This step is site specific. To minimize the downtime, the database DDL commands to grant and enable constraints that should be developed prior to initiating the switchover.

Start the Primary Extract (S)

Prior to moving users to the standby system, run the ALTER EXTRACT command to change the BEGIN timestamp. Then, start the primary extract only. Do not start the data pump extract. Ensure the storage level for the local trail is preserved to handle the primary system maintenance timeframe.

```
GGSCI (apps-srv2) 6> ALTER EXTRACT et1e203, BEGIN now
EXTRACT altered.

GGSCI (apps-srv2) 7> START EXTRACT et1e203

Sending START request to MANAGER ...
EXTRACT ET1E203 starting

GGSCI (apps-srv2) 8>
```

Since the manager process is configured to handle local trail space management when the trail files are shipped, they are automatically purged and keep disk space

level optimized.

Switch User to the Live-Standby System

Run the necessary commands (scripts) to switchover the application servers to connect to the standby system. This enables the user to connect and perform transactions on the standby system.

Perform System Maintenance (P)

The time incurred to complete the system maintenance tasks impacts the storage level on the live-standby system. Performing the system maintenance on a separate test system minimizes the downtime of the primary system and reveals site specific issues.

The downtime for performing a planned switchover from the primary system to the live-standby system depends on Steps 5 & 7. These are the steps to run the database DDL commands to grant privileges, enable constraints, and start the application servers respectively. To successfully complete Steps 1 –7, it will take an estimated 10-30 minutes, when the commands are well-tested in the form script files.

Planned Standby to Primary Switchover Workflow

After completing the maintenance tasks on the primary system, a planned switchover from the live-standby to the primary system is performed. Figure 5-9 depicts the workflow steps to move users back to the primary system. Ascertain the current active database instance for a connected session using the following SQL statement:

```
SQL> CONN / AS SYSDBA
Connected.
SQL> COLUMN INSTANCE FORMAT A10
SQL> COLUMN DATABASE FORMAT A10
SQL> SELECT SYS_CONTEXT('USERENV', 'INSTANCE_NAME') INSTANCE,
  2         SYS_CONTEXT('USERENV', 'DB_NAME') DATABASE
  3  FROM   dual;

INSTANCE   DATABASE
---------- ----------
S1E2        S1E2

SQL>
```

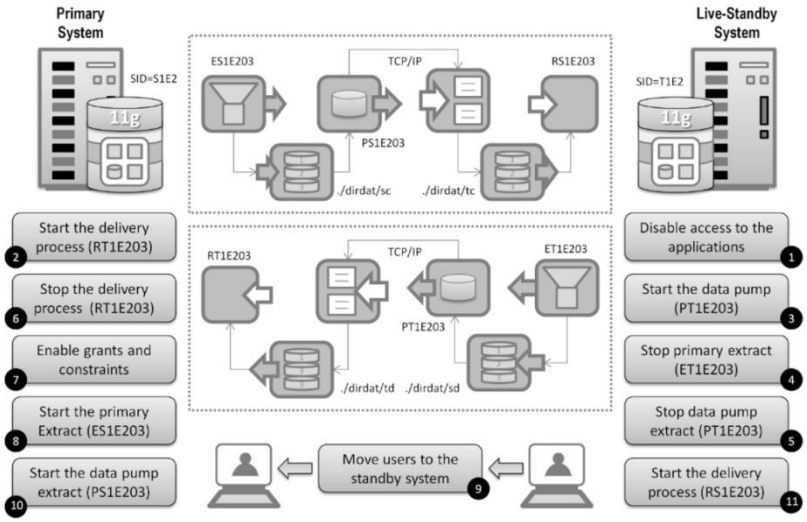


Figure 5-9: Planned Switchover from Live Standby to Primary System

Disable Application Access (S)

Start disabling users access to the applications by a shutdown of the application servers. The primary extract on the standby system remains running to capture the end of the transactions logs.

Start the Delivery Process (P)

To apply the transaction that occurred during the maintenance timeframe, start the delivery process (RT1E203).

```
GGSCI (apps-srv1) 9> START REPLICAT rt1e203

Sending START request to MANAGER ...
REPLICAT RT1E203 starting

GGSCI (apps-srv1) 10>
```

Start the Data Pump (S)

Since the data pump process stopped for the duration of the maintenance on the primary system, it needs to be started to send the transactions to the primary system.

```
GGSCI (apps-srv2) 9> START EXTRACT pt1e203
```

```
Sending START request to MANAGER ...
EXTRACT PT1E203 starting
```

Stop the Primary Extract (S)

While the applications are not running, keep issuing the command LAG EXTRACT until the message "At EOF, no more records to process" returns. This indicates all transaction logs are captured.

```
GGSCI (apps-srv2) 10> LAG EXTRACT et1e203

Sending GETLAG request to EXTRACT ET1E203 ...
No records yet processed.
At EOF, no more records to process.

GGSCI (apps-srv2) 11> STOP EXTRACT et1e203

Sending STOP request to EXTRACT ET1E203 ...
Request processed.

GGSCI (apps-srv2) 12>
```

Stop the Data Pump Extract (S)

Repeatedly issue the command LAG EXTRACT until it returns the message "At EOF, no more records to process." This indicates all transaction have been sent to the primary system. Then, issue the command STOP EXTRACT.

```
GGSCI (apps-srv2) 12> LAG EXTRACT pt1e203

Sending GETLAG request to EXTRACT PT1E203 ...
No records yet processed.
At EOF, no more records to process.


GGSCI (apps-srv2) 13> STOP EXTRACT pt1e203

Sending STOP request to EXTRACT PT1E203 ...
Request processed.

GGSCI (apps-srv2) 14>
```

Stop the Delivery Process (P)

The next step is to make sure the data processed during the maintenance is applied to the primary database. Keep issuing the command LAG REPLICAT until it returns "At EOF, no more records to process." This indicates all transactions have been applied to the primary database.

```
GGSCI (apps-srv1) 10> LAG REPLICAT rt1e203

Sending GETLAG request to REPLICAT RT1E203 ...
No records yet processed.
At EOF, no more records to process.

GGSCI (apps-srv1) 11> STOP REPLICAT rt1e203

Sending STOP request to REPLICAT RT1E203 ...
Request processed.

GGSCI (apps-srv1) 12>
```

Enable Grants and Constraints (P)

The primary system is now ready for switchover. Before doing so, run the database DDL command to grant privileges and enable constraints.

Start the Primary Extract (P)

Start the primary extract to capture from the transaction logs.

```
GGSCI (apps-srv1) 12> START EXTRACT es1e203

Sending START request to MANAGER ...
EXTRACT ES1E203 starting
```

Switch User Activity

Starting the application servers allows users to connect to the primary system. Now users are connected back to the primary system. Trails created need to be shipped to the live standby system.

Start the Data Pump (P)

Start the data pump to begin sending data to the live-standby system. In the case of data pump process failure due to connection rejection by the target system, verify the hostname or IP address is reachable by performing PING from within GGSCI command-line and the manager listening port on the live-standby system is correct.

```
GGSCI (apps-srv1) 13> START EXTRACT ps1e203

Sending START request to MANAGER ...
EXTRACT PS1E203 starting

GGSCI (apps-srv1) 14>
```

Start the Delivery Process (S)

The delivery process (RS1E203) is started to keep applying transactions to the live standby system.

```
GGSCI (apps-srv2) 14> START REPLICAT rs1e203

Sending START request to MANAGER ...
REPLICAT RS1E203 starting

GGSCI (apps-srv2) 15>
```

On the primary system, verify the status of the primary and secondary extracts. This is where users are currently executing transactions. On the live standby system, verify the status of the delivery process. Using tools such Oracle GoldenGate Veridata to compare the primary and live-standby databases allows database administrators to discover and fix missing rows, proactively. Another effective administration tool is the Oracle GoldenGate 11g Monitor, which is used for monitor health checks and diagnostics.

Unplanned Primary to Standby Failover Workflow

Situations such as a primary system disk failure are unplanned situations where users are moved to the live standby system until the primary system is recovered. Figure 5-10 depicts the steps to move users from the primary to the live-standby system. The unplanned failover from the primary to live-standby system presumes the live-standby Oracle GoldenGate configuration is in-place, and the database scripts to enable constraints are available.

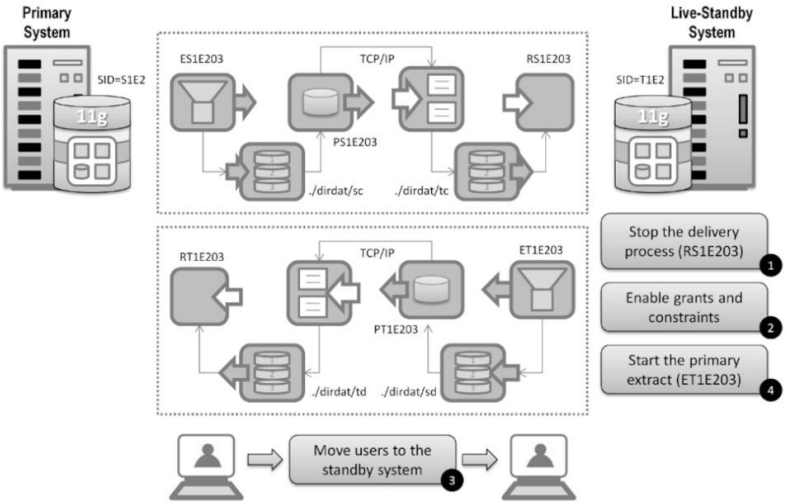


Figure 5-10: Primary to Standby Failover

Stop the Delivery Process (S)

Ensure transactions have been applied to the live standby system by continuing to issue the command LAG REPLICAT, until it returns "At EOF, no more records to process." This indicates all transactions have been applied to the live standby database.

```
GGSCI (apps-srv2) 1> LAG REPLICAT rs1e203

Sending GETLAG request to REPLICAT RS1E203 ...
Last record lag 8 seconds.
At EOF, no more records to process.

GGSCI (apps-srv2) 2> STOP REPLICAT rs1e203

Sending STOP request to REPLICAT RS1E203 ...
Request processed.

GGSCI (apps-srv2) 3>
```

Enable Grants and Constraints (S)

Since grants and constraints remain disabled during normal operation, before moving users, run the database DDL commands to grant database privileges and enable constraints.

Start the Primary Extract (S)

By starting the primary extract only, it continuously captures the transactions to the local trail, so that it is sent by the data pump when moving users back to the primary system.

```
GGSCI (apps-srv2) 4> START EXTRACT et1e203

Sending START request to MANAGER ...
EXTRACT ET1E203 starting

GGSCI (apps-srv2) 5>
```

Switch Users to Primary System

Redirect the application servers to the standby system. Since the primary system is not available until it is fixed, ensure the disk space is monitored to preserve the size of the local trail because the data pump is not available.

Unplanned Standby to Primary Failover Workflow

In the event of an unplanned failover, users are quickly moved to the standby system where business continues to operate. The downtime associated with an unplanned failover is minimal and mostly proportional to the unapplied amount of the remote trail files currently being applied to the live-standby system.

The next step is to determine the cause of the primary system failure and re-instantiate it from the live-standby system in preparation to move users back to the primary system. Figure 5-11 depicts the workflow for an unplanned live-standby to primary system failover.

The amount of downtime is greatly minimized by performing an initial load from the active live-standby system, then applying the transactions that occurred during the initial load timeframe to the primary system database. When Step 4 is completed, users are unable to execute transactions until Step 12 is completed (which enables the primary system database grants and constraints). Step 16 includes starting the application servers and moving users back to the primary system. The amount of downtime varies and depends on the size of the live-standby system local trail.

Task preformed on the primary system referred to by (P), whereas task performed on the live-standby system referred to by (S).

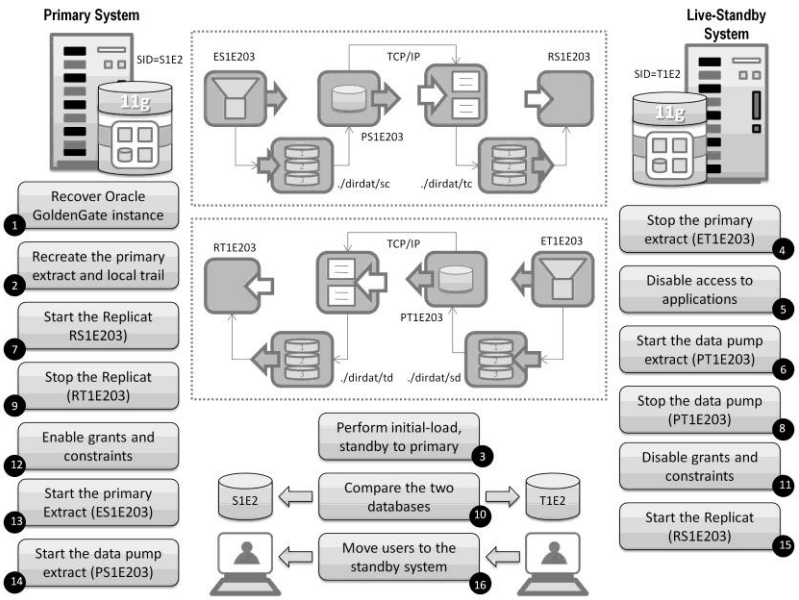


Figure 5-11: Unplanned Standby to Primary System Failover

Recover Oracle GoldenGate Instance (P)

Recovering Oracle GoldenGate instance may require installing the software, preferably to the same location to avoid modifying the instance parameter files. Then, restore the parameter files from the backup and start the manager process, using the same TCP/IP port to avoid modifying the manager and data pump parameter files. Verify the manager is running on the same port.

```
GGSCI (apps-srv1) 19> START mgr
Manager started.

GGSCI (apps-srv1) 20> INFO mgr

Manager is running (IP port apps-srv1.7805, Process ID 11252).

GGSCI (apps-srv1) 21>
```

Recreate the Primary Extract (P)

Regardless of the type of system failure, deleting and creating the primary extract is necessary. This avoids any type of reference to the old primary group.

```
GGSCI (apps-srv1) 13> DELETE EXTTRAIL ./dirdat/sc
Deleting extract trail ./dirdat/sc for extract ES1E203

GGSCI (apps-srv1) 14> DELETE EXTRACT es1e203
Deleted EXTRACT ES1E203.


GGSCI (apps-srv1) 15>
```

```
GGSCI (apps-srv1) 15> ADD EXTRACT es1e203, TRANLOG, BEGIN now, THREADS 1
EXTRACT added.

GGSCI (apps-srv1) 16> ADD EXTTRAIL ./dirdat/sc, EXTRACT es1e203, MEGABYTES 128
EXTTRAIL added.

GGSCI (apps-srv1) 17>
```

Perform Initial Load

There are several techniques to establish the primary database from the live-standby database system. Refer to Chapter 3 for details regarding the available techniques. Oracle GoldenGate direct load technique is illustrated below.

Before performing initial load to establish the primary system database using the direct load method, the database table structure must be prepared. Several techniques are available including the Oracle Database Data Pump Utility.

For direct initial load method, start by preparing the initial load delivery process – as it will be referenced by the initial load extract. The objective is to perform initial load from an active live-standby system database to minimize the downtime.

Create the initial load Replicat using the SPECIALRUN option.

```
GGSCI (apps-srv1) 21> EDIT PARAMS rload03
```

```
REPLICAT rload03
SPECIALRUN
ASSUMETARGETDEFS
HANDLECOLLISIONS
USERID ggs_admin, PASSWORD oracle
EXTFILE ./dirdat/initload03.dat
DISCARDFILE ./dirrpt/rload03.dsc, PURGE
MAP osm$repapi.*, TARGET osm$repapi.*;
END RUNTIME
```

```
GGSCI (apps-srv1) 22> ADD REPLICAT rload03, SPECIALRUN
REPLICAT added.

GGSCI (apps-srv1) 23>
```

Configure and create the initial load extract, ELOAD03. This extract reads from the active standby database and performs direct inserts using an initial load delivery process.

```
GGSCI (apps-srv2) 6> EDIT PARAMS eload03
```

```
EXTRACT eload03
USERID ggs_admin, PASSWORD oracle
RMTHOST apps-srv1, MGRPORT 7805
RMTTASK REPLICAT, GROUP rload03
TABLE osm$repapi.policy_types;
TABLE osm$repapi.discount types;
TABLE osm$repapi.policies;
TABLE osm$repapi.customers;
```

```
GGSCI (apps-srv2) 7> ADD EXTRACT eload03, SOURCEISTABLE
EXTRACT added.

GGSCI (apps-srv2) 8>
```

The primary system database is established by starting the initial load extract of the live standby system.

```
GGSCI (apps-srv2) 16> START EXTRACT eload03

Sending START request to MANAGER ...
EXTRACT ELOAD03 starting
```

```
GGSCI (apps-srv2) 17>
```

The following shows part of the report to verify the successful completion of the initial load.

```
GGSCI (apps-srv1) 5> VIEW REPORT rload03

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


Report at 2014-01-24 17:28:40 (activity since 2014-01-24 17:27:48)

From Table OSM$REPAPI.POLICY_TYPES to OSM$REPAPI.POLICY_TYPES:
       #                    inserts:         4
       #                    updates:         0
       #                    deletes:         0
       #                   discards:         0
From Table OSM$REPAPI.DISCOUNT_TYPES to OSM$REPAPI.DISCOUNT_TYPES:
       #                    inserts:         3
       #                    updates:         0
       #                    deletes:         0
       #                   discards:         0
From Table OSM$REPAPI.POLICIES to OSM$REPAPI.POLICIES:
       #                    inserts:    102100
       #                    updates:         0
       #                    deletes:         0
       #                   discards:         0
From Table OSM$REPAPI.CUSTOMERS to OSM$REPAPI.CUSTOMERS:
       #                    inserts:    102100
       #                    updates:         0
       #                    deletes:         0
       #                   discards:         0
```

Disable Application Access (S)

Stop the applications. This is "application server dependent."

Stop the Primary Extract (S)

Stop the primary extract in preparation to applying the changes that occurred during the initial load timeframe. Issue the LAG EXTRACT before stopping the extract, this ensures all transactions have been processed by the extract.

```
GGSCI (apps-srv2) 19> LAG EXTRACT et1e203

Sending GETLAG request to EXTRACT ET1E203 ...
Last record lag 2 seconds.
At EOF, no more records to process.
```

```
GGSCI (apps-srv2) 20> STOP EXTRACT et1e203

Sending STOP request to EXTRACT ET1E203 ...
Request processed.

GGSCI (apps-srv2) 21>
```

Start the Data Pump (S)

The data pump is started to send data to the primary system. These are the changes that occurred during the initial load.

```
GGSCI (apps-srv2) 21> START EXTRACT pt1e203

Sending START request to MANAGER ...
EXTRACT PT1E203 starting

GGSCI (apps-srv2) 22>
```

Start the Delivery Process (P)

Apply the changes to the primary database. Ensure handling collision to avoid a unique constraint error.

```
GGSCI (apps-srv1) 7> START REPLICAT rt1e203

Sending START request to MANAGER ...
REPLICAT RT1E203 starting

GGSCI (apps-srv1) 8>
```

Stop the Data Pump (S)

Stop the data pump on the standby system. Keep issuing the command LAG EXTRACT until it returns "At EOF, no more records to process." This indicates all transactions have been processed.

```
GGSCI (apps-srv2) 23> LAG EXTRACT pt1e203

Sending GETLAG request to EXTRACT PT1E203 ...
Last record lag 621,259 seconds.
At EOF, no more records to process.

GGSCI (apps-srv2) 24> STOP EXTRACT pt1e203

Sending STOP request to EXTRACT PT1E203 ...
Request processed.

GGSCI (apps-srv2) 25>
```

Stop the Delivery Process (P)

Stop the delivery process on the primary system. Keep issuing the command LAG REPLICAT until it returns "At EOF, no more records to process." This indicates all transactions have been applied to the primary database.

```
GGSCI (apps-srv1) 9> LAG REPLICAT rt1e203

Sending GETLAG request to REPLICAT RT1E203 ...
Last record lag 621,594 seconds.
At EOF, no more records to process.

GGSCI (apps-srv1) 10> STOP REPLICAT rt1e203

Sending STOP request to REPLICAT RT1E203 ...
Request processed.

GGSCI (apps-srv1) 11>
```

Compare Primary and Standby Databases

For data consistency, prior to proceeding, compare the primary and standby databases. Use tools such as Oracle GoldenGate Veridata to discover the missing rows or SQL*Plus to compare table rows count. For now, compare the two databases using rows count for tables CUSTOMERS and POLICIES using SQL*Plus for the tables.

```
SQL> conn osm$repapi/oracle
Connected.
SQL> select count(*) from customers;

  COUNT(*)
----------
    102100

SQL>
```

Execute the statement for the databases S1E2 and T1E2. Refer to Chapter 6 to learn more about Oracle GoldenGate Veridata.

Disable Grants and Constraints (S)

The receiving system must have grants and constraints disabled. This avoids referential integrity violations.

Enable Grants and Constraints (P)

Enable the database grants and constraints on the primary system.

Start the Primary Extract (P)

Start the primary extract. Since the primary was recreated, it will start transaction logs capture from the creation timestamp.

```
GGSCI (apps-srv1) 13> START EXTRACT es1e203

Sending START request to MANAGER ...
EXTRACT ES1E203 starting

GGSCI (apps-srv1) 14>
```

Start the Data Pump (P)

Start the data pump. This starts sending data to the standby system.

```
GGSCI (apps-srv1) 15> START EXTRACT ps1e203

Sending START request to MANAGER ...
EXTRACT PS1E203 starting

GGSCI (apps-srv1) 16>
```

Start the Delivery Process (S)

Start the delivery process to apply transactions sent to the standby system.

```
GGSCI (apps-srv2) 25> START REPLICAT rs1e203

Sending START request to MANAGER ...
REPLICAT RS1E203 starting

GGSCI (apps-srv2) 26>
```

Move Users to the Primary System

Now, confirm the Replicat has applied all the transactions to the standby system database, start the application servers to move users back to the primary system. This step is application server/site specific. For example, if it's PL/SQL portal application such as Oracle APEX, it's changing the database connect string only from the live-standby system to the primary system database.

High Availability

Oracle High Availability has been always associated with Oracle Real Application Clusters (RAC), where within the same data center, multiple tightly coupled servers (nodes) access the same database. Another advanced architecture of RAC is Extended RAC, where nodes are loosely coupled and spread across two or more data centers. Figure 5-12 compares the high availability of Oracle GoldenGate and Extended RAC.

The upper part of Figure 5-12 is Oracle GoldenGate configured using an active-active bi-directional topology model. Over the TCP/IP network, data flows from the site A (node-1) to site B (node-2) and vice versa. The self-evident advantage of this solution is the unlimited distance between node-1 and node-2. The drawback is the chance of scalability and data conflict handling. However, because the data is delivered in canonical format for a well-developed application, node-1 and node-2 may use different platforms. For example node-1 is Oracle Database 11g on Oracle Solaris while node-2 is Oracle Database 11g on Oracle Linux. Avoid heterogeneous database architecture for a high availability solution, unless the applications developed to support all database types transparently.

The lower part of Figure 5-12 shows Oracle Extended RAC. The databases are mirrored using dense wavelength division multiplexing (DWDM) over a dark fiber link. This enables always in-sync copies of the database. The self-evident advantage is out-of-box scalability and high availability. The drawback is the distance limitation between site A and site B for up to 100 kilometers and cost-factor.

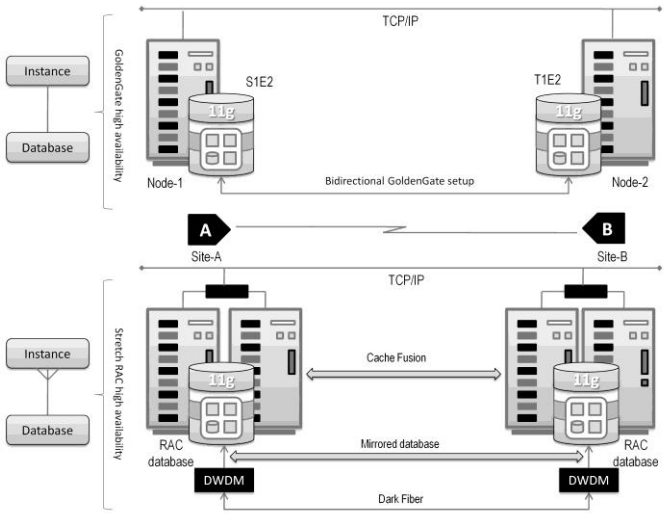


Figure 5-12: Comparison of 2-Node Oracle GoldenGate and Extended RAC

When using Oracle GoldenGate to implement a high availability platform, it is configured as an active-active bi-directional topology model. The source system (node-1) and target system (node-2) are available for user connections. Users may explicitly connect to either system based on predetermined attributes such as location, range of TCP/IP address, built-in application context, etc.

Implicit connections will be based on database and/or system workload. Users requests are routed to the system with minimal workload. This provides connection time workload evaluation to distribute database sessions across both systems. Figure 5-12 shows an implicit form of handling users connections. The database load balancer evaluates the preferred database, which is the database with lesser workload. Then, send the request to the application server which has its own HTTP load balancer. For more details regarding a high availability load balancer, refer to Figure 5-2.

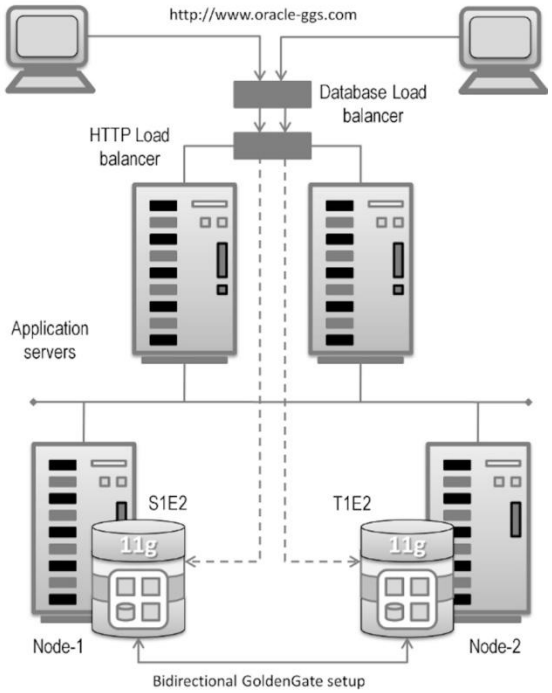


Figure 5-13: Load Balancing on Bi-Directional Topology

Applications running on Windows such as Oracle for .NET employ Oracle SQLNET to connect to the database. Whereas Java desktop applications employs JDBC drivers to connect to the database. For this type of applications, HTTP Servers are eliminated, the client applications establish stateful types of database connections.

Bi-Directional Replication Considerations

Since data manipulation language (DML) statements may occurs at 'almost' the same time from either direction, the below must be considered and configured on a bi-directional active-active Oracle GoldenGate configuration.

Endless Loop Detection

In an active-active bi-directional topology, because the Extract and the Replicat are operating on the same table, the Replicat must be prevented from sending the same applied record back to the source to avoid an endless loop situation.

There are two techniques for loop detection to avoid re-extracting and re-replicating the same record:

- Replicat-Extract trace table
- Replicat-Extract parameter

Replicat-Extract Trace Table

When the Replicat applies transactions to the target database, it writes to a trace table which the Extract on the same system uses for loop detection by eliminating the Replicat transactions. When the Replicat starts, if the parameter TRACETABLE is configured then it uses the specified trace table. If the parameter TRACETABLE is not configured, it tries the default trace table name GGS_TABLE owned by the Oracle GoldenGate user of the DBLOGIN parameter. The following commands illustrate a non-default trace table GGS_T1E2_TRC is created and configured as follows:

- Create the trace table GGS_T1E2_TRC. This requires connecting to the database from within Oracle GoldenGate and issues the two GGSCI commands below:

```
GGSCI (apps-srv2) 1> DBLOGIN USERID ggs_admin, PASSWORD oracle
Successfully logged into database.

GGSCI (apps-srv2) 2> ADD TRACETABLE ggs_admin.ggs_t1e2_trc

Successfully created trace table GGS_ADMIN.GGS_T1E2_TRC.

GGSCI (apps-srv2) 3>
```

- Add the parameter TRACETABLE to the Extract and the Replicat parameter file. The GGS_ADMIN is the Oracle GoldenGate database schema.

```
TRACETABLE ggs_admin.GGS_T1E2_TRC
```

Replicat Extract Parameter

The second method uses the Replicat TRANLOGOPTIONS parameter only. This methods eliminates the processing overhead associated with using the database trace table explained above. The option EXCLUDEUSER or EXCLUDEUSERID of the parameter TRANLOGOPTIONS explicitly avoid the re-capture of the transactions applied by specified user. This requires using a dedicated Oracle GoldenGate user so that transactions applied by this user are eliminated – avoiding re-sending the transaction back to the originated system. The TRANLOGOPTIONS is shown next.

```
TRANLOGOPTIONS EXCLUDEUSER ggs_admin
```

Query DBA_USERS to replace the username with the user ID.

```
SQL> SELECT username, user_id
  2  FROM   dba_users
  3  WHERE  username='GGS_ADMIN';

USERNAME                          USER_ID
------------------------------ ----------
GGS_ADMIN                            1101

SQL>
```

Data Conflicts

This issue is related to data integrity that must be addressed comprehensively in an active-active bi-directional configuration. There are several types of conflict, avoidance, and detection techniques. Because applications are mostly not replication-ware, resolving data conflict is required and should be governed by the business rules. Oracle GoldenGate best practices recommend resolving data conflicts in the following order:

- Apply avoidance techniques of data segmentation and application partitioning
- Adapt Oracle GoldenGate automatic conflict detection and resolution (CDR)
- Develop custom conflict detection and resolution routines

Details regarding data conflicts and detection techniques are described in this chapter.

Truncate Table

Since the TRUNCATE TABLE command is not supported in an active-active bi-directional configuration, at any point in time, Extracts and Replicats must be configured for one-and-only-one dedicated system to issue the TRUNCATE TABLE database command. This is achieved by adding the parameter GETTRUNCATES

on the allowed system and the parameter IGNORETRUNCATES on the disallowed system. To ensure the TRUNCATE TABLE command is issued from the allowed database only, revoke the TRUNCATE TABLE privilege from the application roles.

Database Sequences

When using database sequence numbers for tables primary and unique keys values on an active-active bi-directional configuration, avoid primary and unique key uniqueness conflicts. There are several techniques for implementing database sequences on an active-active bi-directional configuration. Common techniques for using database sequences are:

- Node-1 database takes odd numbers, while node-2 database takes even numbers
- Node-1 database uses a different range of numbers than node-2 database
- Implement database sequence values based on built-in application contexts

Details regarding database sequence implementations are described in this chapter.

Real-Time Disconnected Transactions

When the applications retrieve and manipulate data synchronously from either of the two databases, database links are created which enable the application to fetch and to manipulate data from either databases. There is no data conflict. This is known as distributed database applications and adhere to database dependency, locks, and resource management as illustrated below.

Create two database links from S1E2 to T1E2 and from T1E2 to S1E2. Before creating the database link, ensure the database connect string is able to establish remote connection.

Create the database link from S1E2 to T1E2.

```
SQL> CREATE PUBLIC DATABASE LINK t1e2.precisetrace.com
  2  CONNECT TO osm$repapi IDENTIFIED BY oracle
  3  USING 'T1E2';

Database link created.

SQL>
```

Create the database link from T1E2 to S1E2.

```
SQL> CREATE PUBLIC DATABASE LINK s1e2.precisetrace.com
  2  CONNECT TO osm$repapi IDENTIFIED BY oracle
  3  USING 'S1E2';

Database link created.
```

```
SQL>
```

The following update statement executed remotely on S1E2 over the database link. Do not issue COMMIT statement to continue locking resources.

```
SQL> host hostname
apps-srv2

SQL> CONN osm$repapi/oracle
Connected.
SQL> UPDATE customers@s1e2 SET
  2     cust_email='falswaimil@gmai.com'
  3  WHERE cust_no=100100;

1 row updated.

SQL>
```

The following update statement waits for the above remote transaction to release resources, issuing COMMIT or ROLLBACK, then execute the update.

```
SQL> host hostname
apps-srv1

SQL> CONN osm$repapi/oracle
Connected.
SQL> UPDATE customers SET
  2     cust_email='falswaimil@yahoo.com'
  3  WHERE cust_no=100100;

SQL>
```

This illustrates how synchronous transactions used by distributed applications do not trigger data conflict. They adhere to the target database resource management, locks, and queue handling.

However, for near real-time 'disconnected' transactions that are running asynchronously, the need to consider and handle data conflict caused by executing the applications is very obvious. Conflicts are either avoided using data and applications segmentation or handled using Oracle GoldenGate built-in and customs conflict detection and resolution. The section 'conflicts resolution' describes the built-in and advance custom techniques for handling data conflict.

Applications and Database Design for High Availability

The key to avoid data conflict in an asynchronous environment is to partition data and application dynamically, which is referred to as segmentation. This guarantees an application's group interacts and manipulates a logically partitioned subset of

the database that is isolated from others application's groups. This is the fundamental concept of virtual private database (VPD). Figure 5-14 shows two application's groups in an active-active bi-directional topology model that are designed to be virtually isolated.

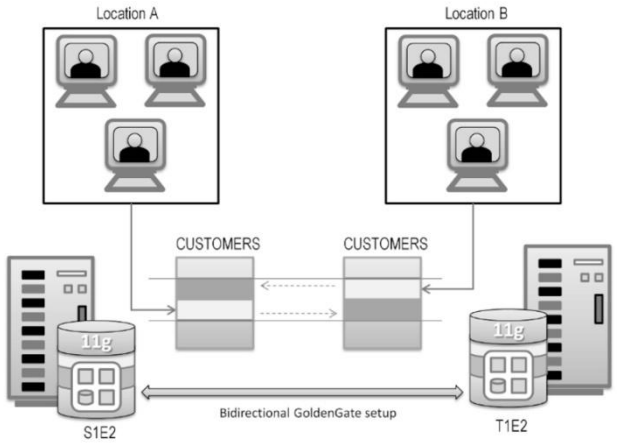


Figure 5-14: Conflict Avoidance Using Data and Applications Segmentation

The above data conflict avoidance technique is either implemented at the applications level or the database. Best practices recommend using the Oracle Database Enterprise Edition Virtual Private Database (VPD) feature. VPD uses policy functions for dynamically modifying SQL Statements to enforce data segmentation through data access control based on application contexts. Virtual Private Database implementation is explained in more detail in the section 'Virtual Private Database.'

An valuable feature for implementing segmentation by location is performance. Users are located in close proximity to access data. For example, a call center with two locations, where each location handles sales for specific products list avoids data conflict and has local data access which eliminates the overhead associated with network failure. Users continue to enter transactions locally. When the network becomes available, Oracle GoldenGate starts routing data to a remote system.

Bi-Directional Extract and Delivery Groups

The following Oracle GoldenGate configuration is built on the components used by the disaster recovery described above. Since Oracle GoldenGate high availability is deployed as an active-active bi-directional topology architecture, it is recommended to divide it into two phases:

Phase I: Working Bi-Directional Topology Architecture – An end-to-end optimized and well-tested bi-directional replication capable of handling the transactions volume for near real-time data movement. Data conflict must be addressed as it's an unavoidable.

Phase II: Comprehensive Conflicts Handling – Implementing avoidance and detection techniques for handling data conflicts. Since Oracle GoldenGate operates asynchronously, near real-time data movement minimizes conflict occurrences.

Figure 5-15 outlines the groups and trails. The upper part is for node-1 to node-2 data flow and the lower part is the opposite configuration, from node-2 to node-1.

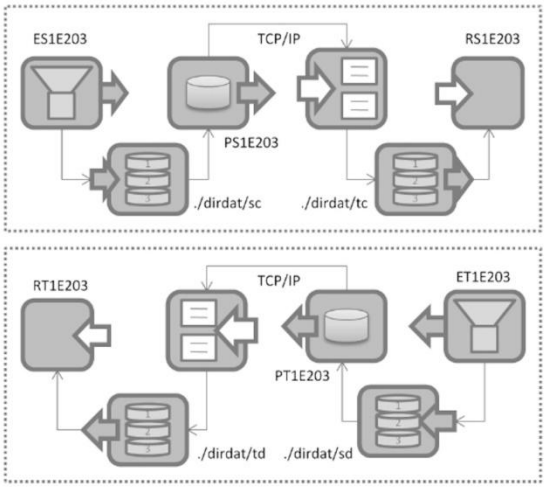


Figure 5-15: Bi-Directional High Availability Topology

Node-1 Parameter Files, Extracts, and Replicats Processes

Node-1 Oracle GoldenGate instance operates the processes listed in Table 5-2. Since an end-to-end data integrity among referenced tables must be enforced, designing multiple capture-and-delivery groups requires understanding the data model, our illustration is based on a single process architecture.

Component	Description
ES1E203	Primary capture process from the primary system transactions logs
./dirdat/sc	Local trail
PS1E203	Secondary extract data pump process send data to node-2 system
./dirdat/tc	Remote trail
RT1E203	Replicat process for node-2 database

Table 5-2: Node-1 Components

The parameter files must be modified to support bi-directional replication for conflict handling. Re-start the processes to activate the newly added parameters.

Node-1 primary extract process ES1E203 is shown below. Similarly modify Node-2 primary extract.

```
EXTRACT es1e203
SETENV  (ORACLE SID=S1E2)
SETENV  (ORACLE_HOME="/d01/app/oracleSE")
SETENV  (NLS LANG=AMERICAN AMERICA.AR8ISO8859P6)
USERID  ggs_admin, password oracle
EXTTRAIL ./dirdat/sc
TABLE osm$repapi.CUSTOMERS;
TABLE osm$repapi.POLICIES
```

Node-2 delivery process RS1E203 is shown below. Similarly modify node-1 delivery process.

```
REPLICAT rt1e203
SETENV (ORACLE_SID=S1E2)
SETENV (NSL_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rt1e203.dsc, PURGE
MAP osm$repapi.customers, TARGET osm$repapi.customers;
MAP osm$repapi.policies, TARGET osm$repapi.policies;
```

Verify the status of the processes on node-1 Oracle GoldenGate instance. The status of RUNNING indicates the Extract, the data pump, and the Replicat processes are configured correctly, at least.

The current status of the primary extract, ES1E203.

```
GGSCI (apps-srv1) 4> INFO EXTRACT es1e203

EXTRACT    ES1E203   Last Started 2014-01-28 22:23   Status RUNNING
Checkpoint Lag       03:42:54 (updated 00:00:05 ago)
Process ID           6378
Log Read Checkpoint  Oracle Redo Logs
                     2014-01-28 18:41:00  Thread 1, Seqno 30, RBA 39536
                     SCN 0.1997389 (1997389)
```

The local trail details, ./dirdat/sc.

```
GGSCI (apps-srv1) 5> INFO EXTTRAIL ./dirdat/sc

       Extract Trail: ./dirdat/sc
             Extract: ES1E203
               Seqno: 10
                 RBA: 1601
           File Size: 128M
```

The current status of the secondary (data pump) extract, PES1E203.

```
GGSCI (apps-srv1) 8> INFO EXTRACT ps1e203

EXTRACT    PS1E203   Last Started 2014-01-28 22:25   Status RUNNING
Checkpoint Lag       100:45:36 (updated 00:00:01 ago)
Process ID           6410
Log Read Checkpoint  File ./dirdat/sc000009
                     2014-01-24 17:40:22.000000  RBA 44546492
```

The current status delivery process from standby to primary system, RT1E203.

```
GGSCI (apps-srv1) 10> INFO REPLICAT rt1e203

REPLICAT   RT1E203   Last Started 2014-01-28 22:27   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:08 ago)
Process ID           6428
Log Read Checkpoint  File ./dirdat/td000008
                     First Record  RBA 45282

GGSCI (apps-srv1) 11>
```

Node-2 Parameter Files, Extracts, and Replicats Processes

Node-2 Oracle GoldenGate instance operates the processes listed in Table 5-3. When Oracle GoldenGate is configured for high availability, node-1 and node-2 systems should have similar resources in terms of CPU speed and RAM size. This allows similar workload handling across the two nodes.

Component	Description
ET1E203	Primary Extract process from node-2 transactions logs
./dirdat/sd	Local trail
PT1E203	Secondary extract data pump process from node-2 to node-1
./dirdat/td	Remote trail
RS1E203	Replicat process on node-1

Table 5-3: Node-2 Components

Verify the status of the processes on node-2 Oracle GoldenGate instance.

```
GGSCI (apps-srv2) 4> INFO EXTRACT et1e203

EXTRACT    ET1E203   Last Started 2014-01-28 22:31   Status RUNNING
Checkpoint Lag       100:58:00 (updated 00:00:07 ago)
Process ID           17047
Log Read Checkpoint  Oracle Redo Logs
                     2014-01-24 17:33:26  Thread 1, Seqno 25, RBA 42078224
                     SCN 0.1889870 (1889870)

GGSCI (apps-srv2) 6> INFO EXTRACT pt1e203

EXTRACT    PT1E203   Last Started 2014-01-28 22:32   Status RUNNING
Checkpoint Lag       273:31:53 (updated 00:00:04 ago)
Process ID           17083
```

```
Log Read Checkpoint  File ./dirdat/sd000005
                     2014-01-17 13:00:31.000000  RBA 45235

GGSCI (apps-srv2) 8> INFO REPLICAT rs1e203

REPLICAT   RS1E203   Last Started 2014-01-28 22:32   Status RUNNING
Checkpoint Lag       00:00:29 (updated 00:00:00 ago)
Process ID           17097
Log Read Checkpoint  File ./dirdat/tc000016
                     2014-01-28 22:32:32.880946  RBA 1686454

GGSCI (apps-srv2) 9>
```

Oracle Sequence for Artificial Primary Key Values

Oracle database sequence is a memory-based object. Hence, it is an efficient technique for generating primary key values. For a non-replicated database, sequences are suitable for generating primary key values without major issues. However, when it is used in a database synchronized by Oracle GoldenGate, it should be created to avoid primary key conflict for SQL insert statements. The following illustrates techniques for using Oracle sequences in databases synchronized in real-time by Oracle GoldenGate.

Odd-Even Numbers Distribution

The node-1 database fetches the odd number values only, while node-2 database fetches the even number values. Figure 5-16 depicts the sequence design created below:

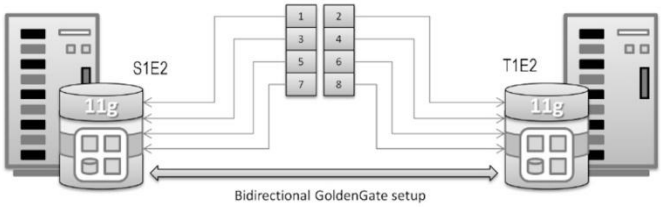


Figure 5-16: Odd-Even Numbers Sequencing

Node-1 database sequence is created to start with 1 and fetches odd numbers.

```
SQL> CREATE SEQUENCE cust_no_seq
  2  START WITH 1
  3  INCREMENT BY 2;

Sequence created.

SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
         1
```

```
SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
         3

SQL>
```

Node-2 database sequence is created to start with 2 and fetches even numbers.

```
SQL> CREATE SEQUENCE cust_no_seq
  2  START WITH 2
  3  INCREMENT BY 2;

Sequence created.

SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
         2

SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
         4

SQL>
```

Range Numbers Distribution

Node-1 database fetches from one range while node-2 database fetches from a second range. Figure 5-17 depicts the sequence design created below:

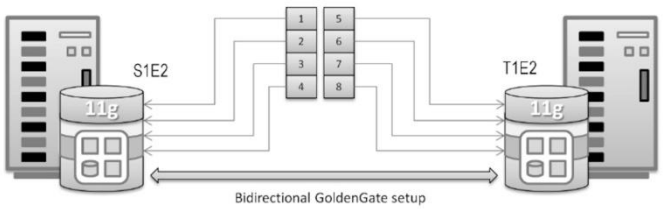


Figure 5-17: Range Numbers Sequencing

Node-1 database begins from 1 to n-1, where n is node-2 and begins sequence number.

```
SQL> CREATE SEQUENCE cust_no_seq
  2  START WITH 1;

Sequence created.
```

Node-2 begins at n, where n=5.

```
SQL> CREATE SEQUENCE cust_no_seq
  2  START WITH 5;

Sequence created.

SQL>
```

Application Context of Numbers Distribution

The following illustrates the use of sequence to generate unique values based on the application context. Figure 5-18 depicts the application context value generator for a two-node architecture.

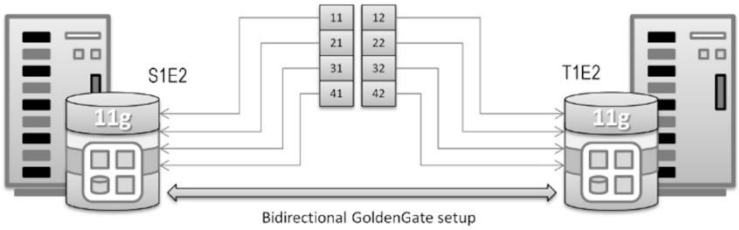


Figure 5-18: Numbers Sequencing based on Application Context

```
SQL> CREATE SEQUENCE cust_no_seq
  2  START WITH 1;

Sequence created.

SQL>
```

Use the database function SYS_CONTEXT to concatenate the sequence number with the instance number (INSTANCE) for the session. This technique is similar to a sequence created with INCREMENT BY 10.

```
SQL> CREATE SEQUENCE cust_no_seq;

Sequence created.

SQL> SELECT TO_NUMBER(cust_no_seq.nextval||SYS_CONTEXT('USERENV', 'INSTANCE'))
  2  FROM   dual;

TO_NUMBER(CUST_NO_SEQ.NEXTVAL||SYS_CONTEXT('USERENV','INSTANCE'))
------------------------------------------------------------------
                                                                11
SQL>
```

Instead of using the generated number by the sequence alone for the primary key values, when combined with SYS_CONTEXT('USERENV', 'DB_NAME), it generates a unique alphanumeric value of a character data type primary key.

Multi-Master Numbers Distribution

When Oracle GoldenGate is configured to support 3 systems for multi-master data replication, the first system sequence number begins at 1, the second system sequence begins at 2, and the third system sequence number begins at 3. All sequences increment by 3 as shown in Figure 5-19.

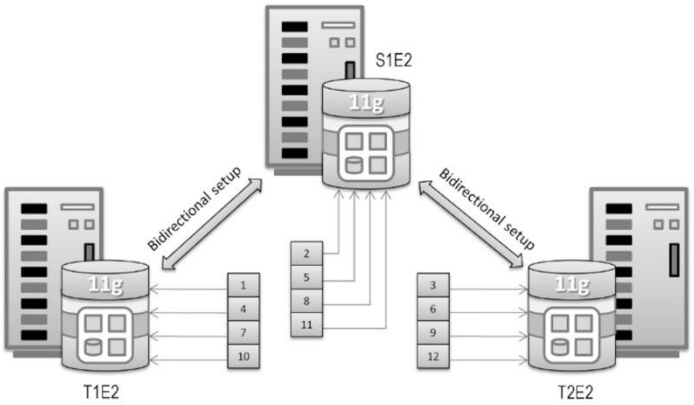


Figure 5-19: Multi-Master Numbers Sequencing

On node-1 (T1E2), create the sequence to begin at 1.

```
SQL> CREATE SEQUENCE cust_no_seq
  2  START WITH 1
  3  INCREMENT BY 3;

Sequence created.

SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
         1

SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
         4

SQL>
```

On node-2 (S1E2), create the sequence to begin at 2.

```
SQL> CREATE SEQUENCE cust_no_seq
  2  START WITH 2
  3  INCREMENT BY 3;

Sequence created.

SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
         2

SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
         5

SQL>
```

On node-3 (T2E2), create the sequence to begin at 3.

```
SQL> CREATE SEQUENCE cust_no_seq
  2  START WITH 3
  3  INCREMENT BY 3;

Sequence created.

SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
         3

SQL> SELECT cust_no_seq.nextval FROM dual;

   NEXTVAL
----------
         6

SQL>
```

Database sequences created to avoid SQL insert conflicts are not to be included on the TABLE and MAP statements. To avoid encountering gaps due to database restart, use the NOCACHE attribute when creating sequences. Since the sequence START WITH and INCREMENT BY attributes are different on both systems, ascertain that the Oracle GoldenGate DDL replication is disabled when creating sequences and manually create them on both systems.

Data Conflicts

Without explicit conflict detection, applications will continue processing transactions of inconsistent data. Logical data conflict and inconsistencies should be handled using either Oracle GoldenGate, a built-in, or custom developed conflict detection and resolution techniques. In addition, virtual private database implementation minimizes data conflict as explained in this chapter. The demonstrated conflict resolution techniques are developed for an online application system for a two-node configuration architecture, synchronized using Oracle GoldenGate. The primary database (node-1) is located at Boston, MA and the target database (node-2) is located at Albany, NY as shown in Figure 5-20.

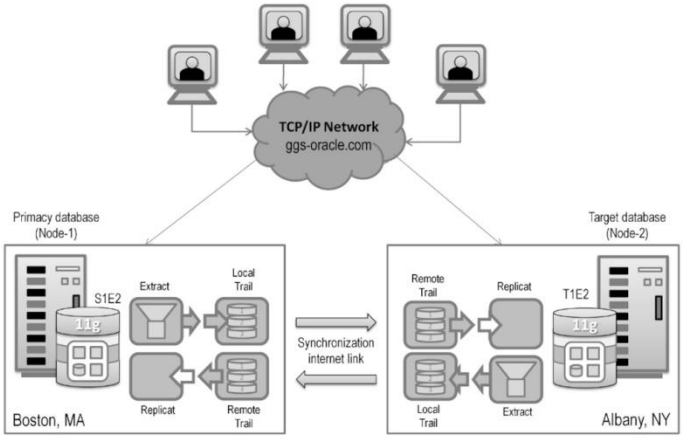


Figure 5-20: Bi-Directional Replication for Online Applications

Let us start by explaining the usages and applications of SQLEXEC parameter. This parameter is used in conjunction with custom data conflict handling technique developed in PL/SQL.

SQLEXEC Parameter

The use of the SQLEXC parameter extends Oracle GoldenGate capabilities, it enables invoking PL/SQL procedures and executes queries from within Oracle GoldenGate Extract or Replicat to support complex data conflict resolutions. The vast majority of data conflicts are handled using queries only, while procedures are mostly applied to more complex data conflicts and transformations.

Data manipulation and data conflict resolution are developed by utilizing Oracle GoldenGate built-in functions. The use of the functions @IF, @STREQ, @COMPUTE, @GETVAL and @GETENV is common. The following illustration

of using SQLEXEC is developed to support a de-normalized POLICIES table which includes a PT_DESC column.

```
SQL> DESC policies
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 POL_NO                                    NOT NULL NUMBER
 POL_FROM                                  NOT NULL DATE
 POL_TO                                    NOT NULL DATE
 POL_VALUE                                 NOT NULL NUMBER(8,2)
 POL_SUB_TOTAL                             NOT NULL NUMBER(8,2)
 PT_CODE                                   NOT NULL VARCHAR2(8)
 DT_CODE                                            VARCHAR2(8)
 POL_TOTAL                                 NOT NULL NUMBER(8,2)
 CUST_NO                                   NOT NULL NUMBER
 PT_DESC                                            VARCHAR2(128)

SQL>
```

The PT_DESC (Policy Type Description) column exists on the target database only. Hence, a lookup operation within the Replicat is necessary to complete the transaction successfully.

SQLEXEC with Queries

The components of SQLEXEC with a query specifies the query ID, which is used by the built-in function @GETVAL to assign the value to the target column. The PARAMS clause lists the pair of parameter name and value used in the query. The last clause is the COLMAP to perform column mapping which references the query columns list using the query ID.

The Replicat MAP statement uses SQLEXEC to fetch the value for the column PT_DESC using PT_CODE as a parameter.

```
MAP osm$repapi.policies, TARGET osm$repapi.policies, &
    SQLEXEC (ID lookup_pt, &
             QUERY " SELECT pt_desc
                     FROM   osm$repapi.policy_types
                     WHERE  pt_code = :pt_code_param ", &
             PARAMS (pt_code_param = pt_code)), &
             COLMAP (USEDEFAULTS, pt_desc = @GETVAL(lookup_pt.pt_desc));
```

The Replicat (RS1E203) parameter file using SQLEXEC with query clause.

```
REPLICAT rs1e203
SETENV (ORACLE_SID=T1E2)
SETENV (NSL_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rs1e203.dsc, PURGE
MAP osm$repapi.customers, TARGET osm$repapi.customers;
```

```
MAP osm$repapi.policies, TARGET osm$repapi.policies, &
    SQLEXEC (ID lookup_pt, &
             QUERY " SELECT pt desc
                     FROM   osm$repapi.policy_types
                     WHERE  pt_code = :pt_code_param ", &
             PARAMS (pt_code_param = pt_code)), &
             COLMAP (USEDEFAULTS, pt_desc = @GETVAL(lookup_pt.pt_desc));
```

SQLEXEC with Stored Procedures

SQLEXEC invokes stored procedure to perform advanced data transformations and conflicts resolutions. The simplest form of using SQLEXEC with stored procedure is performing a table look operation. The below PL/SQL database procedure performs a table look up operation instead of using query.

```
SQL> CREATE OR REPLACE PROCEDURE lookup_pt(pt_code_param IN VARCHAR2,
  2                                        pt_desc_param OUT VARCHAR2) IS
  3  BEGIN
  4     SELECT pt_desc into pt_desc_param
  5     FROM   osm$repapi.policy_types
  6     WHERE  pt_code = pt_code_param;
  7  END;
  8  /

Procedure created.

SQL> GRANT EXECUTE on lookup_pt to ggs_admin;

Grant succeeded.

SQL>
```

The Replicat parameter file is modified to reference the stored procedure using the SQLEXEC parameter.

```
REPLICAT rs1e203
SETENV (ORACLE_SID=T1E2)
```

```
SETENV (NSL LANG=AMERICAN AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rs1e203.dsc, PURGE
MAP osm$repapi.customers, TARGET osm$repapi.customers;
MAP osm$repapi.policies, TARGET osm$repapi.policies, &
    SQLEXEC (SPNAME osm$repapi.lookup pt, &
             ID     lookup_pt, &
             PARAMS (pt code param = pt code)), &
             COLMAP (USEDEFAULTS, &
                     pt_desc = @GETVAL(lookup_pt.pt_desc_param));
```

The above SQLEXEC invokes a stored procedure by specifying the following clauses:

SPNAME is the name of the stored procedure. Ascertain granting Oracle GoldenGate database user to EXECUTE privilege on the called procedure.

ID is the logical name for the procedure and is needed by the @GETVAL function to retrieve the look up value.

PARAMS is the list of input parameters and their initial values.

COLMAP performs column mapping which references the output parameters using the logical name.

SQLEXEC as Stand-Alone Parameter

The third form of SQLEXEC is a stand-alone. This is a one-time call-out to execute query, DDL command, or stored procedure. This type of SQLEXEC is valid for Extract and Replicat parameter files.

The next parameter files ES1E203.PRM use SQLEXEC parameters. The first parameter executes a query to avoid IDLE_TIME enforced by the user profile.
The second parameter executes the DDL command TRUNCATE. The third parameter executes a stored procedure. Notice the procedure must not include input or output formal parameters.

```
EXTRACT es1e203
SETENV (ORACLE_SID=S1E2)
SETENV (ORACLE_HOME=/d01/app/oracleSE/product/11.2.0/dbhome_1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
EXTTRAIL ./dirdat/sc
TRANLOGOPTIONS EXCLUDEUSER ggs_admin
SQLEXEC "SELECT * FROM DUAL" EVERY 55 MINUTES
SQLEXEC "TRUNCATE TABLE osm$repapi.logging" EVERY 1 DAYS
SQLEXEC "CALL send_notify()" EVERY 10 MINUTES
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
```

Custom Data Conflicts Handling

Consider an online application for managing a single inventory site. The inventory database is replicated across two cities – Boston, MA and Albany, NY as depicted in Figure 5-20. Customers' requests are directed to either database based on workload determined by the load balancer. Hence, data conflict is unavoidable. A classical data conflict may occur due to updates occurring at the time. When a conflict occurs, several resolutions are considered and should be governed by business rules.

- Use the net difference instead of using the after image value
- Perform comparison on other columns to accept, reject, or resolve transaction modifications
- Use the timestamp (date and time) then decide to accept or to reject the transaction modifications
- Manually handle the conflict by mapping data to an exception table

Conflicts Resolution Using SQLEXEC

When simultaneous bi-directional transactions update the same non-key row value on node-1 and on node-2, data divergence occurs slightly. This is a form of data conflict and must be detected and resolved.

Data Divergence Scenario

The customer has sent a change of email address request (non-key column). The request is received and routed to an agent at Boston. Shortly thereafter, the same customer sends another change of email address request with a different email from the previous request. The request is received and routed to an agent at Albany. Here is an illustration of processing this request without data conflict handling in place.

Node-1 Database: Boston	Node-2 Database: Albany
SQL> SELECT cust_email 2 FROM customers 3 WHERE cust no=100200; CUST EMAIL --------------- fawzi@yahoo.com SQL>	SQL> SELECT cust_email 2 FROM customers 3 WHERE cust no=100200; CUST EMAIL --------------- fawzi@yahoo.com SQL>
Data pump PS1E203 is down due to network failure	Data pump PT1E203 is down due to network failure
SQL> UPDATE customers SET 2 cust email='fawzi@gmail.com' 3 WHERE cust_no=100200; 1 row updated. SQL> COMMIT; Commit complete. SQL>	SQL> UPDATE customers SET 2 cust email='fawzi@mail.com' 3 WHERE cust_no=100200; 1 row updated. SQL> COMMIT; Commit complete. SQL>
SQL> SELECT cust email 2 FROM customers 3 WHERE cust_no=100200; CUST_EMAIL	SQL> SELECT cust email 2 FROM customers 3 WHERE cust_no=100200; CUST_EMAIL

---------- fawzi@gmail.com SQL>	---------- fawzi@mail.com SQL>
Network restored. The data pump PS1E203 started automatically.	Network restored. The data pump PT1E203 started automatically.
SQL> SELECT cust email 2 FROM customers 3 WHERE cust_no=100200; CUST EMAIL ---------- fawzi@mail.com SQL>	SQL> SELECT cust email 2 FROM customers 3 WHERE cust_no=100200; CUST EMAIL ---------- fawzi@gmail.com SQL>

There is no error, but notice that the change of address resulted in mismatched CUST_EMAIL values between the two databases. This is a simple data conflict, yet critical to be addressed to avoid a data integrity problem.

Data Divergence Handling

Oracle GoldenGate captures before and after image of updates to support custom and built-in conflict resolutions by performing a compare operation before applying the transactions on the target database. Data divergence conflicts are prevented. Figure 5-21 shows the workflow of updating the email for CUST_NO=100200. Since there is no data segmentation at the application level, a custom data conflict is developed.

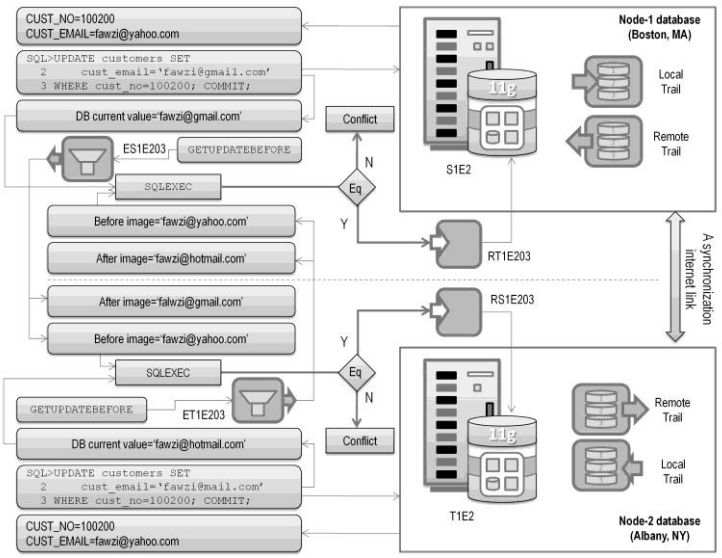


Figure 5-21: Resolving Conflict Using SQLEXEC

The Replicat applies the transaction to the database and the conflict occurs silently. The conflict is resolved either using the built-in conflict detection and resolution (CDR) or customer resolution using the SQLEXEC parameter. Using either technique, the database current value is compared with before image value from the trail record.
If the two values are equal, then the transaction is applied. Otherwise, a data conflict is detected and needs to be handled accordingly.

Using the parameter SQLEXEC to handle the above data conflict requires using a query or a procedure to fetch the current database value and compare with the before image from the trail record. If the two values are identical, then apply the transaction.
Otherwise report or handle data conflict. The stored procedure GET_EMAIL for handling this type of update conflict follows.

```
SQL> CREATE OR REPLACE PROCEDURE get_email(pcust_no    IN NUMBER,
  2                                        pcust_email OUT VARCHAR2) IS
  3  BEGIN
  4     SELECT cust_email into pcust_email
  5     FROM   osm$repapi.customers
  6     WHERE  cust_no = pcust_no;
  7  END;
  8  /

Procedure created.

SQL>
```

The Extract ES1E203 uses the parameter GETUPDATEBEFORES to force capture the transaction post values and before image values for use by the Replicat to detect conflict. Also, to support operation granularity, the parameter GETBEFORECOLS is used in conjunction with GETUPDATEBEFORES to associate the before image with the operation type of UPDATE, DELETE, or both. The argument ALL specifies the before image value is for all columns. The example below illustrates using GETBEFORECOLS to capture the before image values for update and delete operations on the CUST_EMAIL column only. The argument KEYINCLUDING includes capture of the primary key before image value.

```
TABLE osm$repapi.customers &
      GETBEFORECOLS(ON UPDATE KEYINCLUDING (cust_email), &
                    ON DELETE KEYINCLUDING (cust_email));
```

The parameter GETUPDATEBEFORES remains in effect for all subsequent listed tables until the parameter IGNOREUPDATEBEFORES is encountered – enabling control of the before image capture for a specific list of tables.

```
EXTRACT es1e203
SETENV (ORACLE_SID=S1E2)
SETENV (ORACLE_HOME=/d01/app/oracleSE/product/11.2.0/dbhome_1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
```

```
USERID ggs_admin, PASSWORD oracle
EXTTRAIL ./dirdat/sc
TRANLOGOPTIONS EXCLUDEUSER ggs admin
GETUPDATEBEFORES
TABLE osm$repapi.customers GETBEFORECOLS(ON UPDATE ALL);
IGNOREUPDATEBEFORES
TABLE osm$repapi.policies;
```

The Replicat RS1E203 parameter file includes the SOURCEDEFS parameter for mapping non-identical tables structure. Using SQLEXEC to invoke the procedure GET_EMAIL returns the current database value for compression with the before image value. The COLMAP clause includes reference to the Oracle GoldenGate built-in function to detect conflict. If the built-in function @STREQ returns 1 (true) then no conflict was detected. Otherwise, a conflict has been detected and the column CUST_EMAIL assigned the value 'conflict@pt.com' as a means of notification and intervention.

```
cust_email     = @IF(@STREQ(BEFORE.cust_email,
                     @GETVAL(get_email.pcust_email)),
                     cust_email, 'conflict@pt.com'),
```

```
REPLICAT rs1e203
SETENV (ORACLE_SID=T1E2)
SETENV (NSL_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs admin, PASSWORD oracle
SOURCEDEFS /d01/oracle/ggs/dirdef/defgen.def
--ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rs1e203.dsc, PURGE
GETUPDATES
GETINSERTS
GETDELETES
MAP osm$repapi.customers, TARGET osm$repapi.customers, &
    SQLEXEC (ID get email, &
        SPNAME osm$repapi.get_email, &
        PARAMS (pcust_no = cust_no)), &
        COLMAP (cust_no        = cust_no,
                cust name      = cust name,
                cust_email     = @IF(@STREQ(BEFORE.cust_email,
                                     @GETVAL(get email.pcust email)),
                                     cust_email, 'conflict@pt.com'),
                cust_mobile    = cust_mobile,
                cust address   = cust address,
                cust_commit_ts = @GETENV('GGHEADER', 'COMMITTIMESTAMP'));
--
MAP osm$repapi.policies, TARGET osm$repapi.policies, &
    SQLEXEC (SPNAME osm$repapi.lookup_pt, &
        ID     lookup_pt, &
        PARAMS (pt code param = pt code)), &
        COLMAP (USEDEFAULTS, &
                pt_desc = @GETVAL(lookup_pt.pt_desc_param));
```

The following scenario illustrates conflict detection for an update operation. The Replicat applying the transaction detects the conflict and assigns the

CUST_EMAIL column with temporary email 'conflict@pt.com' for further action.

Node-1 Database: Boston	Node-2 Database: Albany
SQL> SELECT cust email 2 FROM customers 3 WHERE cust_no=4846; CUST EMAIL -------------------- deema@yahoo.com SQL>	SQL> SELECT cust email 2 FROM customers 3 WHERE cust_no=4846; CUST EMAIL -------------------- deema@yahoo.com SQL>
Data pump PS1E203 is down due to network failure	Data pump PT1E203 is down due to network failure
SQL> UPDATE customers SET 2 cust email='deema@gmail.com' 3 WHERE cust_no=4846; 1 row updated. SQL> COMMIT; Commit complete. SQL>	SQL> UPDATE customers SET 2 cust email='deema@hotmail' 3 WHERE cust_no=4846; 1 row updated. SQL> COMMIT; Commit complete. SQL>
SQL> SELECT cust_email 2 FROM customers 3 WHERE cust no=4846; CUST_EMAIL --------------- deema@gmail.com SQL>	SQL> SELECT cust_email 2 FROM customers 3 WHERE cust no=4846; CUST_EMAIL -------------- deema@hotmail.com SQL>
Network restored. The data pump PS1E203 started automatically.	Network restored. The data pump PT1E203 started automatically.
SQL> SELECT cust_email 2 FROM customers 3 WHERE cust_no=4846; CUST_EMAIL ------------- deema@hotmail SQL>	SQL> SELECT cust_email 2 FROM customers 3 WHERE cust_no=4846; CUST_EMAIL --------------- conflict@pt.com SQL>

Refer to Chapter 11 to learn more regarding generating the definition file and advanced transformation.

Applying Net Difference Resolution Using SQLEXEC

When the application manipulates a single entity such as inventory level, bank account, order entry, etc., in an active-active bi-directional Oracle GoldenGate topology, conflicts

may occur and must be handled. Figure 5-22 illustrates an ATM workflow for simultaneous transactions executed at the same time.

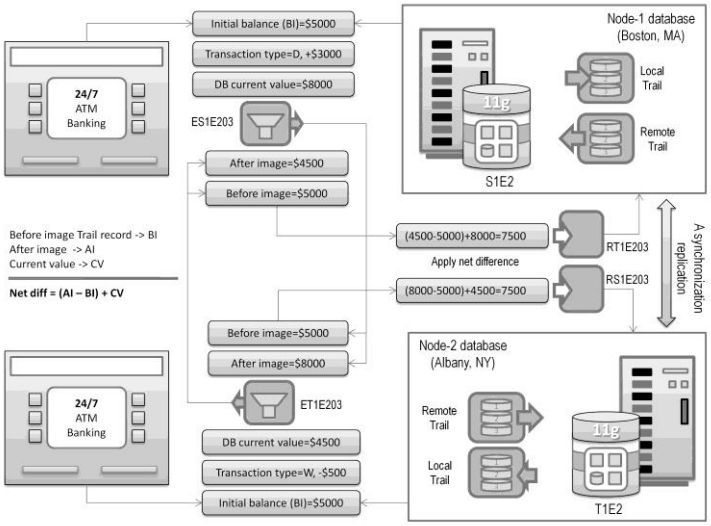


Figure 5-22: Applying Net Difference Resolution Using SQLEXEC

The initial account balance is $5,000. A deposit transaction (D) for the amount of $3,000 is entered using the primary ATM card. This transaction is located at Boston, MA increasing the account balance to $8,000.

Almost simultaneously, while the initial balance is still $5000, a withdrawal transaction (W) for the amount of $500 is entered using the secondary ATM card. This transaction is located at Albany, NY decreasing the account balance to $4,500.

When the Replicat applies the transaction on either database, conflict occurs. The database located at Boston, MA has a current value of $8,000 and the database located at Albany, NY has a current value of $4,500. By comparing the trail record and the current database values, the data conflict is detected and needs to be resolved — using either SQLEXEC for applying the net difference or Oracle GoldenGate built-in conflict detection and resolution (CDR) to calculate and apply the transaction correctly.

The resolution below is developed using SQLEXEC. The Replicat COLMAP clause determines and applies the net difference of: $5000+$3000-$500=$7500.

The net balance applied at Boston, MA is calculated as:

Net balance = ((Trail after image - Trail before image) + current value)
= ((4500 – 5000) + 8000) = $7,500

The net balance applied at Albany, NY is calculated as:

Net balance = ((Trail after image - Trail before image) + current value)
= ((8000 – 5000) + 4500) = $7,500

To demonstrate data conflict by applying the net difference value, the table CURRENT_ACCOUNTS exists on node-1 database (S1E2) and node-2 database (T1E2). It is configured on bi-directional configuration with transactions entered from both nodes.

```
SQL> DESC current_accounts
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 ACCT_NO                                   NOT NULL NUMBER
 ACCT_NAME                                 NOT NULL VARCHAR2(64)
 ACCT_AMOUNT                                        NUMBER
 ACCT_TS                                            TIMESTAMP(6)

SQL>
SQL> SELECT ACCT_AMOUNT
  2  FROM   current_accounts
  3  WHERE  acct_no=49664996;

ACCT_AMOUNT
-----------
       5000

SQL>
```

The Extract parameter file lists the tables. The parameters GETUPDATEBEFORES and GETBEFORECOLS enable capture of the before image values.

```
EXTRACT es1e203
SETENV (ORACLE_SID=S1E2)
SETENV (ORACLE_HOME=/d01/app/oracleSE/product/11.2.0/dbhome_1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
EXTTRAIL ./dirdat/sc
TRANLOGOPTIONS EXCLUDEUSER ggs_admin
GETUPDATEBEFORES
TABLE osm$repapi.customers &
      GETBEFORECOLS(ON UPDATE KEYINCLUDING (cust_email), &
                    ON DELETE KEYINCLUDING (cust_email));
IGNOREUPDATEBEFORES
TABLE osm$repapi.policies;
GETUPDATEBEFORES
TABLE osm$repapi.current_accounts &
      GETBEFORECOLS(ON UPDATE ALL);
IGNOREUPDATEBEFORES
```

The Replicat parameter file uses SQLEXEC to execute a query then compares the before image with the query return value. The COLMAP clause uses Oracle GoldenGate built-in @COMPUTE function to determine the net difference.

```
REPLICAT rs1e203
SETENV (ORACLE SID=T1E2)
SETENV (NSL_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs admin, PASSWORD oracle
SOURCEDEFS /d01/oracle/ggs/dirdef/defgen.def
DISCARDFILE ./dirrpt/rs1e203.dsc, PURGE
GETUPDATES
GETINSERTS
GETDELETES
MAP osm$repapi.customers, TARGET osm$repapi.customers, &
    SQLEXEC (ID     get_email, &
             SPNAME osm$repapi.get_email, &
             PARAMS (pcust_no = cust_no)), &
    COLMAP (cust no        = cust no,
            cust_name      = cust_name,
            cust email     = @IF(@STREQ(BEFORE.cust email,
                               @GETVAL(get_email.pcust_email)), cust_email,
                               'conflict@pt.com'),
            cust_mobile    = cust_mobile,
            cust address   = cust address,
            cust_commit_ts = @GETENV('GGHEADER', 'COMMITTIMESTAMP'));
--
MAP osm$repapi.policies, TARGET osm$repapi.policies, &
    SQLEXEC (SPNAME osm$repapi.lookup_pt, &
             ID     lookup pt, &
             PARAMS (pt_code_param = pt_code)), &
             COLMAP (USEDEFAULTS, &
                     pt_desc = @GETVAL(lookup_pt.pt_desc_param));
--
MAP osm$repapi.current_accounts, TARGET osm$repapi.current_accounts,
    SQLEXEC (ID lookup acct amount, &
        QUERY " SELECT acct_amount FROM osm$repapi.current_accounts
                WHERE  acct no = :vacct no ", &
        PARAMS (vacct_no = acct_no)), &
        COLMAP (acct_no     = acct_no,
                acct_name   = acct_name,
                acct amount = @COMPUTE(((acct amount - BEFORE.acct amount) +
                              @GETVAL(lookup_acct_amount.acct_amount))),
                acct_ts     = @GETENV('GGHEADER', 'COMMITTIMESTAMP'));
```

The below sequence of query and update statements demonstrates the same ACCT_NO is updated when the data pump is ABENDED due to network failure, which should lead to data conflict when the data pump is re-started. However, by applying the net difference value, the ACCT_AMOUNT value column is correctly calculated and converged.

Node-1 database, Boston	Node-2 database, Albany
SQL> SELECT ACCT_AMOUNT 2 FROM current accounts 3 WHERE acct_no=49664996; ACCT_AMOUNT	SQL> SELECT ACCT_AMOUNT 2 FROM current accounts 3 WHERE acct_no=49664996; ACCT_AMOUNT

``` ----------- 5000 SQL> ```	``` ----------- 5000 SQL> ```
Data pump PS1E203 is down due to network failure	Data pump PT1E203 is down due to network failure
``` SQL> UPDATE current accounts SET 2 acct_amount=acct_amount+3000 3 WHERE acct_no=49664996; 1 row updated. SQL> COMMIT; ```	``` SQL> UPDATE current accounts SET 2 acct_amount=acct_amount-500 3 WHERE acct_no=49664996; 1 row updated. SQL> COMMIT; ```
``` SQL> SELECT acct amount 2 FROM current_accounts 3 WHERE acct no=49664996; ACCT_AMOUNT ----------- 8000 ```	``` SQL> SELECT acct amount 2 FROM current_accounts 3 WHERE acct no=49664996; ACCT_AMOUNT ----------- 4500 ```
Network restored. The data pump PS1E203 started automatically.	Network restored. The data pump PT1E203 started automatically.
``` SQL> SELECT acct_amount 2 FROM current_accounts 3 WHERE acct no=49664996; ACCT_AMOUNT ----------- 7500 SQL> ```	``` SQL> SELECT acct_amount 2 FROM current_accounts 3 WHERE acct no=49664996; ACCT_AMOUNT ----------- 7500 SQL> ```

DML Conflict Types

Fundamentally, conflicts occur and must be handled. Why? It is because Oracle GoldenGate Replicat preserves data integrity when applying transactions. It does so by maintaining the transactions commit order and uses the target database native calls when applying transactions, which have database constraints enabled.

An active-active bi-directional configuration must implement, detect, and resolve data conflict techniques. These conflicts arising from DML transactions manipulate the same row stored on the source and target databases. The resolution is detected and resolved using either built-in or custom developed conflict detection and resolution techniques. The resolution is developed as part of the transaction integrity or by applying basic resolution rules such as timestamps. There are (3) types of conflicts that occur when executing SQL statements on asynchronous mode:

Insert Conflicts

Due to enabled and valid primary key or unique key constraints on tables, insert conflicts occur when duplicate values violate the constraint. Without conflict handling the Replicat terminates with an error.

Update Conflicts

Modifying the same row at the same time results in update conflicts. However, because access to the updated row's before and after image is available, conflicts are resolved by applying suitable resolution methods.

Delete Conflicts

This type of conflict commonly occurs when the Replicat attempts to apply a delete transaction, but the row is not found on the target database. This is because the row is already deleted or updated by another transaction.

Data Conflict and Resolution Roadmap

The fundamental rule when developing conflict resolution is to consider the technique consistency. This allows applying the same technique across the vast majority of the database tables. There are several approaches to data conflict avoidance, detection, and resolution. The combination of avoidance and resolution promotes data convergence and data consistency. Oracle GoldenGate provides its own built-in function and conflict detection and resolution (CDR). It is the distinctive feature of Oracle GoldenGate 11g Release 2, explained and demonstrated shortly.

The techniques and order for handling data conflicts in an active-active bi-directional topology is shown in Figure 5-23. Since Oracle GoldenGate is mostly deployed after the application, it is less likely the application is conflict-aware. To minimize the application changes and make Oracle GoldenGate data replication transparent on bi-directional topology, the roadmap begins by implementing data segmentation as an avoidance technique. When data segmentation may not totally avoid data conflict, partitioning the applications provides the second line of defense.

In addition to data segmentation and application partitioning, Oracle GoldenGate built-in conflict detection and resolution applies consistent methods to resolve conflicts using declarative parameter settings for all types of DML conflict types.

For complex data conflict detection and resolution, using the SQLEXEC parameter requires developing PL/SQL sub-programs and SQL queries that are invoked from within Oracle GoldenGate.

When SQLEXEC is not meeting the requirements, user exits is another option to consider for handling sophisticated conflicts beyond those provided by Oracle GoldenGate built-in functions and techniques. User exits are developed in C and C++ programming language and invoked from the Extract and Replicat, but mostly adapted for advanced integration and data transformation.

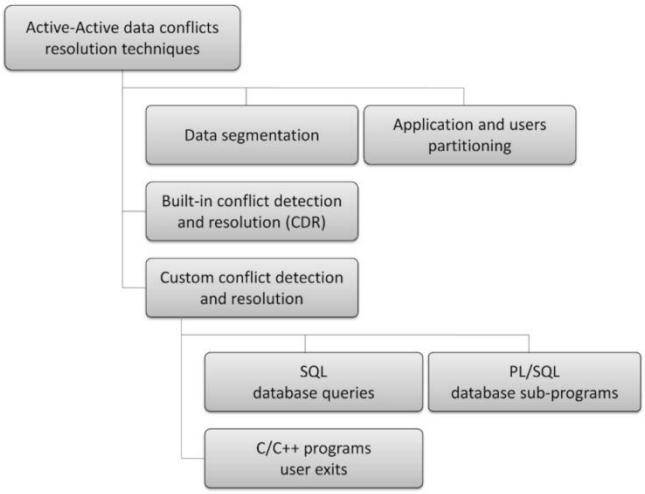


Figure 5-23: Conflicts Resolutions Roadmap

Data Segmentation

When the application is not replication-aware, then implementing data segmentation is the first approach of the data conflict avoidance technique. It is developed and deployed within the database only. An example is implementing Oracle Virtual Private Database (VPD) to dynamically apply a table-level security policy function to avoid the likelihood of having the same row concurrently accessed by multiple users.

Application Segmentation

Application segmentation is implemented from within the virtual private database – making the application replication-aware. Then by segregating applications so that groups of users are associated with an application, accessing a subset of tables reduces the possibility of data conflicts. Application segmentation is developed using application

contexts such as locations, transaction type, etc. This is an avoidance technique to minimize and not to entirely eliminate data conflicts.

Built-In Conflict Detection and Resolution

Oracle GoldenGate 11g Release 2 is shipped with built-in conflict detection and resolution (CDR). This is a core feature of any data replication software. Using simple declarative conflicts and methods, Oracle GoldenGate handles the majority of conflicts for database applications accessing active-active bi-directional topology.

Custom Conflict Resolution

Using queries, PL/SQL sub-programs, and user exits enable developing conflict resolution beyond the capabilities of Oracle GoldenGate built-in conflict resolution methods. The technique extends the capabilities of Oracle GoldenGate to include using the database built-in features for events handling.

Built-In Conflict Detection and Resolution

While complex business logic custom data conflicts resolutions are developed using PL/SQL, SQL queries, and C/C++ user exits, the majority of data conflicts can be handled by adapting Oracle GoldenGate automatic conflicts detection and resolution (CDR) techniques.

The advantages of employing the built-in CDR are performance, consistency, and lower maintenance cost. Resolving conflicts using CDR delivers better performance when compared to custom resolution using SQLEXEC to invoke database PL/SQL procedures or SQL executing queries. The built-in resolution and detection avoid runtime context switching. Since the vast majority of tables are handled using similar conflict resolution methods, CDR delivers consistency using declarative conflict types and methods. Lower maintenance cost is also an advantage of using CDR. For example, when modifying CDR methods, deploying CDR new features, adding new table columns and database tables, maintenance is executed within GoldenGate only. Oracle GoldenGate built-in conflict detection and resolution uses the four parameters listed in Table 5-4, promoting ease of configuration and deployment.

Parameter	Description
GETUPDATEBEFORES	An Extract parameter. It forces capture of the transaction post values (after image) and before image values for use by the Replicat to detect conflicts.
GETBEFORECOLS	An Extract parameter. It associates the before image with

	the operation type of UPDATE, DELETE, or both and the column list.
COMPARECOLS	A Replicat parameter. It lists the columns to check for conflicts with the target database current value. A mismatch is a conflict and handled by the RESOLVECONFLICT settings.
RESOLVECONFLICT	A Replicat parameter. It specifies the operation type, the resolution type, and the method applied to resolve the conflict. The parameter is repeated for each resolution to handle the different operations.

Table 5-4: Built-In Conflict Detection and Resolution Parameters

For tables referenced by transactions on an active-active bi-directional topology, the parameter RESOLVECONFLICT is configured to handle conflicts. Within the same MAP statement, it resolves all types of data conflict transactions for UPDATE, DELETE, and INSERT SQL statements. The below illustrates resolving update conflicts. The named resolution DELTA_RESOLUTION_METHOD uses the method USEDELTA for the column ACCT_AMOUNT. The resolution MAX_RESOLUTION_METHOD uses the method USEMAX for the columns ACCT_NAME and ACCT_TS.

```
RESOLVECONFLICT (UPDATEROWEXISTS,
                    (DELTA_RESOLUTION_METHOD, USEDELTA, COLS (acct_amount)),
                    (MAX_RESOLUTION_METHOD, USEMAX(acct_ts),
                     COLS (acct_name, acct_ts))
```

Figure 5-24 depicts the hierarchy of using Oracle GoldenGate built-in conflict detection and resolution (CDR).

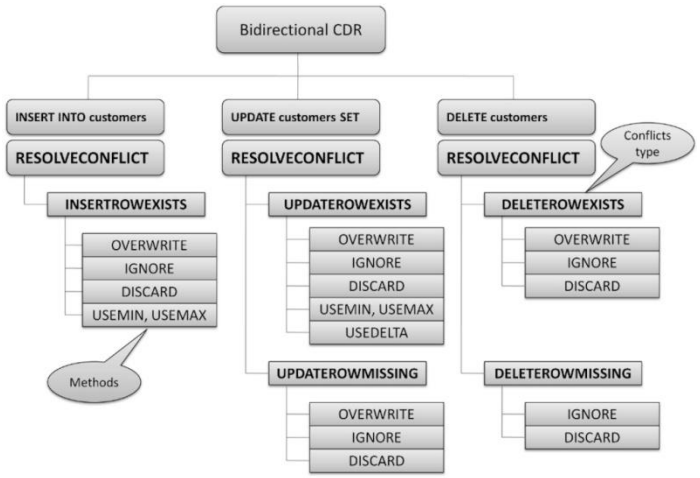


Figure 5-24: Built-In Conflicts Detection and Resolution

The parameter RESOLVECONFLICT takes any of the arguments listed in Table 5-5. Depending on the business rules for data conflict resolution handling, associate the argument with a resolution method.

Conflict type	Description
INSERTROWEXISTS	Handles unique constraint violation. The target database has primary key or unique key constraint enabled.
UPDATEROWEXISTS	Handles update data conflict when the before image value for one or more columns differ from the current database value.
UPDATEROWMISSING	Handles data conflict when the row does not exist on the target database.
DELETEROWEXISTS	Handles delete data conflict when before image value for one or more columns differ from the current database value.
DELETEROWMISSING	Handles data conflict when the row deleted on source does not exist on the target database.

Table 5-5: RESOVLECONFLICT Parameter Conflict Types

Each conflict type is associated with one resolution method. The following is the list of methods applied to a DEFAULT or named resolution:

- OVERWRITE
- IGNORE
- DISCARD
- USEMIN
- USEMAX
- USEDELTA

Figure 5-25 outlines the syntax of the Replicat built-in conflict detection and resolution. The combination of COMPARECOLS and RESOLVECONFLICT parameters handles the operation types and the conflict types. For each conflict type, a resolution method is selected and may be applied to one or more columns.

```
MAP  <source table>, TARGET <target table>,
COMPARECOLS (
  { ON UPDATE {ALL | KEY | KEYINCLUDING(col1, col2,..., coln) |
                          KEYEXCULDING(col1, col2,..., coln)}|
   ,ON DELETE {ALL | KEY | KEYINCLUDING(col1, col2,..., coln) |
                          KEYEXCULDING(col1, col2,..., coln)} )
RESOLVECONFLICT (
    { INSERTROWEXISTS  |
      UPDATEROWEXISTS  | UPDATEROWMISSING |
      DELETEROWEXISTS  | DELETEROWMISSING }
                    (
                      { DEFAULT | <resolution name>},
                      { USEMAX (column) |
                        USEMIN (column) |
                        DELTA           |
                        DISCARD         |
                        OVERWRITE }
                    ),
                       COLS (col1, col2,..., coln)
                  )
```

Figure 5-25: Built-In Conflict Detection and Resolution Syntax

Configuring Built-In Conflict Detection and Resolution

This section implements active-active configuration using Oracle GoldenGate built-in conflict detection and resolution (CDR). Consider the CURRENT_ACCOUNTS table, the Extract and Replicat are configured for CDR. The ACCT_AMOUNT column is handled using the DELTA method. The ACCT_NAME and ACCT_TS columns are handled using the USEMAX method.

```
SQL> DESC current_accounts
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 ACCT_NO                                   NOT NULL NUMBER
 ACCT_NAME                                 NOT NULL VARCHAR2(64)
 ACCT_AMOUNT                                        NUMBER
 ACCT_TS                                            TIMESTAMP(6)

SQL>
```

As part of the change data capture (CDC), ensure to run the ADD TRANDATA command on node-1 and node-2 databases using the syntax below.

```
GGSCI (apps-srv1) 4> ADD TRANDATA osm$repapi.current_accounts ALLCOLS
```

Node-1 Extract and Replicat Parameter Files

The parameter files for node-1 configured to support the Extract ES1E203, the data pump PS1E203 and the Replicat RT1E203.

The use of the parameters GETUPDATEBEFORES, GETBEFORECOLS and IGNOREUPDATEBEFORES is for supporting capture of the before image on the trail records.

```
EXTRACT es1e203
SETENV (ORACLE_SID=S1E2)
SETENV (ORACLE_HOME=/d01/app/oracleSE/product/11.2.0/dbhome_1)
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
EXTTRAIL ./dirdat/sc
TRANLOGOPTIONS EXCLUDEUSER ggs_admin
GETUPDATEBEFORES
TABLE osm$repapi.customers &
      GETBEFORECOLS(ON UPDATE KEYINCLUDING (cust_email), &
                    ON DELETE KEYINCLUDING (cust_email));
IGNOREUPDATEBEFORES
TABLE osm$repapi.policies;
GETUPDATEBEFORES
TABLE osm$repapi.current_accounts &
      GETBEFORECOLS(ON UPDATE ALL, ON DELETE ALL);
IGNOREUPDATEBEFORES
```

The data pump PS1E203 is configured on PASSTHRU mode, sending data to node-2 without transformation.

```
EXTRACT ps1e203
USERID ggs_admin, PASSWORD oracle
PASSTHRU
RMTHOST apps-srv2, MGRPORT 7806
RMTTRAIL ./dirdat/tc
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
TABLE osm$repapi.current_accounts;
```

The Replicat RT1E203 applies transactions to the node-1 database.

```
REPLICAT rt1e203
SETENV (ORACLE_SID=S1E2)
SETENV (NSL_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rt1e203.dsc, PURGE
MAP osm$repapi.customers, TARGET osm$repapi.customers;
MAP osm$repapi.policies, TARGET osm$repapi.policies;
MAP osm$repapi.current_accounts, TARGET osm$repapi.current_accounts,
    COMPARECOLS (ON UPDATE ALL, ON DELETE ALL),
    RESOLVECONFLICT (UPDATEROWEXISTS,
```

```
                    (DELTA_RESOLUTION_METHOD, USEDELTA, COLS (acct_amount)),
                    (MAX_RESOLUTION_METHOD, USEMAX(acct_ts),
                     COLS (acct name, acct ts)),
                                        (DEFAULT, USEMAX(acct_ts))),
    RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)),
    RESOLVECONFLICT (DELETEROWEXISTS,  (DEFAULT, OVERWRITE)),
    RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD)),
    RESOLVECONFLICT (INSERTROWEXISTS,  (DEFAULT, USEMAX(acct_ts)));
```

Node-2 Extract and Replicat Parameter Files

The set of node-2 parameter files are similar to node-1. The parameter files support the Extract ET1E203, the data pump PT1E203 and the Replicat RS1E203.

The Extract ET1E203 parameter file.

```
EXTRACT et1e203
SETENV (ORACLE SID="T1E2")
SETENV (ORACLE_HOME="/d01/app/oracleSE/product/11.2.0/dbhome_1")
SETENV (NLS_LANG="AMERICAN_AMERICA.AL32UTF8")
USERID ggs_admin, PASSWORD oracle
TRANLOGOPTIONS EXCLUDEUSER ggs_admin
EXTTRAIL ./dirdat/sd
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
GETUPDATEBEFORES
TABLE osm$repapi.current_accounts &
      GETBEFORECOLS(ON UPDATE ALL, ON DELETE ALL);
IGNOREUPDATEBEFORES
```

The data pump PT1E203 parameter file.

```
EXTRACT pt1e203
USERID ggs admin, PASSWORD oracle
PASSTHRU
RMTHOST apps-srv1, MGRPORT 7805
RMTTRAIL ./dirdat/td
TABLE osm$repapi.customers;
TABLE osm$repapi.policies;
TABLE osm$repapi.current_accounts;
```

The Replicat RS1E203 parameter file.

```
REPLICAT rs1e203
SETENV (ORACLE_SID=T1E2)
SETENV (NSL LANG=AMERICAN AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rs1e203.dsc, PURGE
GETUPDATES
GETINSERTS
GETDELETES
```

```
MAP osm$repapi.customers, TARGET osm$repapi.customers, &
    SQLEXEC (ID get_email, &
             SPNAME osm$repapi.get email, &
             PARAMS (pcust_no = cust_no)), &
             COLMAP (cust_no        = cust_no,
                     cust_name      = cust_name,
                     cust_email     = @IF(@STREQ(BEFORE.cust_email,
                                      @GETVAL(get_email.pcust_email)),
                                      cust email, 'conflict@pt.com'),
                     cust_mobile    = cust_mobile,
                     cust address   = cust address,
                     cust_commit_ts = @GETENV('GGHEADER',
                                              'COMMITTIMESTAMP'));
--
MAP osm$repapi.policies, TARGET osm$repapi.policies, &
    SQLEXEC (SPNAME osm$repapi.lookup_pt, &
             ID     lookup_pt, &
             PARAMS (pt_code_param = pt_code)), &
             COLMAP (USEDEFAULTS, &
                     pt_desc = @GETVAL(lookup_pt.pt_desc_param));
--
MAP osm$repapi.current_accounts, TARGET osm$repapi.current_accounts,
    COMPARECOLS (ON UPDATE ALL, ON DELETE ALL),
    RESOLVECONFLICT (UPDATEROWEXISTS,
                     (DELTA_RESOLUTION_METHOD, USEDELTA, COLS (acct_amount)),
                     (MAX RESOLUTION METHOD, USEMAX(acct ts),
                                             COLS (acct name, acct ts)),
                                       (DEFAULT, USEMAX(acct ts))),
    RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)),
    RESOLVECONFLICT (DELETEROWEXISTS,  (DEFAULT, OVERWRITE)),
    RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD)),
    RESOLVECONFLICT (INSERTROWEXISTS,  (DEFAULT, USEMAX(acct_ts)));
```

Highly Available Applications

Most call center's applications are the front line for customer service companies that provide 24/7 services. Requests and inquiry calls coming from various input devices such as mobile phones, smart phone applications, and browser-based applications are system received and routed to the next available agent. Figure 5-26 shows the workflow for incoming customer calls. This is a high level description to demonstrate using Oracle Virtual Private Database (VPD) in supporting the data conflict avoidance technique for this kind of business environment.

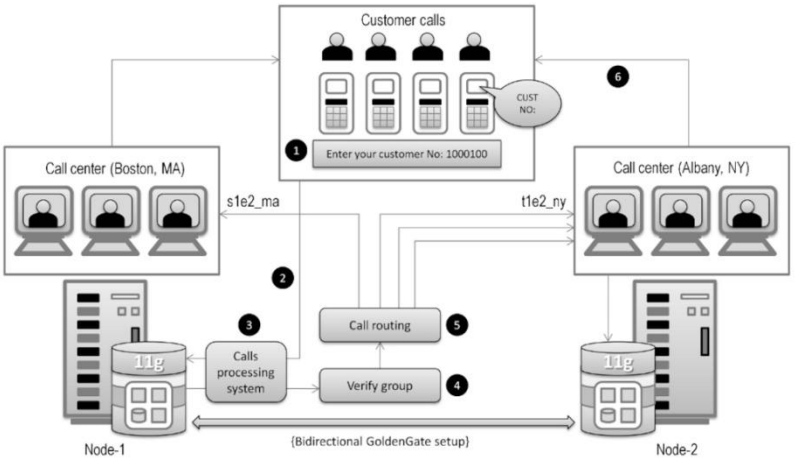


Figure 5-26: Call Center Applications Workflow

The customer calls in for inquires, such as requesting a policy renewal quotation. An automated system welcomes the customer. Then, the customer call is processed by assigning an agent to fulfill the request as follows:

1. The caller enters their customer number, which is the unique identifier.
2. The number is sent to the current database dedicated for receiving calls.
3. The call processing system validates the call and updates the calls log.
4. Based on the customer number, the owner group is determined.
5. Call routing is made based on the group name of the customer number.
6. An assigned agent receives the routed call and is locked with the customer.

The next section is developed on the basis of the above applications workflow. Data separation is achieved by routing requests and inquires to either Boston or Albany call centers. The objective is to maximize the return of investment (ROI) from licensing Oracle Database Enterprise Edition and Oracle GoldenGate software bundles for achieving a two-node architecture as an active-active topology applied for high availability and disaster recovery business requirements.

Transparent Avoidance Using Virtual Private Database

Virtual Private Database (VPD) is an advanced data conflict avoidance technique for transparent data segmentation. After creating database's table polices and implementing fine-grain access control and application contexts, SQL statements are dynamically modified to control rows (and columns) level access by database users. Figure 5-27 shows how users on node-1 and on node-2 access different subset of data, even when the session

is connected to the same database. Fine-grain access control limits rows access by implementing a database function that returns a predicate clause (WHERE) to virtually limit row access by the database users.

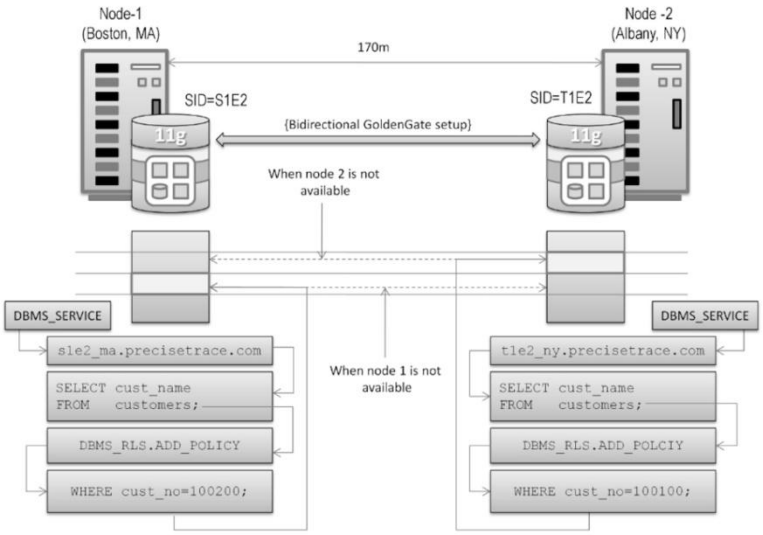


Figure 5-27: Data Segmentation Using Virtual Private Database

Our objective for using Virtual Private Database is to enforce data access based on ownership, so that a group of users within a database sees only their own accounts. The implementation is transparent – defined using policies that are independent of the applications. These active policies are cached in the SGA shared pool allowing in-memory access for faster evaluation and processing of fine-grain access control using SYS_CONTEXT function.

Begin by using DBMS_SERVICE PL/SQL supplied package to create non-default service names which will be used to associate a database user to one-and-only-one group at any time. However, in the event of a node failure, the user connects to the surviving node using the same service name but referencing the surviving node database service. Figure 5-27 shows the service name s1e2_ma for users connecting to node-1 database and t1e2_ny for users connecting to node-2 database.

The calls to DBMS_SERVICE.CREATE_SERVICE create the service names s1e2_ma and t1e2_ny, then start these service names for user connections. The procedure DBMS_SERVICE.CREATE_SERVICE requires the service name and service network name, use the service name to manage the service name.

```
SQL> BEGIN
  2   DBMS_SERVICE.CREATE_SERVICE(service_name=>'s1e2_ma',
  3                               network_name=>'s1e2_am.precisetrace.com');
  4  END;
  5  /

PL/SQL procedure successfully completed.

SQL> EXEC DBMS_SERVICE.START_SERVICE('s1e2_ma');

PL/SQL procedure successfully completed.

SQL> BEGIN
  2   DBMS_SERVICE.CREATE_SERVICE(service_name=>'t1e2_ny',
  3                               network_name=>'t1e2_ny.precisetrace.com');
  4  END;
  5  /

PL/SQL procedure successfully completed.

SQL> EXEC DBMS_SERVICE.START_SERVICE('t1e2_ny');

PL/SQL procedure successfully completed.
```

Use 'netca utility' to create the database connect string for the new service names and perform quick tests. By creating the two service names on node-1 and node-2 instances, they are readily available in the event of either node failure. The connect string for s1e2_ma is configured (t1e2_ny is not shown), followed by a successful test using the connect string s1e2_ma and t1e2_ny.

```
S1E2_MA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = ggs_source)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = s1e2_ma.precisetrace.com)
    )
  )

SQL> CONN fawzi/oracle@s1e2_ma
Connected.
SQL> CONN deema/oracle@t1e2_ny
Connected.
SQL>
```

How Does Virtual Private Database Work?

The major advantage for using Virtual Private Database is ease of deployment and flexibility. After the policy is created, developed, and enabled, it becomes part of the database and access control applies across all applications of the table or view. The workflow of executing an SQL application enforced by the Virtual Private Database security policy on the CUSTOMERS table is as follows:

1. The user accesses the CUSTOMERS table using SQL applications.
2. The application invokes the package for preparing and setting the context attribute values.
3. The policy function returns the predicate. It uses the session application context attributes stored in memory.
4. The database server dynamically modifies the statement by appending the returned predicate to SQL applications of step 1.
5. The session executes the dynamically generated SQL applications.

Implementing Virtual Private Database

The implementation of Virtual Private Database may require designing the database to provide the application context attribute values at runtime. Figure 5-28 shows a modified data model where customers are created by a group, meaning the customers and policies are owned and modified by a specific location (group). This model provides data conflict avoidance as the data is modified by transactions within the same database.

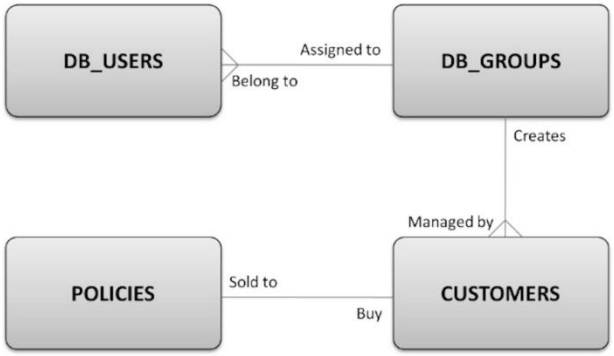


Figure 5-28: Data Modeling for VPD

The table structure of Figure 5-28 is listed below. The column GROUP_ID for each customer supports developing a conflict avoidance by group (location) and not by users. Transactions are processed as one unit of work and sent to the remote database after being processed by the local database.

```
SQL> DESC db_groups
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 GROUP_ID                                  NOT NULL VARCHAR2(30)
 GROUP_DESC                                NOT NULL VARCHAR2(128)

SQL> DESC db_users
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
```

```
 USER_ID                                   NOT NULL VARCHAR2(30)
 USER_NAME                                 NOT NULL VARCHAR2(128)
 GROUP_ID                                  NOT NULL VARCHAR2(30)

SQL> DESC customers
 Name                                      Null? Type
 ----------------------------------------- -------- ----------------------
 CUST_NO                                   NOT NULL NUMBER
 CUST_NAME                                 NOT NULL VARCHAR2(128)
 CUST_EMAIL                                NOT NULL VARCHAR2(128)
 CUST_MOBILE                                        VARCHAR2(16)
 CUST_ADDRESS                                       VARCHAR2(128)
 GROUP_ID                                  NOT NULL VARCHAR2(30)

SQL> DESC policies
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 POL_NO                                    NOT NULL NUMBER
 POL_FROM                                  NOT NULL DATE
 POL_TO                                    NOT NULL DATE
 POL_VALUE                                 NOT NULL NUMBER(8,2)
 POL_SUB_TOTAL                             NOT NULL NUMBER(8,2)
 PT_CODE                                   NOT NULL VARCHAR2(8)
 DT_CODE                                            VARCHAR2(8)
 POL_TOTAL                                 NOT NULL NUMBER(8,2)
 CUST_NO                                   NOT NULL NUMBER

SQL>
```

Next, using the PL/SQL supplied package DBMS_RLS to create the CUSTOMERS table, access control policy named CUST_POLICY_VPD adapted for transactions of type SELECT, UPDATE, INSERT, and DELETE.

Creating the Application Groups and Users

The groups are created to identify users' location. Each group is associated with a specific instance service name. The below INSERT statements creates two groups. Each correspond to a service name.

```
SQL> INSERT INTO db_groups VALUES('s1e2_ma', 's1e2_ma.precisetrace.com');

1 row created.

SQL> INSERT INTO db_groups VALUES('t1e2_ny', 't1e2_ny.precisetrace.com');

1 row created.

SQL> COMMIT;

Commit complete.

SQL>
```

Then, the users are created using the format to associate them with the service network name for node 1 or node 2, but not both. For the database username 'fawzi', the application username is 'fawzi$s1e2_ma.precisetrace.com.'

```
SQL> INSERT INTO db_users VALUES
  2  ('fawzi$s1e2_ma.precisetrace.com', 'F. Alswaimil', 's1e2_ma');

1 row created.

SQL> INSERT INTO db_users VALUES
  2  ('deema$t1e2_ny.precisetrace.com', 'D. Alswaimil', 't1e2_ny');

1 row created.

SQL> COMMIT;

Commit complete.

SQL>
```

Virtual Private Database Prerequisite

There are several related virtual private policy privileges that need to be granted to certain users for managing and troubleshooting objects encompassed by virtual private database policies. The following DDL statements grant capabilities to the application owner, Oracle GoldenGate database users, and SYSTEM administrative when working with tables and views enforced by Virtual Private Database.

```
SQL> CONN / as sysdba
Connected.
SQL> GRANT EXEMPT ACCESS POLICY TO SYSTEM;

Grant succeeded.

SQL> GRANT EXEMPT ACCESS POLICY TO ggs_admin;

Grant succeeded.

SQL> GRANT EXEMPT ACCESS POLICY TO osm$repapi;

Grant succeeded.

SQL> GRANT EXECUTE ON DBMS_RLS to osm$repapi;

Grant succeeded.

SQL>
```

Create the Security Policy

Use the procedure DBMS_RLS.ADD_POLICY to create the policy CUST_POLICY_VPD. This implements fine grain row level access on

the CUSTOMERS table. The STATEMENT_TYPES parameter lists the SQL application types enforced by this policy. The POLICY_FUNCTION parameter specifies the function name that returns the predicate.

```
SQL> BEGIN
  2     DBMS_RLS.ADD_POLICY(
  3        object_schema=>'system',
  4        object_name=>'customers',
  5        policy_name=>'cust_policy_vpd',
  6        function_schema=>'system',
  7        policy_function=>'ggs_security_p.cust_order',
  8        statement_types=>'select, insert, update, delete');
  9  END;
 10  /

PL/SQL procedure successfully completed.

SQL>
```

Create the Context Management Package

Develop the package to prepare the application context using the function SYS_CONTEXT and set the context attributes values using the procedure DBMS_SESSION.SET_CONTEXT. For all transaction types, the package set the value for context attribute GROUP_ID according to the USER_ID.

```
SQL> CREATE OR REPLACE PACKAGE ggs_security is
  2      PROCEDURE set_user_group;
  3  END;
  4  /

Package created.

SQL> CREATE OR REPLACE PACKAGE BODY ggs_security is
  2     PROCEDURE set_user_group(puser_id varchar2)  IS
  3        vgroup_id VARCHAR2(30);
  4     BEGIN
  5        SELECT group_id into vgroup_id
  6        FROM   db_users
  7        WHERE  UPPER(user_id) =
  8               UPPER(puser_id)||'$'||UPPER(SYS_CONTEXT('USERENV',
                  'SERVICE_NAME'));
  9        DBMS_SESSION.SET_CONTEXT('oeapp', 'group_id', vgroup_id);
 10     EXCEPTION
 11        WHEN OTHERS THEN
 12             DBMS_SESSION.SET_CONTEXT('oeapp', 'group_id', NULL);
 13     END;
 14  END;
 15  /

Package body created.

SQL>
```

Create the Security Function

Develop the function to return the predicate clause (WHERE). The return predicate is appended to the existing where clause is using AND as logical operator.

```
SQL> CREATE OR REPLACE PACKAGE ggs_security_p is
  2      FUNCTION cust_order(object_schema VARCHAR2, object_name VARCHAR2)
  3      RETURN VARCHAR2;
  4  END;
  5  /

Package created.

SQL> CREATE OR REPLACE PACKAGE BODY ggs_security_p is
  2      FUNCTION cust_order(object_schema VARCHAR2, object_name VARCHAR2)
  3      RETURN VARCHAR2 IS
  4      BEGIN
  5         RETURN 'group_id=SYS_CONTEXT(''oeapp'', ''group_id'')';
  6      END;
  7  END;
  8  /

Package body created.

SQL>
```

Create the Database Application Context

Connect as privileged user to create the context. The context references the package created above.

```
SQL> CREATE CONTEXT oeapp USING ggs_security;

Context created.

SQL>
```

Execute the Application

Now, it is time to run the application accessing tables enforced by Virtual Private Database. Depending on the database username, the context attributes assign runtime values allowing dynamic modified SQL application to be executed.

Figure 5-29 shows the original SQL statement tunnel through the security policy CUST_POLICY_VPD, the ultimate modified SQL statement restricts access only to rows that are within the context of the username session.



Figure 5-29: Dynamic SQL Modification for Data Access Control

The application starts by invoking the package that prepares and initializes the context attributes, which are used by the security function that return the predicate clause. The application heavily relies on the built-in database function SYS_CONTEXT which uses a pre-defined list of attributes such as DB_NAME and SERVICE_NAME for application runtime.

The following shows how data segmentation is achieved. It considers the combination of the database username and the service name to produce an active row set specific to the user's session. The following demonstrates the row set when connected to the Albany, NY system.

```
SQL> CONN deema/oracle@t1e2_ny
Connected.
SQL> DECLARE
  2     CURSOR c1 IS SELECT * FROM system.customers;
  3     vcust system.customers%ROWTYPE;
  4  BEGIN
  5     SYSTEM.GGS_SECURITY.SET_USER_GROUP(USER);
  6     OPEN c1;
  7     LOOP
  8        FETCH c1 into vcust;
  9        EXIT WHEN c1%NOTFOUND;
 10        DBMS_OUTPUT.PUT_LINE(vcust.cust_name);
 11     END LOOP;
 12     CLOSE c1;
 13  END;
 14  /
CAROL P. THOMPSON
MARK B. MILLER
RONALD H. ALLEN
NANCY U. WILSON
JESSICA Z. WILSON
DAVID O. JOHNSON
SANDRA J. ALSWAIMIL
SARAH N. KING
KENNETH O. ROBINSON
LINDA K. WILSON

PL/SQL procedure successfully completed.
```

The following demonstrates when the user connects to the Boston, MA system. It is

the same application driven by dynamic context attributes and values to retrieve different active row sets.

```
SQL> CONN fawzi/oracle@s1e2_ma
Connected.
SQL> DECLARE
  2     CURSOR c1 IS SELECT * FROM system.customers;
  3     vcust system.customers%ROWTYPE;
  4  BEGIN
  5     SYSTEM.GGS_SECURITY.SET_USER_GROUP(USER);
  6     OPEN c1;
  7     LOOP
  8        FETCH c1 into vcust;
  9        EXIT WHEN c1%NOTFOUND;
 10        DBMS_OUTPUT.PUT_LINE(vcust.cust_name);
 11     END LOOP;
 12     CLOSE c1;
 13  END;
 14  /
JOHN V. KING
NOUR Z. LEWIS
KAREN Y. WALKER
KIMBERLY O. THOMPSON
KAREN Q. ALIBRAHIM
SARAH P. WHITE
JOHN C. DAVIS
KEVIN H. ALIBRAHIM
MARK L. MILLER
CHARLES N. WALKER

PL/SQL procedure successfully completed.

SQL>
```

Virtual Private Database Considerations

Be aware of Virtual Private Database (VPD) as an Oracle Database Enterprise feature. It is adapted mutually for data security by developing fine-grain access control at the row level and enhances application performance by reading metadata, context attribute values from Oracle shared pool memory of the System Global Area (SGA). The length of the returned predicate by the security function is limited to 4000 characters only, which may not be sufficient for some complex WHERE clauses.
An alternative of DBMS_RLS is to use Oracle Enterprise Manager to create and manage policies from the graphical user interface (GUI).

Oracle GoldenGate High Availability Runtime

The following illustrates the runtime for high availability on an active-active

bi-directional configuration. The building blocks for the demonstration are:

- Identical Tables Structure
- Oracle GoldenGate Active-Active Bi-Directional Topology Architecture
- Oracle GoldenGate Built-In Conflict Detection and Resolution
- Database Sequences, odd numbers for node-1 and even numbers for node-2
- Database Connections load-balancer for two-node Oracle GoldenGate high availability configuration

Figure 5-30 shows the load balancer used to handle high availability and workload balancing across the two databases, S1E2 and T1E2.

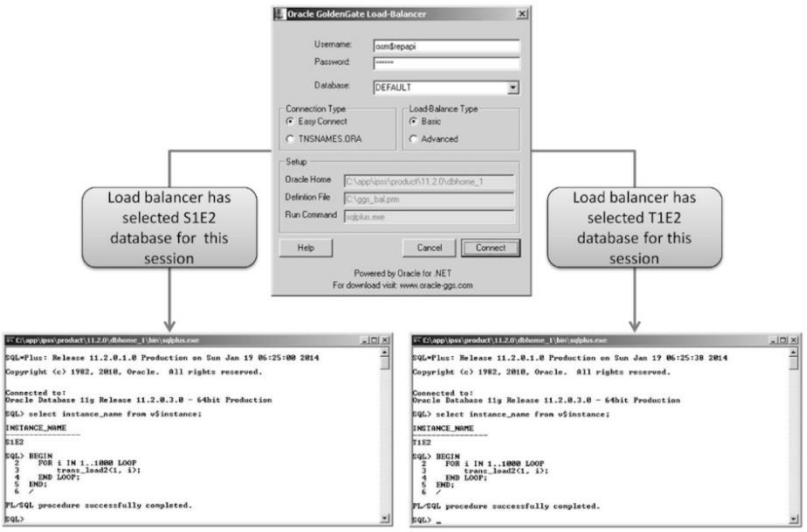


Figure 5-30: Runtime on High Availability Environment

Verify Transactions

The TRANS_LOAD2 is a workload program for generating database transactions. Oracle GoldenGate active-active bi-directional configuration synchronizes node-1 and node-2 databases, keeping the two databases identical using real-time data movement.

The test was completed successfully. The transaction size of 1000 performs concurrent 1000 inserts and replication. The logging table records conflict detection and error, if any.

Database (S1E2)	Database (T1E2)
SQL> BEGIN 2 FOR i IN 1..1000 LOOP 3 trans_load2(1, i);	SQL> BEGIN 2 FOR i IN 1..1000 LOOP 3 trans_load2(1, i);

```
  4     END LOOP;
  5  END
  6  /
```

```
SQL> SELECT cust_no
  2  FROM   customers
  3  WHERE  rownum < 11
  4  ORDER BY 1;

   CUST_NO
----------
   1006007
   1006009
   1006010
   1006011
   1006012
   1006013
   1006014
   1006015
   1006016
   1006017

10 rows selected.

SQL>
```

```
  4     END LOOP;
  5  END
  6  /
```

```
SQL> SELECT cust_no
  2  FROM   customers
  3  WHERE  rownum < 11
  4  ORDER BY 1;

   CUST_NO
----------
   1006007
   1006009
   1006010
   1006011
   1006012
   1006013
   1006014
   1006015
   1006016
   1006017

10 rows selected.

SQL>
```

To effectively monitor data divergence and proactively detect data inconsistency, Oracle GoldenGate Veridata is a robust tool for detecting missing rows between source and target tables. Refer to Chapter 6 to learn about implementing and using Oracle GoldenGate Veridata.

Workload Simulator

Yes, Oracle GoldenGate is a non-destructive software. It is the software installable on a production environment without major negative performance impact. However, use of the workload simulator is not to be underestimated and can be used for:

- Data Replication Testing Environments
- Benchmarking the server performance for different levels of transaction volume
- Ejection of Data Conflict for detection and resolution

To download the TRANS_LOAD2 workload simulator, visit http://www.learn-ogg.com.

Summary

The use of Oracle GoldenGate 12c for business continuity has advantages and disadvantages in using the same bi-directional topology architecture to build the foundation for a disaster recovery site and high availability on supported platforms.

When using Oracle GoldenGate for disaster recovery (DR), the major point of discussion is unsupported data types. For applications not using any of the unsupported data types, Oracle GoldenGate is an option for disaster recovery solutions. Switchover and failover from the primary system to the live standby system and vice versa is an interactive process and requires manual execution of commands or scripts.
The downtime associated with switchover and failover is very low and varies from site-to-site. The chapter has illustrated the following:

- Planned moving of users from primary to live standby system for switchover
- Planned moving of users back from live standby to primary system for switchover
- Unplanned moving of users from primary to live standby system for failover
- Unplanned moving of users back from live standby to primary system for failover

When using Oracle GoldenGate for high availability (HA), detailed analysis is carried out to evaluate Oracle GoldenGate capability of achieving the level of scalability for low-high transactions volume. When Oracle GoldenGate is deployed from the ground up, the applications become replication-aware (transparent replication) which promote Oracle GoldenGate for a high availability platform. However, deploying Oracle GoldenGate at a later stage results in the challenge of handling data conflict.
This chapter has discussed and illustrated techniques of handling data conflicts using:

- Data and Applications Segmentation
- Oracle GoldenGate Built-In Conflict Detection and Resolution (CDR)
- Develop PL/SQL Sub-Programs and implement callout using SQLEXEC
- Oracle Virtual Private Database (VPD) for total segmentation

Throughout the chapter, a two-node bi-directional topology architecture is demonstrated. A three-node topology architecture, multi-master brings challenges and should be carefully qualified for disaster recovery and high availability solutions. Oracle GoldenGate considerations, in respect of disaster recovery and high availability, are discussed to lead to a sound decision and minimize the impact of post-deployment changes.

The chapter also compared Oracle GoldenGate with Oracle Real Application Clusters (RAC) and Oracle Data Guard (DG). It was made clear, for an enterprise, Oracle GoldenGate complements RAC and DG rather than replaces them.

Comparison of High-Volume In-Flight Data Using Oracle GoldenGate Veridata

APPENDIX C

Overview

This appendix covers the Oracle GoldenGate companion tool, *Oracle GoldenGate Veridata*. By all means, the continued deviation on data integrity and consistency severely impacts business reviews and escalates inaccurate results. Hence, the need for a tool to provide online detections and detailed analysis for data discrepancies is crucial and must be accomplished. This is where Oracle GoldenGate Veridata plays the role — supporting data accuracy among heterogeneous data sets by performing advanced table comparisons without impacting online transactions.

To implement Oracle GoldenGate Veridata, it's necessary to perform the following activities:

- Download and install Oracle GoldenGate Veridata agents.
- Download and install Oracle GoldenGate Veridata server instance.
- Explain and configure Oracle GoldenGate Veridata components.
- Explain and configure Oracle GoldenGate Veridata configurable objects.
- Perform compare jobs on compare pairs for compare groups.
- Generate Out-Of-Sync (OOS) summary reports.
- Develop data repair technique for data convergence.

Oracle GoldenGate Veridata becomes the primary tool when using Oracle GoldenGate for delivering:

- Near zero-downtime database upgrade, Oracle GoldenGate Veridata is used to verify zero data-divergence before moving users to the target system.
- High availability, Oracle GoldenGate Veridata compares data set of the primary and the live standby database following failover/switchover operation.
- Unattended online monitor and notification of two data sources without impacting online transactions.

Let us begin by introducing Oracle GoldenGate architecture and building blocks, which are fundamentals for implementing Oracle GoldenGate Veridata.

Architecture

Though there are several topologies to install and configure Oracle GoldenGate Veridata components, the overall architecture remains the same. Figure 6-1 depicts the common architecture, the all-in-one topology, for installing Oracle GoldenGate Veridata. This topology illustrates Oracle GoldenGate Veridata agents installed on the database servers. This avoids the network traffic when the agent is installed remotely. The Oracle GoldenGate Veridata server and its database repository are installed on a dedicated server — configured to communicate with the two agents using JDBC drivers. Apache Tomcat web server is installed, integrated and configured to enable access to Oracle GoldenGate Veridata web interface applications. The Apache Tomcat web server administration tool supports managing groups and users for a secure environment. Oracle GoldenGate Veridata objects configuration deployed using Oracle GoldenGate Veridata server web interface, whereas the command-line utility VERICOM performs basic functions such as configuring and starting jobs.

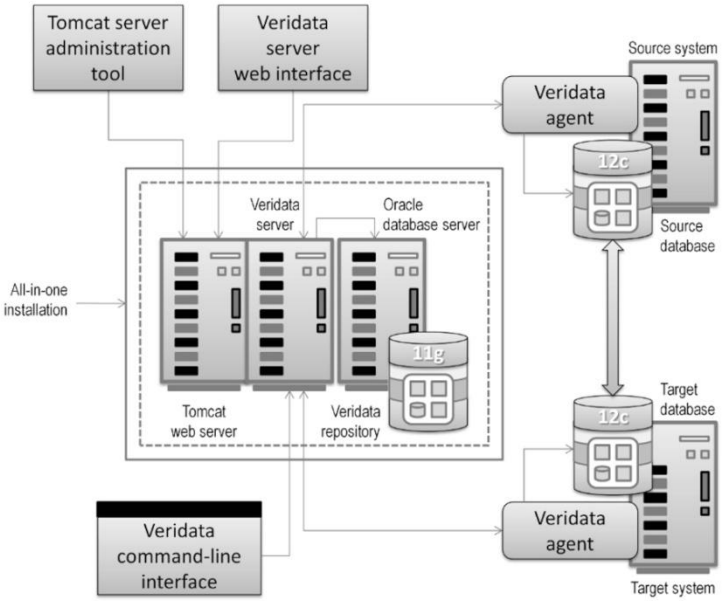


Figure 6-1: Oracle GoldenGate Veridata Architecture (All-In-One Topology)

Regardless of the options and topology adapted to install and configure Oracle GoldenGate Veridata, the components and data flow among them remains the same.

We will work with and manage the following components:

- Oracle GoldenGate Veridata Server

- Oracle GoldenGate Veridata Database Repository
- Oracle GoldenGate Veridata Agents
- Source and Target Database
- Apache Tomcat Web Server
- Oracle GoldenGate Veridata Web Interface
- Oracle GoldenGate Veridata Command-Line Interface

When designing Oracle GoldenGate Veridata topology, the aim is to reduce the processing overhead on the production servers and minimize the overhead of sending data across the network. An optimum topology is achieved by evaluating the alternatives by carefully distributing the components. For example, the agents location is within the database server or remote. Oracle GoldenGate Veridata Database Repository is local or remote, single agent vs. multi-agent configurations, etc.

Before proceeding further, here are brief definitions of these Oracle GoldenGate Veridata components. Refer to Figure 6-1 for logical presentation and further understanding.

Oracle GoldenGate Veridata Server

It is the core of several integrated components. Ultimately, the function of the Oracle GoldenGate Veridata server is to identify out-of-sync pair of rows. This is achieved by performing the following:

- Establishing an agent connection for requesting data fetch from the source and target database.
- Receiving bulk data from the database agent in preparation of rows hash, sort (optional) and compare.
- Coordinating the activities and tasks by executing various Oracle GoldenGate Veridata programs.
- Building an in-memory, out-of-sync queue.
- Confirming out-of-sync data.
- Generating out-of-sync summary and runtime review reports.

When multiple Oracle GoldenGate Veridata server instances reference the same database repository, preferably the shared data location on file system is the same for all instances. Use of network file system (NFS) is necessary for distributed instances.

Oracle GoldenGate Veridata Repository

There are several types of objects configured to enable Oracle GoldenGate Veridata to perform data comparisons and report out-of-sync rows. The metadata about these objects

is stored persistently on a database schema called the repository. There are three supported database types:

1. Oracle Database
2. Oracle MySQL
3. Microsoft SQL Server

Preparation and configuration is database specific. This chapter uses the Oracle Database for the repository. Consider Oracle software license cost of the repository database, as it is not included with Oracle GoldenGate Veridata licensed software.

Oracle GoldenGate Veridata Agents

Upon request by Oracle GoldenGate Veridata server, the agent connects to the target data source for performing sequence of data fetch. Oracle GoldenGate has two types of agents:

1. Java agent (J agent)
2. C agent

Java agent supports a wide range of platforms, since it uses JDBC drivers to connect to the target database. A certified Java Runtime Environment (JRE) is required to operate and manage Java agents. The agent is installable on the database server, on the Oracle GoldenGate Veridata server, or on a dedicated server. Moreover, Oracle provides a custom Java agent built for platforms certified by Oracle GoldenGate.

C agent is only available for Enscribe/SQL/MP and Oracle database. Operating C agents demands Oracle GoldenGate instance and is installable on the database server only. C agent continues to support Oracle database, but there are no plans for further enhancements as of Version 11.2.1.0.0.

This chapter focuses and uses Java agent, but also briefly explains C agent.

Source and Target Database

Oracle GoldenGate Veridata server performs data comparisons from two databases only – source and target. Heterogamous database support is a key feature of the Oracle GoldenGate Veridata server. For example, the source database is an Oracle database while the target database is a Microsoft SQL Server database.

Oracle GoldenGate Veridata server uses internal data types mapping for a heterogamous source and target database. Modifications of the default data types settings are performed using Oracle GoldenGate Veridata web interface.

Apache Tomcat Web Server

It is an Apache HTTP server. It enables access to the Oracle GoldenGate Veridata server administration tools and applications from a browser. Oracle GoldenGate Veridata users' access is configured from within the Tomcat web server. Users, groups, roles, and privileges assignments are performed from within Apache Tomcat web server administration tool. Further data related privileges are configured from Oracle GoldenGate Veridata.

Oracle GoldenGate Veridata Web Interface

A comprehensive set of web-based applications and reports for configuring and managing Oracle GoldenGate Veridata objects. The web interface creates connections, groups, jobs, profiles, and columns mapping. Figure 6-2 shows a Oracle GoldenGate Veridata homepage of a newly installed Oracle GoldenGate Veridata instance. The homepage is composed of the navigation panel, favorites, and display area. Access to the Oracle GoldenGate Veridata web interface requires a valid Tomcat web server username and password.

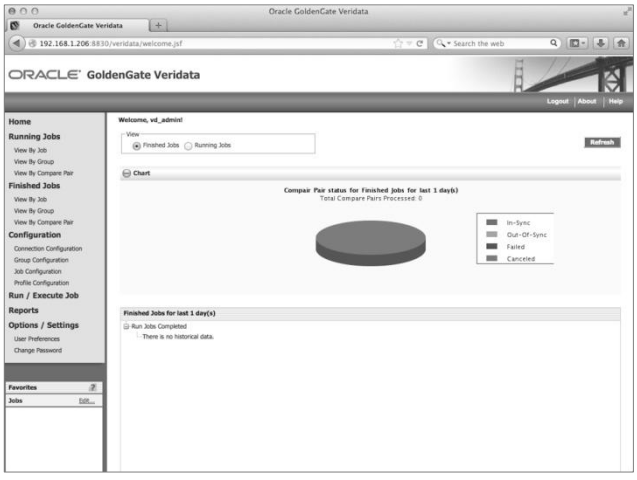


Figure 6-2: Oracle GoldenGate Veridata Web Interface

Oracle GoldenGate Veridata Command-Line Interface

Oracle GoldenGate Veridata Command-Line Interface (CLI), VERICOM, is a client software installed with the Oracle GoldenGate Veridata server. It is intended for performing limited tasks such as running compare jobs, overriding runtime parameters and developing automated batch jobs for data comparison.

Veridata Supported Platforms

Oracle GoldenGate Veridata components (server, agents, and repository) are certified to run on any of the certified combination of database and operating systems listed on Table 6-1. The certification is version specific for the databases and operating systems.

Components	Database	Operating system
Oracle GoldenGate Veridata Java agent	▪ Oracle Database ▪ MS SQL Server ▪ Sybase ▪ Teradata	▪ Oracle Linux ▪ Red Hat EL ▪ Oracle Solaris ▪ Microsoft Windows ▪ IBM AIX ▪ HP-UX ▪ zOS
Oracle GoldenGate Veridata C agent	▪ Oracle Database	
	▪ Enscribe/SQL/MP	▪ HP NonStop
Oracle GoldenGate Veridata server repository	▪ Oracle Database ▪ Oracle MySQL ▪ MS SQL Server	▪ Oracle Linux ▪ Red Hat EL ▪ Oracle Solaris ▪ MS Windows ▪ IBM AIX ▪ HP-UX

Table 6-1: Oracle Veridata Supported Platforms

Customers with a valid Oracle support agreement may request custom builds for Oracle GoldenGate and Java agent for currently unlisted certification. For current certification matrix, visit http://www.learn-ogg.com.

Installing Oracle GoldenGate Veridata Agents

The agent executes tasks on the database on behalf of the Oracle GoldenGate Veridata server. These tasks include:

- Fetch and return blocks of rows to compare
- Return column level details for out-of-sync rows

There are two types of agents, C and Java agents. C agent requires Oracle GoldenGate instance and configuring related manager parameters. Whereas, Java agent is ready to use after configuring the agent parameters file. Java agent is installable on the same system of the source and target database or on a remote server accessing the source and target server over the network. The recommended approach is to install the agent on the same source and target systems — saving the network overhead as a result of returning massive amounts of data to Oracle GoldenGate Veridata server.

C Agent Settings

This agent is written in C language. It is the initial version of the Oracle GoldenGate Veridata agent and requires Oracle GoldenGate instance shipped with the agent. The steps to install and configure C agent are:

1. Install the software by unpacking to the agent location
2. Start GGSCI from the agent root directory
3. Create sub-directory structure
4. Configure and start the agent manager

At the time of writing this book, Oracle GoldenGate Veridata C agent is not available for download from the Oracle software delivery cloud. Hence, installation and configuration of C agent is not covered. Figure 6-3 shows the list of software available for Oracle GoldenGate Veridata Media Pack for Linux x86-64.

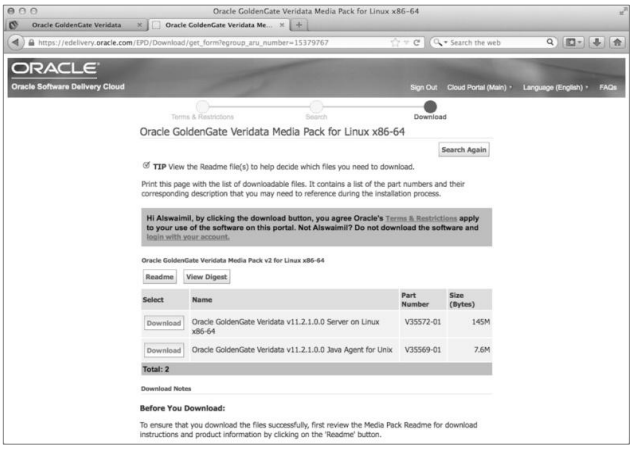


Figure 6-3: Oracle GoldenGate Veridata Media Packs

Java Agent Settings

This agent is written in Java language. It is the new version of the Oracle GoldenGate Veridata agent and does not require Oracle GoldenGate instance. The Java agent uses JDBC to connect to the database. Hence, a Java environment JRE/JSDK is needed. The minimum version of JRE/JSDK is 1.5. Visit http://www.java.com for the Java download archive.

Software Installation

The installation steps apply to Linux and UNIX operating systems. For proper security settings, the agent must be installed using a non-root operating system user with the correct umask setting.

1. Download agent software from the Oracle software delivery cloud.
2. Create the agent root directory. Move the download to this directory.
3. Extract the zipped file using: unzip Linux/UNIX utility.
4. Make a copy of agent.properties.sample and rename it as agent.properties.
5. Configure the parameters listed in Table 6-2.

The following executes step 3 by running unzip Linux/UNIX utility from the command-line.

```
$ unzip V35569-01.zip
Archive:  V35569-01.zip
   creating: agent/
   creating: agent/classes/
   creating: agent/drivers/
   creating: agent/lib/
   creating: agent/licenses/
  inflating: agent/JavaAgent.jar
  inflating: agent/agent.bat
  inflating: agent/agent.properties.sample
  inflating: agent/classes/log4j.properties
  inflating: agent/drivers/gvdb2.jar
  inflating: agent/drivers/gvsybase.jar
  inflating: agent/drivers/ojdbc6.jar
  inflating: agent/drivers/sqljdbc.jar
  inflating: agent/lib/commons-dbcp-1.2.2.jar
  inflating: agent/lib/commons-pool-1.3.jar
  inflating: agent/lib/log4j-1.2.12.jar
  inflating: agent/lib/ucp.jar
  inflating: agent/licenses/Apache-LICENSE-2.0.txt
  inflating: agent/licenses/notices.txt
  inflating: agent/agent.sh
  inflating: ogg-veridata-readme-11.2.1.0.0.doc
  inflating: ogg-veridata-readme-11.2.1.0.0.txt
  inflating: ogg_veridata_relnotes_11.2.1.0.0.doc
  inflating: ogg_veridata_relnotes_11.2.1.0.0.pdf $
```

For Step 4, the configuration file agent.properties.sample is copied to agent.properties and modified as follows:

```
server.port=4000
database.url=jdbc:oracle:thin:@S1E2
server.driversLocation = drivers
server.jdbcDriver=ojdbc6.jar
database.transaction.isolation=READ_COMMITTED
```

Table 6-2 describes the parameters for configuring Java agents. The agent.properties is a file located on the agent root directory. The sample file, agent.properties.sample provides database-specific values. Use the suggested values for the target database.

Parameter	Description
server.port	The server.port property is the port where the Veridata agent listens for connection requests.
database.url	The database connect string that enables the agent to connect to the database using JDBC.
server.driversLocation	The location of the JDBC driver. Enter the location for non-default JDBC software installation.
server.jdbcDriver	The name of the JDBC driver. Agent release 11.2 is compatible with Java 6.
database.transaction.isolation	Specify the transaction level of isolation used during the initial comparison step. For Oracle, it only supports READ_COMMITTED value.

Table 6-2: Java Agent Parameters

Starting and Stopping Java Agents

To start the Java agent, navigate to the agent root directory, export the location of Java runtime environment (JRE) using the JRE_HOME variable, then execute the program agent.sh as shown here:

```
$ export JRE_HOME=/usr/java/jre1.6.0_18
$ ./agent.sh start
$
```

To stop the Java agent, navigate to the agent root directory, export the location of Java runtime environment (JRE) using the JRE_HOME variable, then execute the program agent.sh as shown here:

```
$ export JRE_HOME=/usr/java/jre1.6.0_18
$ ./agent.sh stop
$
```

Java Agent Installation for Microsoft Windows

When the database server is running on Microsoft Windows and the agent to be installed on the same server, then the Oracle GoldenGate Veridata Java agent for Microsoft Windows is required. It is necessary to match the database server operating system architecture with the Oracle GoldenGate Veridata Java agent for Microsoft Windows. The software is available for 32-bit and 64-bit systems. The installation is self-explanatory as it adapts a Graphical User Interface (GUI) wizard-based installer.

Figure 6-4 shows step 5 of 10, which presents the database type and JDBC connection details.

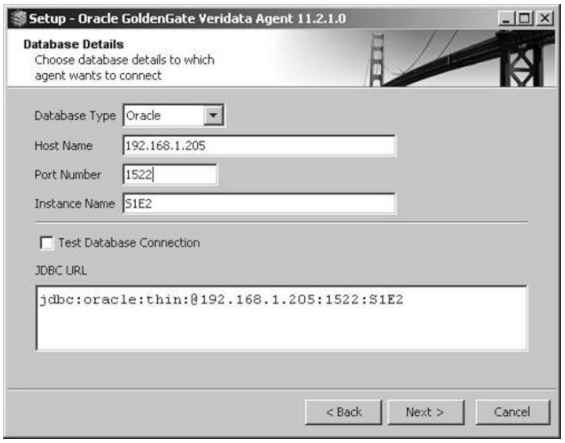


Figure 6-4: Veridata Java Agent Installation (Windows)

Oracle GoldenGate Veridata Setup

Since Oracle GoldenGate requires a database repository for storing configured objects metadata, the initial steps are to create the repository database, create the repository owner, grant database access privileges, and then start installing Oracle GoldenGate Veridata server software. First, it is recommended to install, configure, and start the agent. This enables performing a quick test after successfully installing the Oracle GoldenGate Veridata Server.

Database Repository

Oracle GoldenGate Repository stores details about the source and target database configured components.

Use any of the supported database management systems listed below:

- Oracle Database
- Oracle MySQL
- Microsoft SQL Server

The database repository configured using Oracle Database 11g has the following details on Figure 6-5 of the Oracle Database Configuration Assistant (DBCA).

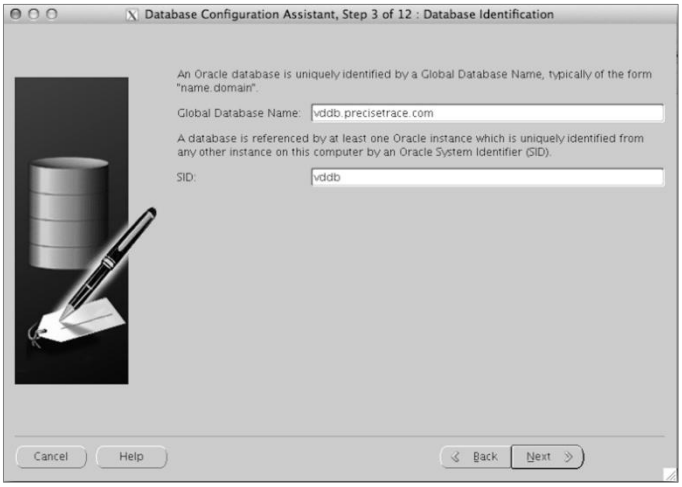


Figure 6-5: Create Database using DBCA

Database Name: vddb
Database Domain: precisetrace.com
Database Instance: vddb

Oracle Corporation recommends using the database automatic memory management (AMM). This enables self-managed memory utilization when running compare jobs and reports. The overall database resource requirements are low and static. Operate the database on archivelog mode to enable online backup, restore, and recover operations.

Repository Database Sizing

Since the database repository is only dedicated for storing configured objects metadata created using Oracle GoldenGate Veridata server web applications, it is classified as a short transactions type where the default database sizing parameters are adequate for the majority of the installations.

However, when the Oracle GoldenGate Veridata Server performs sort operations using the database, the temporary tablespace should be configured to support the largest data set. Database best practices suggest to create the temporary tablespace using groups as illustrated next:

```
SQL> ALTER TABLESPACE temp TABLESPACE GROUP tmp_ts_group;

Tablespace altered.

SQL> CREATE TEMPORARY TABLESPACE temp2
  2  TEMPFILE '/d01/app/oracleSE/oradata/vddb/temp201.dbf' SIZE 1000M
  3  AUTOEXTEND ON;

Tablespace created.

SQL>

SQL> ALTER USER vd_admin TEMPORARY TABLESPACE tmp_ts_group;

User altered.

SQL>
```

Create a dedicated tablespace for the Oracle GoldenGate Veridata repository owner. This promotes highly effective maintenance tasks such as upgrade, backup, etc.

```
SQL> CREATE TABLESPACE vddb_dat
  2  DATAFILE '/u01/app/oracle/oradata/vddb/vddb_dat01.dbf' SIZE 100M
  3  AUTOEXTEND on;

Tablespace created.
```

Now, create Oracle GoldenGate Veridata database role, grant database privileges, create repository owner and the role to the repository owner.

```
SQL> CREATE ROLE veridata_role;

Role created.

SQL> GRANT CREATE SESSION, CREATE TABLE, CREATE VIEW, CREATE PROCEDURE,
  2  CREATE SYNONYM TO veridata_role;

Grant succeeded.

SQL> CREATE USER vd_admin IDENTIFIED BY oracle
  2  DEFAULT TABLESPACE vddb_dat
  3  QUOTA UNLIMITED ON vddb_dat;

User created.

SQL> GRANT veridata_role TO vd_admin;

Grant succeeded.

SQL>
```

Remote Repository Database

When the repository database for the Oracle GoldenGate Veridata Server is created on a remote system, then a database listener and a local database connect string must be configured. The local database connect string is specified during the installation. Figure 6-6 represents the topology when the repository database is remotely located.

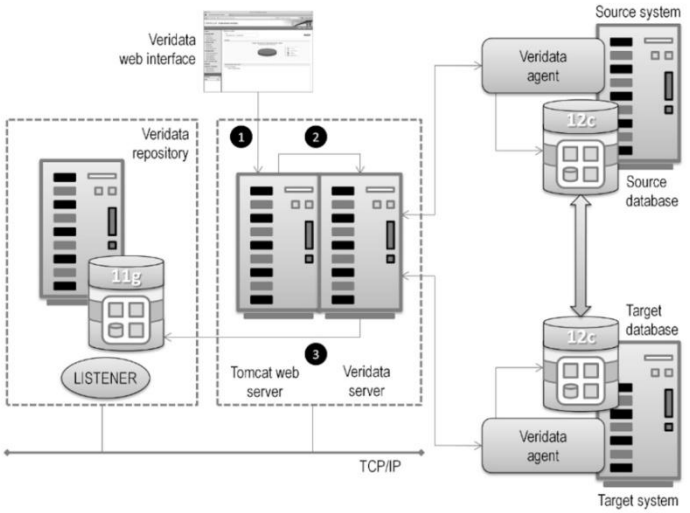


Figure 6-6: Veridata Remote Repository Database Topology

During Oracle GoldenGate Veridata Server installation, there are two options to connect to the repository database:

- Local Database Connect String
- EZCONNECT String

Before starting the installation of the Oracle GoldenGate Veridata Server, verify that the connection is successfully established using the database connect string which provides advanced listener settings.

```
SQL> conn vd_admin/oracle@vddb
Connected.
SQL>
```

Oracle GoldenGate Veridata Server Sizing

Size the Oracle GoldenGate Veridata Server to sustain processing and memory requirements. The agent fetches data from the database, while the Veridata Server

performs sorting and data comparisons. Consider the following when sizing the server:

- CPU Sizing for sort processing location: database or server.
- Memory Sizing for the sort maximum memory usage; default is 50MB.
- Storage Sizing for the temporary storage directory for source and target data.

The major resource consumers, on the Oracle GoldenGate Veridata server, are the row sorting operations of the initial comparison step when the profile sort parameter setting is on server vs. database. Depending on the available server resources, Oracle GoldenGate Veridata supports In-memory and In-disk sorts. The In-Memory sort requires approximately 2.5 the size of the data, whereas the In-disk sort depends on the number of passes (writes) to disk. A One Disk Pass sorts data and writes to the disk only once, a Two Disk Passes sorts data and writes to the disk twice, etc.

Software Installation

Now that the repository database is opened and the listener is started, it is time to begin the Oracle GoldenGate Veridata installation. The wizard-based Java installer interactively guides you through the process. Depending on the options selected, the installer steps vary accordingly. The installation employs an existing database user (VD_ADMIN) and the repository resides on an Oracle database 11g. There are two installation methods:

- Graphical User Interface (GUI) wizard-based installer
- Command-Line Interface (CLI) installer

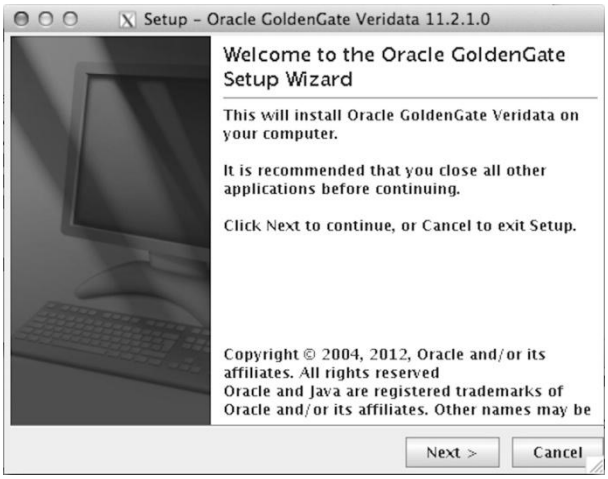
This chapter adapts the Graphical User Interface (GUI) wizard-based installer. The command-line is only recommended in the absence of an X Window system on Linux/UNIX platforms for remote installations. The following installation of Oracle GoldenGate Veridata is remotely executed using an X Window system on Mac OS X.

```
$ unzip V35572-01.zip
Archive:  V35572-01.zip
  inflating: GoldenGate_Veridata_redhatAS40_x64.sh
  inflating: ogg_veridata_relnotes_11.2.1.0.0.pdf
  inflating: ogg_veridata_relnotes_11.2.1.0.0.doc
  inflating: ogg-veridata-readme-11.2.1.0.0.doc
  inflating: ogg-veridata-readme-11.2.1.0.0.txt
$ ./GoldenGate_Veridata_redhatAS40_x64.sh
Unpacking JRE ...
Starting Installer ...
```

Each step of the wizard-based installer is captured, given a title, and briefly described then followed by the launching of Oracle GoldenGate Veridata.

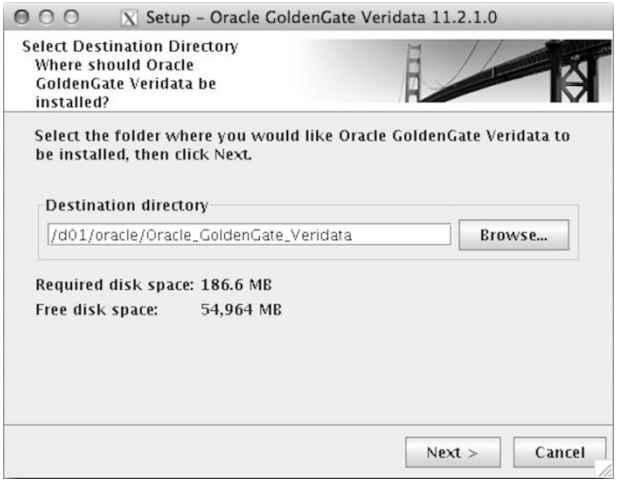
Step 1: Welcome

This step starts the Oracle GoldenGate Veridata programs installation. Click Next to proceed or Cancel to abort the installation. Before proceeding further, closing all currently running applications is recommended.



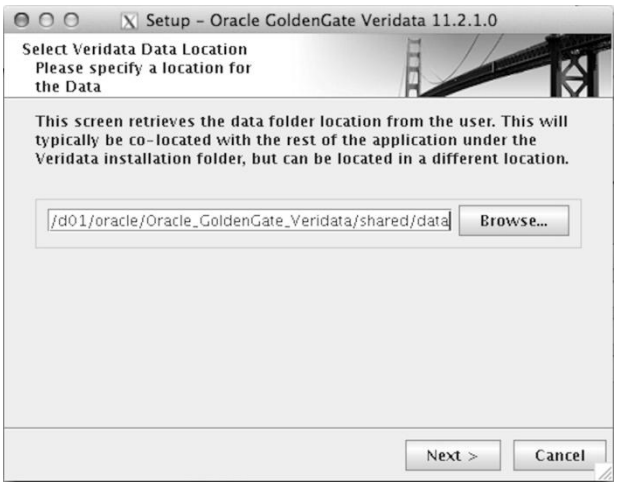
Step 2: Software Location

Specify the software location. For Linux/UNIX security fundamentals, proper directories and files permissions are enforced by setting the shell environment variable umask. Using umask 022 produces files permission setting of 644. Only the owner has read and write permissions; the group and others are read only.



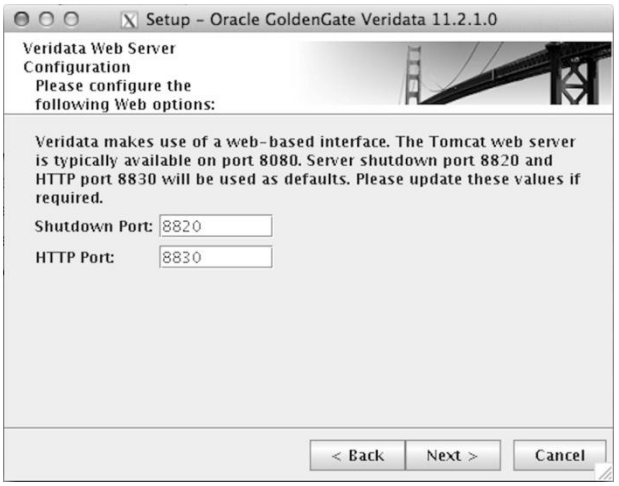
Step 3: Runtime Data Location

Oracle GoldenGate Veridata stores comparison runtime results in files. Out-Of-Sync (OOS) data are stored in two formats, binary and XML. The XML structure describes the comparison pairs. OOS data is further classified by jobs and group names.



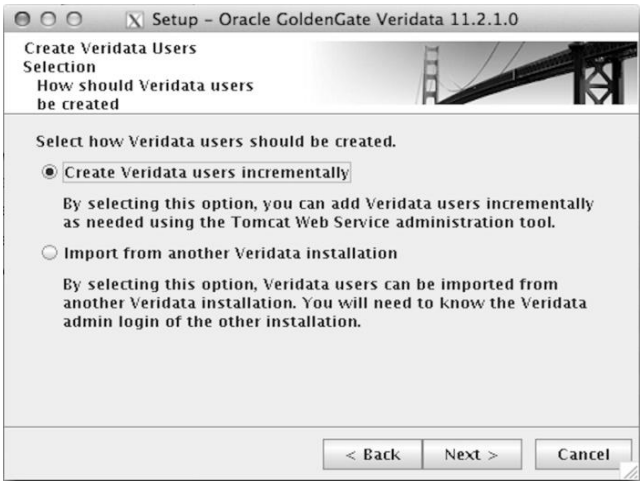
Step 4: Veridata Web Server

Oracle GoldenGate Veridata web server uses Apache Tomcat web server to run the applications for configuring Veridata objects such as connections, groups, comparison pairs, jobs and profiles. Use the URL http://<veridata data IP address>:8830 to access the Veridata web interface.



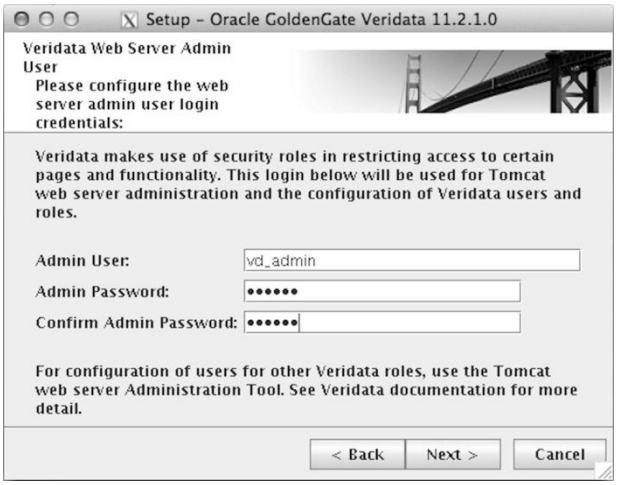
Step 5: Veridata Server User Option

Oracle GoldenGate Veridata users must be authenticated to gain access to the Veridata web server. There are two options for creating users, either manually using Tomcat administration page or by importing Veridata users from another Veridata instance. The installation creates the first user with administration rights.



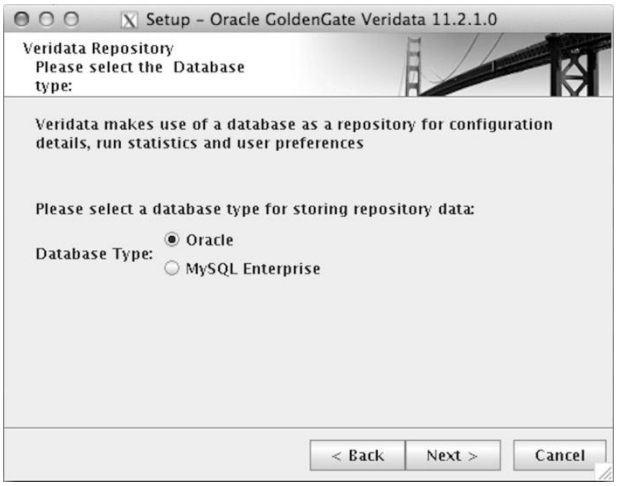
Step 6: Veridata Web Server User

Enter the Veridata web server administrator user name and password. The user vd_admin has administrative rights and is able to create additional users. Use the Tomcat web administration tool for creating groups, users, and assigning privileges.



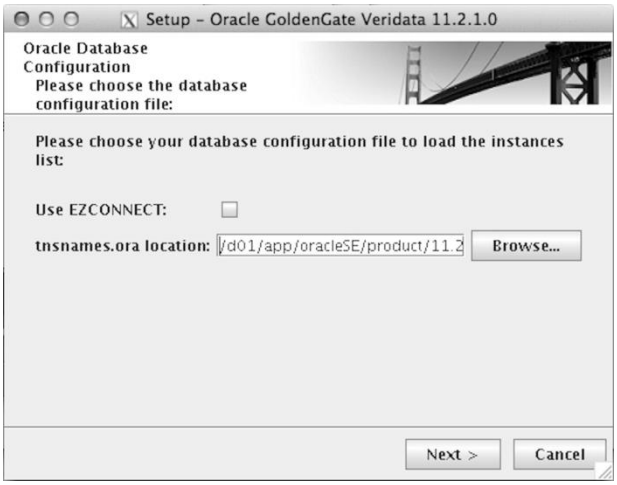
Step 7: Veridata Repository

The repository stores Veridata metadata, runtime parameters (profiles), and Veridata configured objects. Specify the database type. This configuration uses Oracle Database 11g release 2 running on the Oracle Linux server.



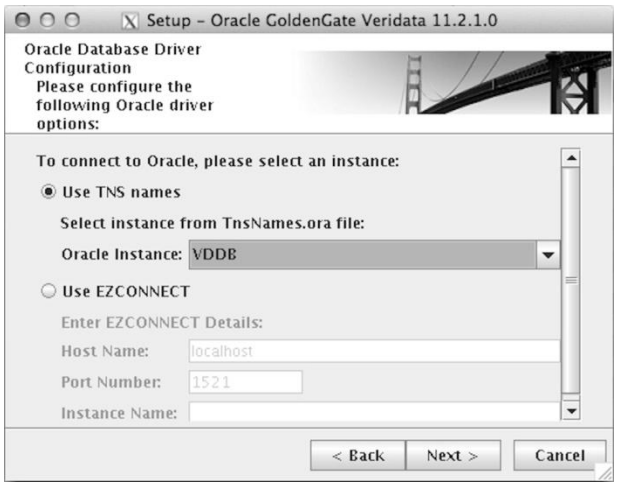
Step 8: Network Files Location

Specify Oracle network files location. If you are uncertain, check the software location for VDDB instance by viewing the /etc/oratab file. Also, verify if the shell environment variable TNS_ADMIN is already exported which overrides the default location. Oracle EZCONNECT is a flexible option for basic configuration.



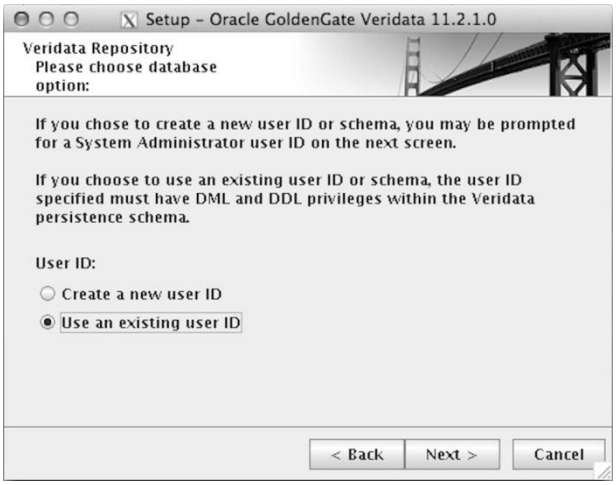
Step 9: Database Connection

The database connection to the repository database has two options: TNS names and EZCONNECT. It is preferred to use the TNS option which requires configuring the local database connect string on the TNSNAMES.ORA network file. Use the NETCA utility to create and test the connection.



Step 10: Veridata Repository

Veridata Repository schema is either an existing database user or newly created during the installation. In either option, this user should be dedicated to Oracle Veridata repository only. Repository maintenance tasks are easier when using a dedicated user.



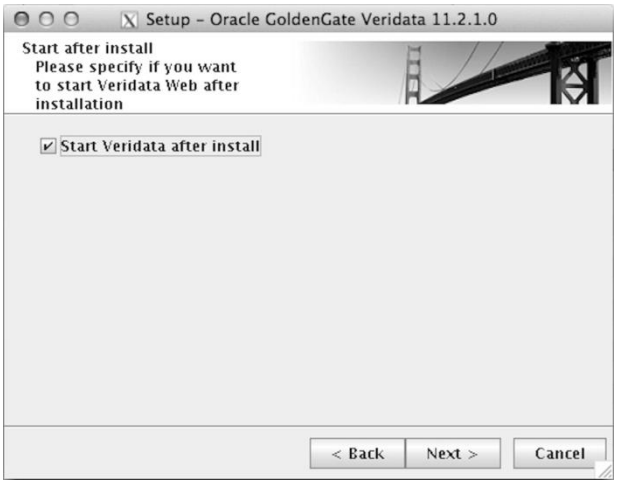
Step 11: Repository Database Schema

Specify the repository database schema (owner) and password. This database schema must be authenticated to access the repository database. This schema assigns the default database profile for password restrictions and memory resource management.



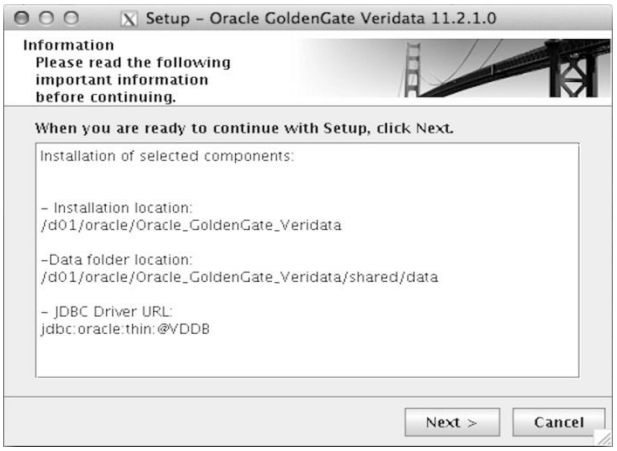
Step 12: Post Installation Tasks

By selecting 'Start Veridata after install', the Oracle GoldenGate Veridata web server is started after successful installation. If left unchecked, use the program veridata_web.sh to manually start and stop Veridata.



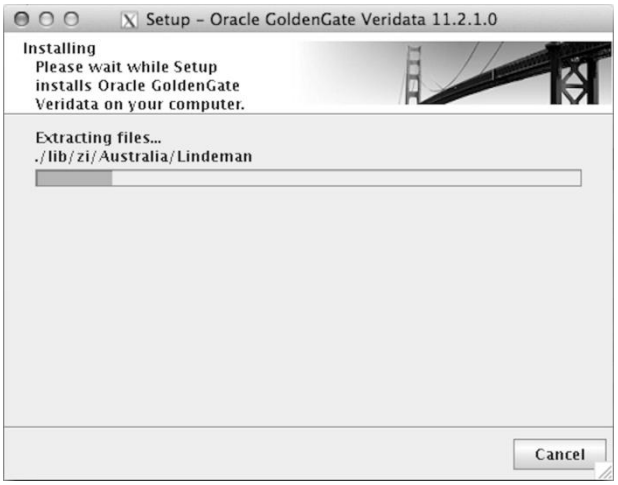
Step 13: Information

Take notes of the information presented on this step. JDBC Driver URL is needed to configure the Java agent. Verify the software location is correct and start the installation by clicking Next. The database connect string is the alias from TNSNAMES.ORA file.



Step 14: Installation Progress

This is a quick installation. It takes approximately 5-10 minutes to complete. It starts by extracting the software, creating the directory structure, and then proceeds with the installation on the specified software location.



Step 15: Successful installation

Click Finish to confirm successful installation. Verify the installation log files installation.log. Search for operating system related issues such as permissions, write access failed, etc. Now, Apache Tomcat web server and Oracle GoldenGate Veridata server are automatically started by the installation program.



Verify Veridata Server Installation

Access the Oracle GoldenGate Veridata server homepage at http://<veridata server IP address>:8830. For authentication, supply the username and password entered on Step 6. Port 8830 is the default HTTP for the first instance.



Data Comparison Steps

To sustain online non-intrusive types of operations, data comparison is divided into two steps. This enables Oracle GoldenGate Veridata to account for data currently in the stream of change data capture and delivery — referred to as in-flight data. For non-key columns, data compare has two options, either the row hash (default) or the actual values of the columns. The row hash value option uses a unique digital signature for calculating the row hash value to produce a compact source and target data. It is highly accurate (but not absolute) to be sent over the network to the Oracle GoldenGate Veridata server for comparison to determine if the two rows have the same or different column values. To make the data comparison absolute, configure the Oracle GoldenGate Veridata server to perform data compare using the actual column-to-column values. The drawback of the column-to-column approach is the increase overhead for processing data compare on tables with a large number of columns; it also increases the network overhead for sending data between the database and Oracle GoldenGate Veridata server.

Figure 6-7 illustrates the comparison steps and sub-steps workflow performed by the Oracle GoldenGate Veridata server to determine out-of-sync (OOS) rows between the source and target tables (compare pairs). The overall efficiency of the comparison is highly dependent on the integration among the different components executing various activities, independently.

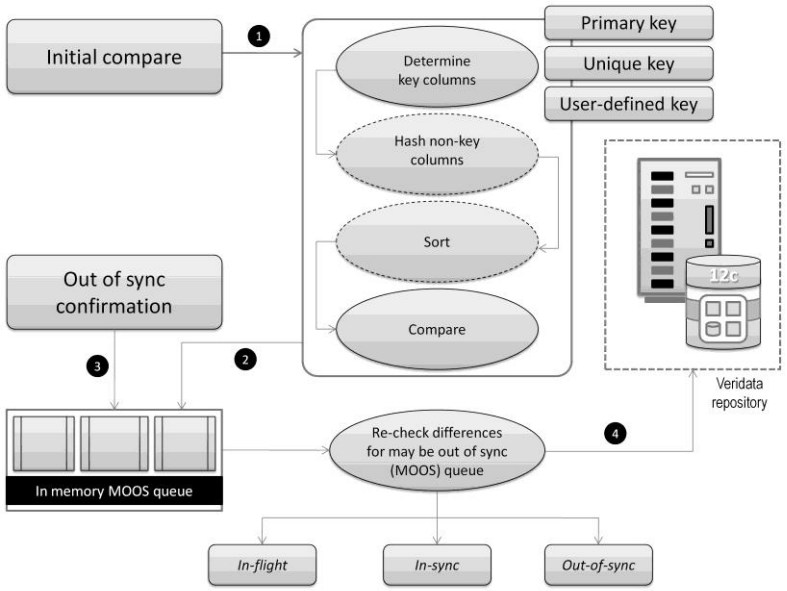


Figure 6-7: Oracle GoldenGate Veridata comparison steps

Unique Key

The Oracle GoldenGate Veridata Server must identify each row uniquely. There are three types of unique key identifiers (UID):

1. Table Primary Key
2. Table Unique Key
3. User-Defined Unique Key

When the table has a primary key, the Oracle GoldenGate Veridata Server uses it for comparing two rows. However, in the absence of a primary key, it uses the table's smallest unique key. When the primary and unique keys are absent, a user-defined unique key is necessary. The user-defined key is created by the column mapping configuration which identifies the columns used as the key.

As shown in Figure 6-7, there are two major steps performed by the Oracle GoldenGate Veridata Server to determine out-of-sync rows. The following explains the two steps in further details.

Step 1: Initial Compare

This step relies on the key to determine the row uniqueness. The compare pair key columns are compared using value-to-value (literally) whereas non-key columns are compared using the hash value (default) evaluated using the row's unique digital signature. If the rows fetch returned by the agent is not sorted, then a sort operation is performed, and the row-to-row data compare operation is carried out. The outcome of the initial compare step is an in-memory, may be out-of-sync (MOOS) queue for accounting in-flight data, their status yet to be confirmed. The hash non-key column is the default. The option of using column-to-column literal value achieves absolute data compare results. The sort task is optional for the Oracle GoldenGate Veridata.

Step 2: Out-of-Sync Confirmation

This step uses the in-memory queue to reconfirm the status of compare pairs rows as in-flight, in-sync, or out-of-sync. The status is highly dependent on latency and transactions volume as data compare is performed while transactional replication is applied to the target database.

- **In-Flight Status**. Due to latency, the row was out-of-sync as determined by the initial step, but it has since been updated. If Oracle GoldenGate Veridata still cannot confirm the row as in-sync, the status is recorded as *in-flight*.

- **In-Sync Status**. The row was out-of-sync after completion of the initial step, but Oracle GoldenGate changed data capture and delivery streams have applied the row to the target database. Oracle GoldenGate Veridata confirmed current row match and the status is recorded as *in-sync*.

- **Out-Of-Sync Status**. The row was recorded out-of-sync by the initial step, but it has not been updated since the initial step took place. Therefore, the status is recorded as permanent *out-of-sync*.

The data compare results metadata is persistently recorded on the Oracle GoldenGate Veridata database repository; but out-of-sync data is recorded on the file system, ready for generating out-of-sync reports.

Tomcat Web Server Administration

Tomcat web server installed under Oracle GoldenGate root directory. Access Tomcat web server administration tool via http://<veridata server IP address>:8830/admin. Use the tool for performing fundamental administration tasks. Figure 6-8 shows the login page. Enter the user name and password submitted during the installation.

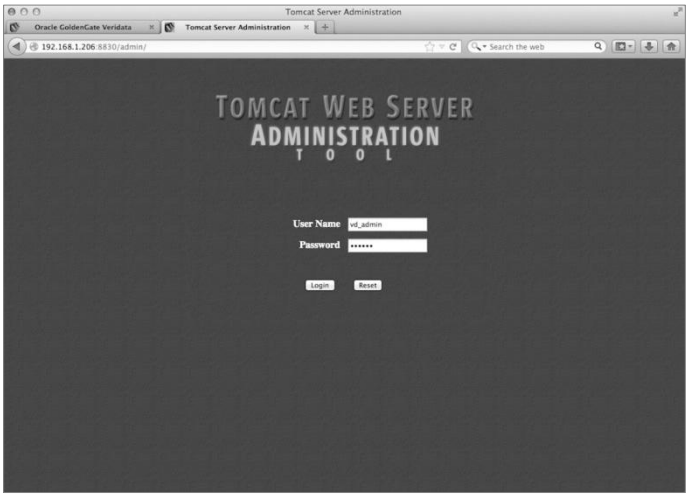


Figure 6-8: Tomcat Web Server

Tomcat Web Server Start-Up and Shutdown

Tomcat web server is an Apache HTTP server. Changes to configuration parameters and maintenance tasks demand to manually restart the Tomcat web server. The following basic commands show how to perform start-up and shutdown.

Tomcat Start-Up

From Oracle GoldenGate Veridata root directory, change directory to /web/bin and execute the script catalina.sh as illustrated next:

```
$./catalina.sh start
```

Tomcat Shutdown

From Oracle GoldenGate Veridata root directory, change directory to /web/bin and execute the script catalina.sh as illustrated next:

```
$./catalina.sh stop
```

User and Group Administration

Oracle GoldenGate Veridata server installation creates the first administrative user, which is capable of creating additional groups and users. Figure 6-9 shows the Tomcat web server administration tool page. Management of groups, users and roles is performed from this web page.

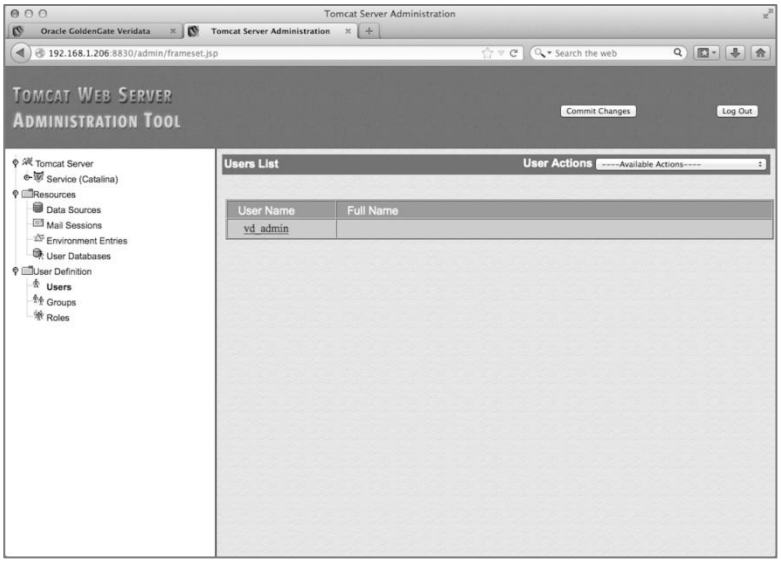


Figure 6-9: Groups, Users and Roles

For security and auditing standard practices, create multiple groups for classifying users then assign users to the relevant groups.

Using Oracle GoldenGate Veridata Web Interface

Oracle GoldenGate Veridata web interface configures objects and rules for performing data comparison activities on compare pairs. Moreover, from Oracle GoldenGate Veridata web interface, several activities are performed such as managing jobs, generating review and out-of-sync summary reports, and handling privileges for users. Figure 6-10 shows Oracle GoldenGate Veridata job, "Run Configuration." A job is linked to one-or-more compare groups and each is executed as a logical unit of work.

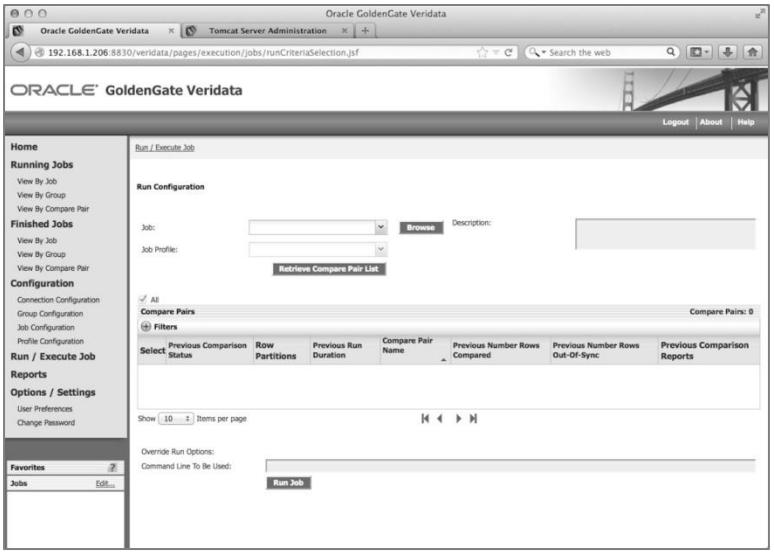


Figure 6-10: Oracle GoldenGate Veridata Web Interface

Stopping and Starting Oracle GoldenGate Veridata Server

For reasons such as altering Web server parameters, a restart of Oracle GoldenGate Veridata is required. From Oracle GoldenGate Veridata root directory, navigate to /web/bin and execute the script veridata_web.sh as illustrated next:

Since Oracle GoldenGate Veridata server already started, execute the shell script veridata_web.sh using the argument "stop," this disables access to the Oracle GoldenGate Veridata web interface.

```
$ ./veridata_web.sh stop
CATALINA_HOME: /u01/app/Oracle_GoldenGate_Veridata/web
JRE_HOME: /u01/app/Oracle_GoldenGate_Veridata/jre
JAVA_OPTS: -Xms1024m -Xmx1024m -Djava.awt.headless=true -
Dveridata.home=/u01/app/Oracle_GoldenGate_Veridata -
```

```
Dveridata.log.dir=/u01/app/Oracle_GoldenGate_Veridata/shared/logs -
Dveridata.log.file=veridataweb.log
Using CATALINA_BASE:   /u01/app/Oracle_GoldenGate_Veridata/web
Using CATALINA_HOME:   /u01/app/Oracle_GoldenGate_Veridata/web
Using CATALINA_TMPDIR: /u01/app/Oracle_GoldenGate_Veridata/web/temp
Using JRE_HOME:        /u01/app/Oracle_GoldenGate_Veridata/jre
$
```

To enable access to the Oracle GoldenGate Veridata web interface, execute the shell script veridata_web.sh using the argument "start."

Configurable Veridata Objects

Regardless of how Oracle GoldenGate Veridata server components are installed and integrated, mandatory and optional objects are configured to enable the Oracle GoldenGate Veridata server to perform data comparison tasks and determine out-of-sync compare pairs. Figure 6-11 represents the associations (data model) among Oracle GoldenGate Veridata components and objects, then explained.

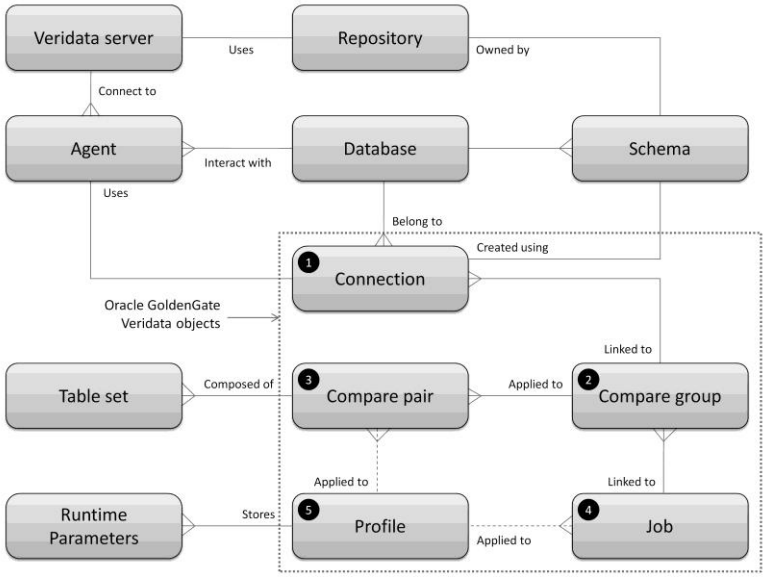


Figure 6-11: The Association among GoldenGate Veridata Objects

An Oracle GoldenGate Veridata Server connects to one or more agents; the agent listening port is required for creating the connections. Each connection is used by an agent to establish one or more database sessions. There can be several connections configured for a database, but this is not necessary. A compare pair belongs to one-and-only-one group (compare group), but a compare group contains one or more compare

pairs — a 2-table set from the source and target database. A job is configured and linked to one or more compare groups to report out-of-sync compare pairs belonging to a compare group. Each compare pair consists of one or more database table sets. A profile is an optional object and in the absence of profiles, the internally defined default profile is applied implicitly. Non-default profiles are applied globally to one or more configured jobs, but also applied to one or more compare pairs to override certain runtime parameters. Non-default profiles inherit their parameters from the default profile. Modifications of profiles are applied dynamically to jobs and compare pairs.

Configure Connections

The Oracle GoldenGate Veridata agent utilizes successfully configured connections to establish database sessions. A single connection is sharable by multiple compare groups. Figure 6-12 shows the list of connections. Select the connection then Edit to modify, or Delete to remove. Click on New to configure a new connection. Testing connections requires the data source database to be available and the agent is started.

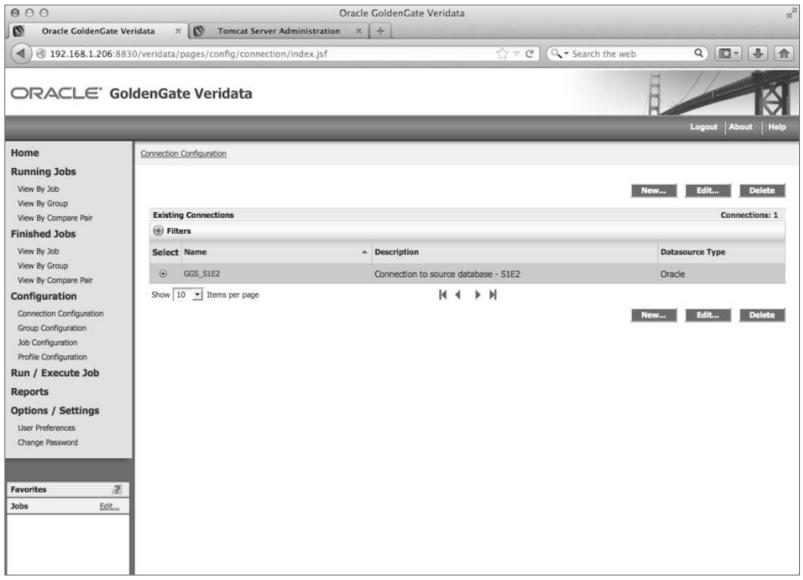


Fig 6-12: Configure Connections

Creating new connections require the following details:

- Agent Host Name or IP Address
- Agent Port
- Data Source Details

From the Oracle GoldenGate Veridata web interface navigation panel, under Configuration, click Connection Configuration. Follow the instructions for configuring the data sources connections.

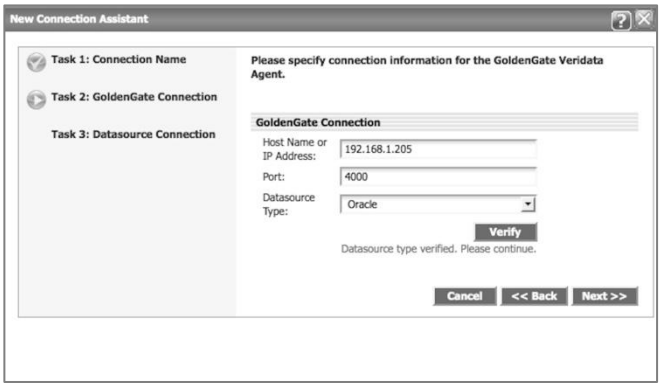
Connection Name

Enter a unique connection name and a short description. For naming convention and consistency, make the database instance name (SID) as part of the connection name. Click Next to enter data source connection details.



GoldenGate Connections

Enter the details for the GoldenGate Veridata agent; the host name or the IP address where the agent is installed, the agent connection port from the agent.properties file, and the database type. Click Verify and make sure the message "Datasource type verified. Please continue." is displayed.



The Oracle GoldenGate Veridata agent may be installed on the database server or elsewhere. The Oracle GoldenGate Veridata Server communicates with the agent using

the connections configured. However, consider network traffic overhead when deciding where the agent should be installed.

Data Source Connection

Provide the data source connection details. Specify a database user with SELECT privilege on compare pairs tables. Click Test Connection. When the message "Datasource connection was successful" is displayed, click Finish.



To reconfigure an existing connection, click the connection name (link), Figure 6-13 shows the connection details, settings, and properties tabs. Make modifications, apply changes, but always verify the agent and test the database connection.

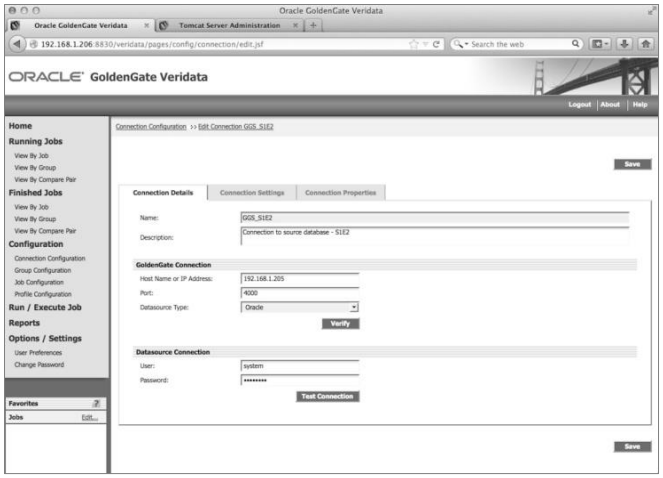


Figure 6-13: Connection Details, Settings, and Properties Tabs

Follow the above steps to configure the connection for the target database T1E2. The target database agent must be running. The agent listening port and the host name/IP address are available to successfully create and test the connection.

Configure Groups

Groups or compare groups are logical containers to establish boundaries among compare pairs. Prior to creating compare groups, it is highly recommended to classify the database tables accessed and manipulated by the applications. The classification of database tables promotes data compares performance, compare pairs maintenance and data repairs. Start by creating the following compare groups:

- Core Tables Compare Groups: Contain set of compare pairs (2 tables) that are accessed and manipulated by transactions as one unit of work. Two rows mismatched on these compare pairs require immediate intervention by the application administrator. For example, the core tables compare groups may be further expanded to include lookup tables.

- Lookup Tables Groups: Contain static data referenced by other tables with minimal DML transactions. In general, it is appropriate to configure one compare group for all lookup tables compare pairs. For example, create multiple lookup table groups to further classify lookup tables groups by applications.

- Non-Core Tables Compare Groups: Contain set of compare pairs (2 tables) that are accessed and manipulated by isolated transactions. For example, create multiple lookup table groups to further classify lookup tables groups by applications or schemas.

For an application with partitioned database schema design, it is technically appropriate to map one compare group for each set of schema compare pairs.
Figure 6-14 depicts examples for configuring groups.

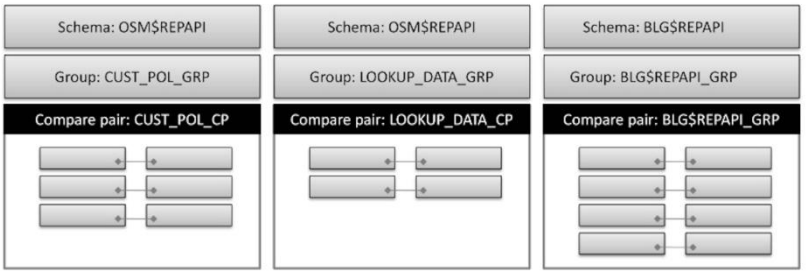


Figure 6-14: Examples of Configuring Compare Groups

Figure 6-15 shows the list of compare groups. Select the group name then Edit to modify or Delete to remove. Click on New to configure a new group (compare group). The next steps create the group CUST_POL_GRP, this compare group contains the compare pairs CUSTOMERS and POLICIES. The compare group needs the source and target connection names.

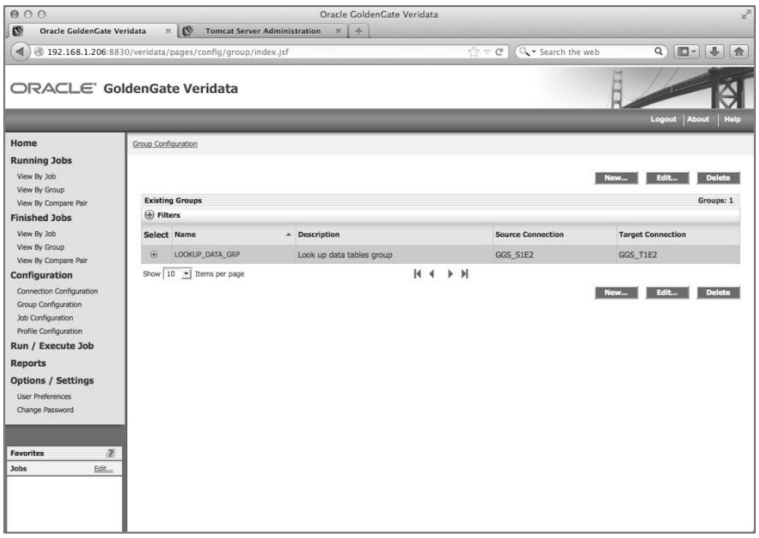


Figure 6-15: List of Compare Groups

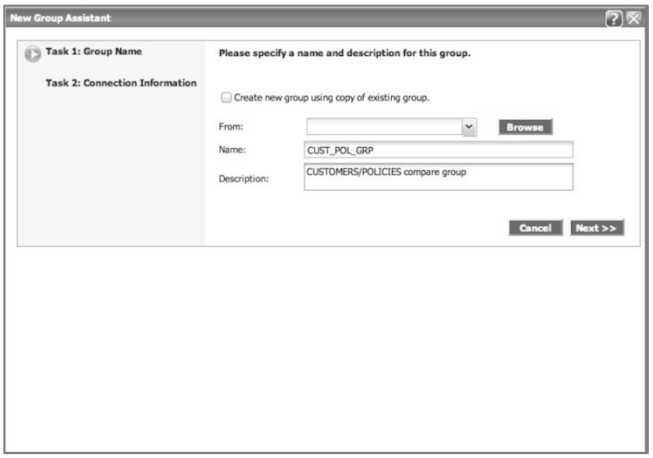
Group Name

Enter a unique short name and a short description for the group. When using the option "Create new group using copy of existing group", select an existing group from the dropdown list of values.

When creating a group, consider the following:

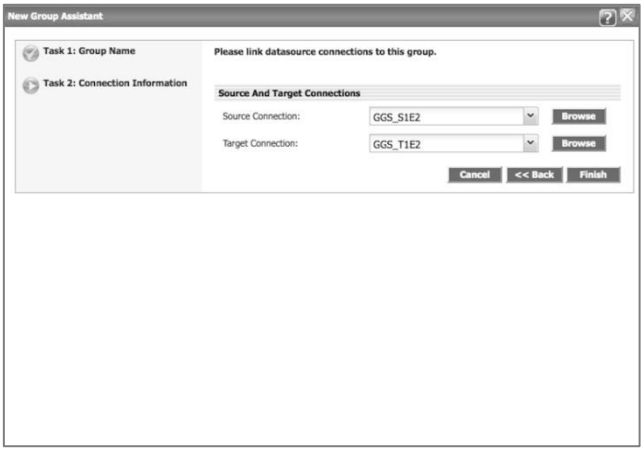
- Since the space character is not permitted for multi-word group names, use characters such as "_" (underline) or "-" (hyphen) to define such group names. A group name such as CUST_POL_GRP is valid.
- Though group names are case insensitive, the use of upper, lower and mixed characters is valid. Follow a consistent naming convention.
- The maximum length for a group name is 200 characters, but the use of short descriptive group names is always preferable.

A unique constraint violation error is returned when entering an existing group name. View the list of groups as shown in Figure 6-15.



Data Source Connection

Enter the data source connections for this group. Carefully specify the source and target connection from the drop list. Click Finish to create the compare group.



Query the table VD_ADMIN.GROUPS, where VD_ADMIN is the repository schema for viewing compare group and the number of compare pairs. Do not update VD_ADMIN.GROUPS from outside of Oracle GoldenGate Veridata web interface.

```
T1E2>COLUMN name FORMAT A30
T1E2>SELECT name, status, num_compare_pairs
  2  FROM   groups;

NAME                           STATUS NUM_COMPARE_PAIRS
------------------------------ ---------- -----------------
```

```
POL_CUST_GRP                              0                 1
CUST_POL_GRP                              0                 2
LOOKUP_DATA                               0                 2
CUST_GRP                                  0                 1
test                                      0                 0
CUSTOMERS_POLICIES_GRP                    0                 2

6 rows selected.

T1E2>
```

Configure Compare Pairs

The next object type configured is the compare pair. Compare pairs are contained within a compare group. Each compare pair is composed of a list of 2-table sets to be compared. Figure 6-16 shows each step and its associated methods/options, if any, for configuring compare pairs. If the compare group (group) is already created, it may have existing compare pairs. There are two mapping methods for adding compare pairs. Every time a compare pair is added, the compare pair must be generated and saved. Edit the column mapping, as necessary (e.g. specifying a user-defined unique key). The last step is to validate columns mapping. Only valid compare pairs are considered for data comparison. In case of an error, investigate, correct the error and re-validate the columns mapping in preparation of running compare pairs jobs for determining out-of-sync rows.

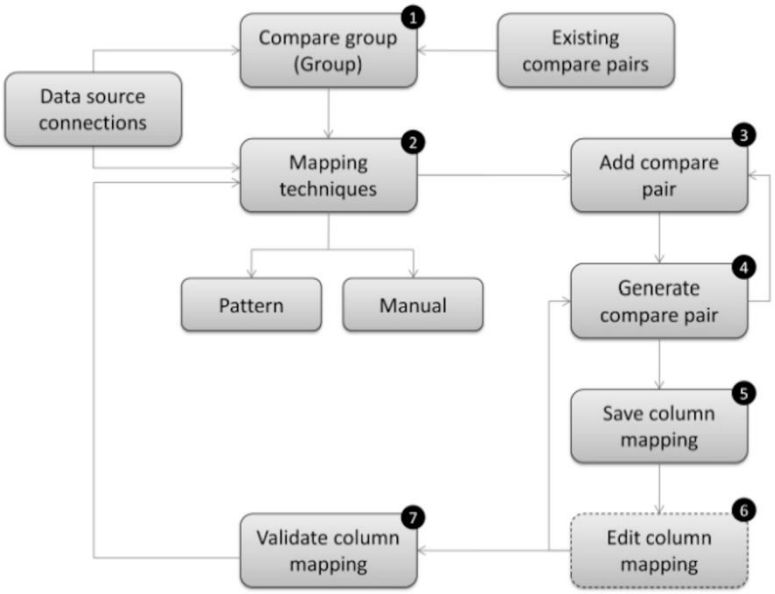


Figure 6-16: Configure Compare Pairs

Oracle GoldenGate Veridata has two techniques for configuring compare pairs.

1. Pattern Mapping
2. Manual Mapping

Pattern Mapping

The pattern mapping technique is suitable for a large set of source and target tables synchronized using Oracle GoldenGate. Without any kind of transformation where the Replicat MAP-TARGET clause has identical table names or wildcard for table names, then the pattern mapping technique is a sound option. Pattern mapping has three methods for creating the compare pairs list.

1. Map Source and Target Tables using exact names
2. Map Source and Target Tables using a SQL % wildcard pattern. Use only one % on each side.
3. Map Source and Target Tables using a GoldenGate * wildcard pattern. Use only one * on each side.

Manual Mapping

The manual mapping is sometimes preferred over the pattern mapping technique (e.g. where the table names are different or for creating a short-listed compare pair from a large list of tables).

The general steps to configure compare pairs are:

1. Identify the compare group name (group)
2. Choose the mapping technique
3. Add the compare pairs
4. Generate the compare pairs
5. Edit column mapping (optional)
6. Save column mapping
7. Validate column mapping

Compare Group Name

By clicking on the compare group, the compare group and connections details presented are followed by the compare pairs section. As shown in Figure 6-17, click on the link "Go to Compare Pair Configuration" to add new compare pairs or modify existing compare pairs for this compare group.

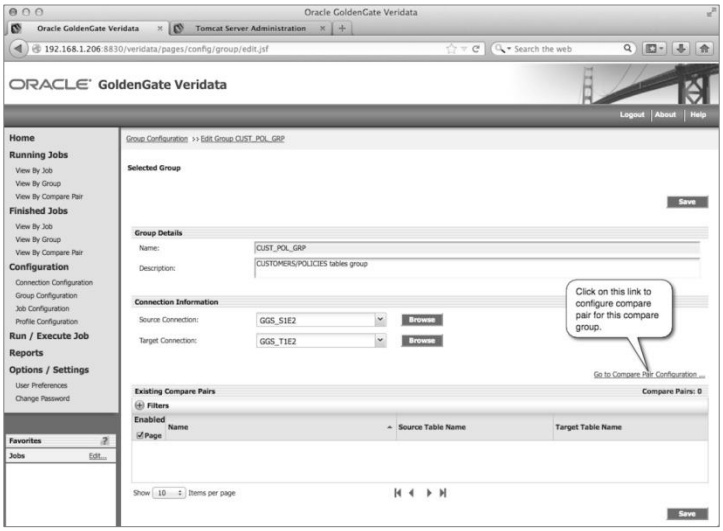


Figure 6-17: Configure A Group's Compare Pairs

Compare Pairs Mapping

For the group CUST_POL_GRP, the manual mapping technique is applied. Figure 6-18 displays the group name information and exiting compare pairs, if any. Click on Manual Mapping tab to select 2-table set from the list of source and target tables.

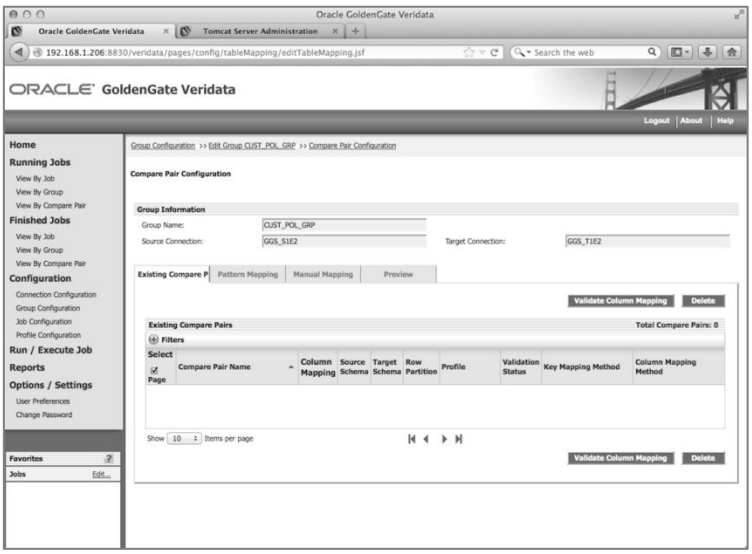


Figure 6-18: Configure Compare Pair Mapping

Adding Compare Pairs

Figure 6-19 shows the manual mapping technique from the data source information. Select the source and target schemes then iteratively build the compare pairs by selecting one table from the source database and mapped table from the target. When completed, click Save.

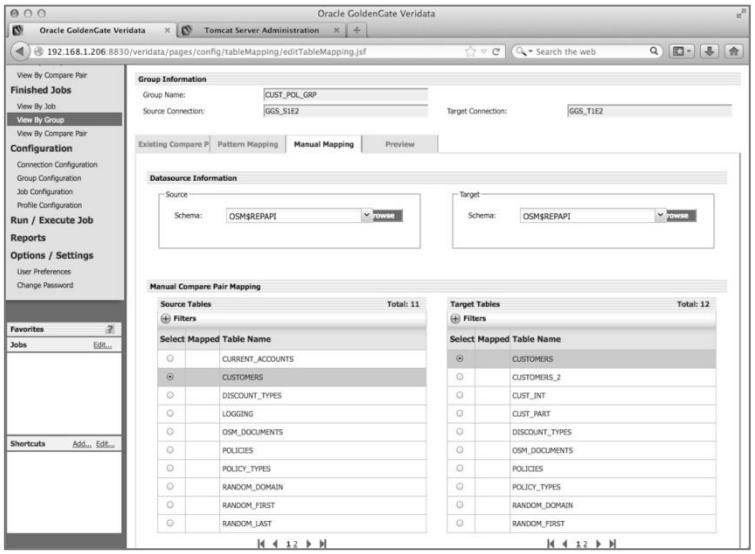


Figure 6-19: Compare Pair Mapping

Generate Compare Pairs

After adding each compare pair, click on Generate Compare Pair located at the bottom of the page. Figure 6-20 shows the added compare pairs for the compare group CUST_POL_GRP. The compare pairs table names are identical.

CUSTOMERS=CUSTOMERS
POLICIES=POLICIES

Applying user-configured profile to compare pair by selecting from the list of profiles, the profile is already created and modified to apply runtime parameters to override the group runtime parameters. Configure row partition to limit comparison to be performed against a subset of the table.

After modifying the compare pair, update the compare pair by clicking on Generate Compare Pair.

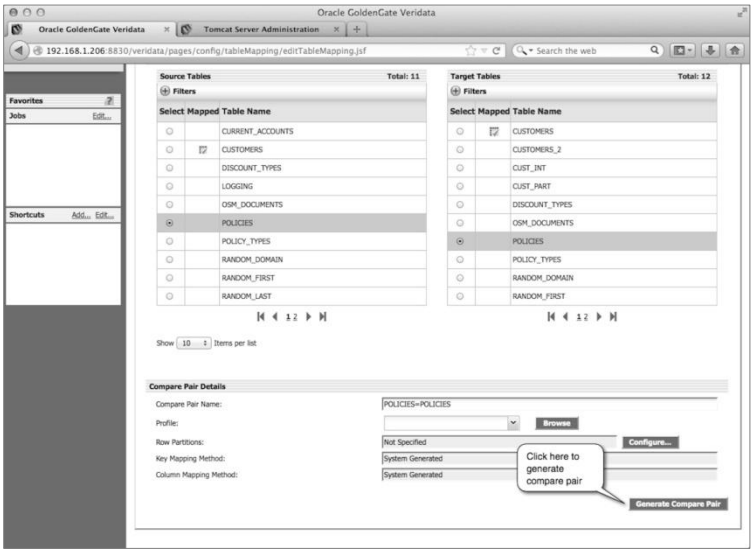


Figure 6-20: Generate Compare Pair

Save Column Mapping

After generating all compare pairs, select the Preview tab and click Save as shown in Figure 6-21.

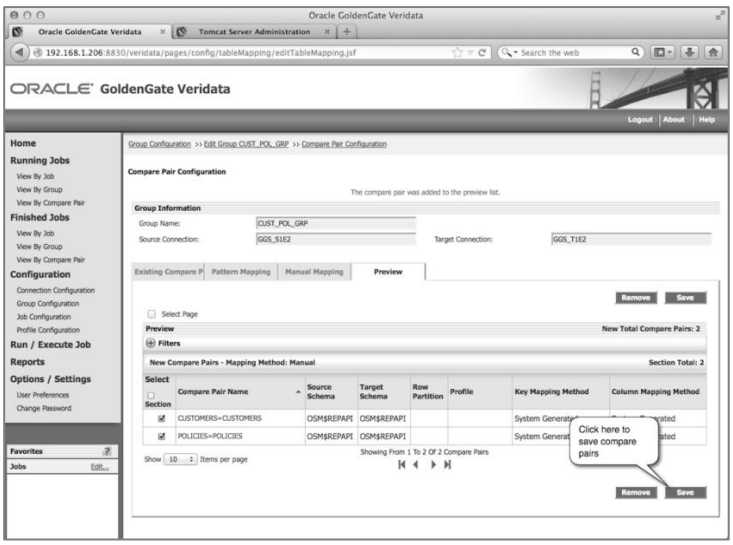


Figure 6-21: Save Compare Pairs

Edit Compare Pairs

The edit compare pair is used to alter key mapping method, comparison column mapping and the mapping type. Click the Edit link as shown in Figure 6-22.

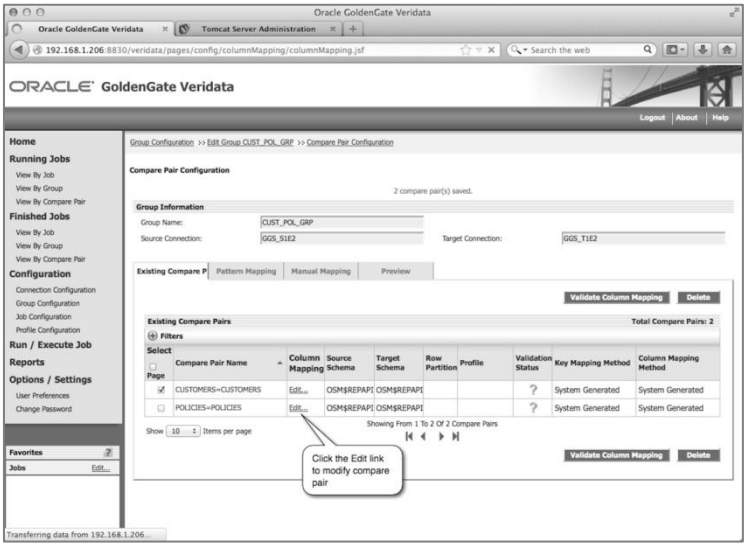


Figure 6-22: Edit Compare Pair Column Mapping

Figure 6-23 alters the mapping type from row hash to row literal. This allows an absolute data compare, but increases processing overhead proportional to the number of columns and their data types.

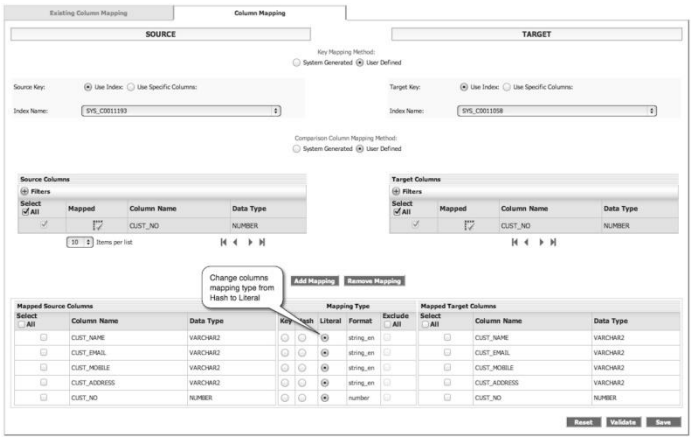


Figure 6-23: Edit Columns Mapping Type from Hash to Literal

Validate Compare Pairs

Whenever a compare pair is added or updated, it must be first saved then validated. Click on Review tab to generate the compare pair by clicking on Generate Compare Pair. Select the compare pairs then click on the Validate Compare Mapping. After the compare pair is successfully validated, the validation status is recorded. Figure 6-24 shows the compare validation step.

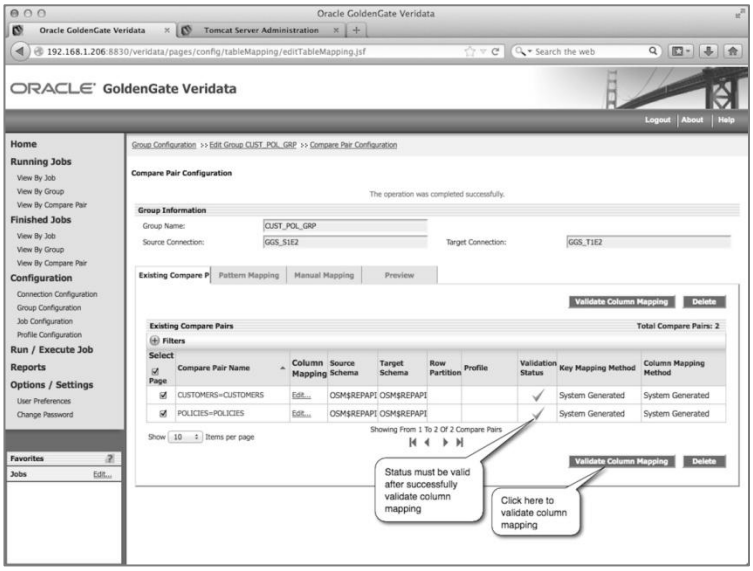


Figure 6-24: Validate Compare Pairs

Configure Jobs

A configured job is linked to one or more compare groups. The tasks to configure a job are:

- Provide the job name and description
- Link the job to one or more compare groups
- Override the compare group connections (optional)
- Specify non-default profile (optional)

From the navigation panel, under Configuration, click on Job Configuration. Figure 6-25 shows existing jobs. Click on New to create a new job. Edit to modify existing job, Delete to remove existing job, or Run to start the job. For more control when running jobs, from

the navigation panel, click on Run/Execute to select the job and modify the row partition by providing the SQL predicate statement.

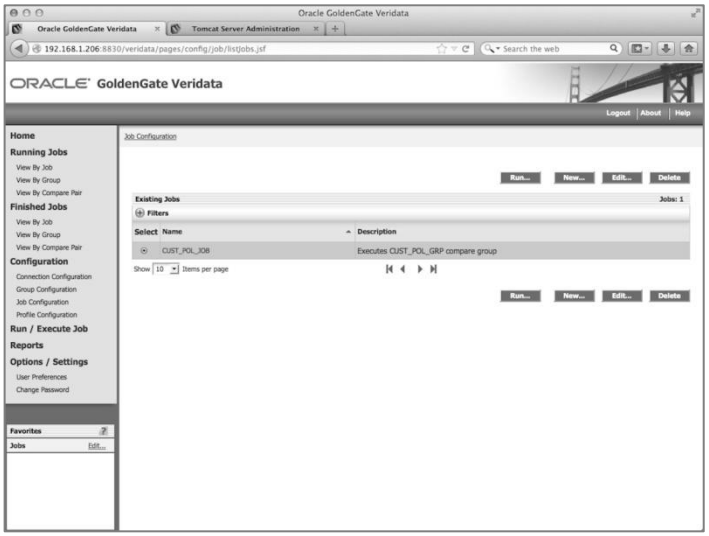
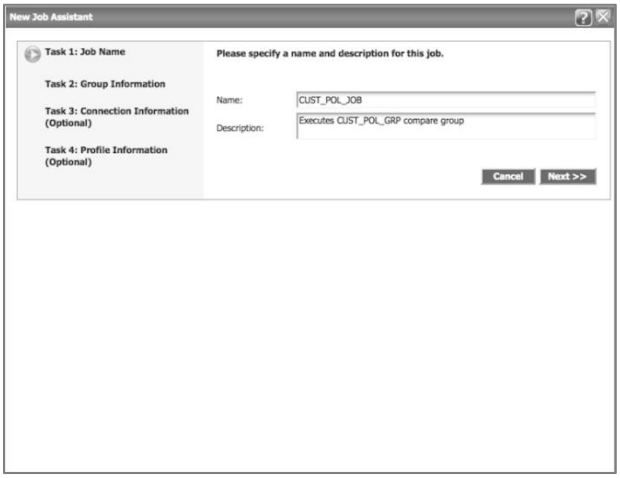


Figure 6-25: List of Jobs

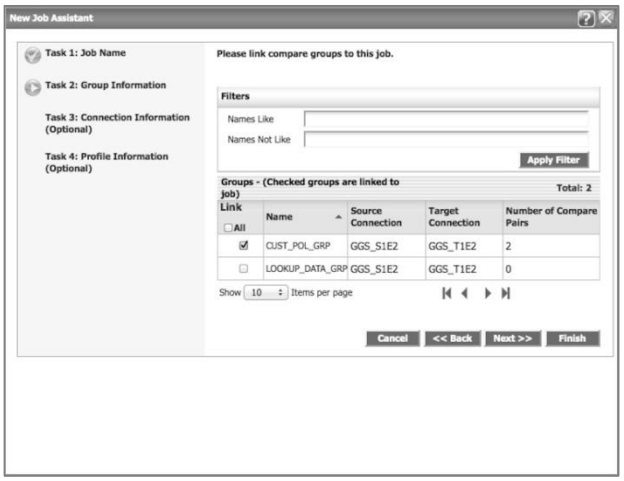
Job Name

Enter the job name and short description. Click Next for connection information.



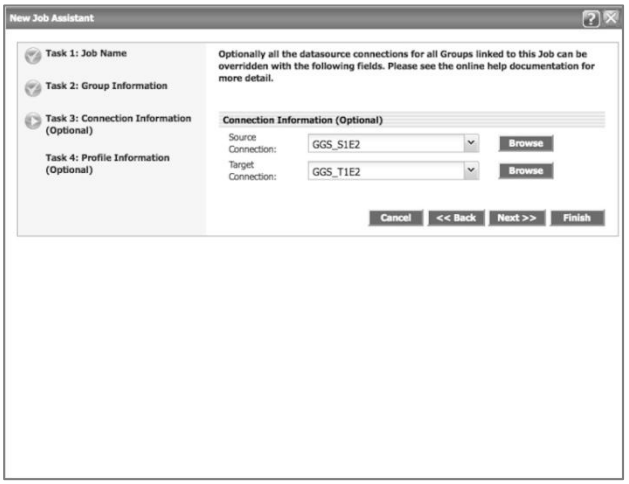
Group Information

A job is associated with one or more compare groups. Check the list of compare groups linked to a job, which executes the compare activities for each compare pair. It is necessary to create multiple jobs for the different compare groups to promote effective resource management incurred during the comparison steps.



Connection Information

The connection information details are optional as they are already part of the compare group configuration. Specifying connection information overrides the compare group connections.



Profile Information

The profile information is optional. Oracle GoldenGate Veridata applies its default profile. When a user-defined profile is specified, it overrides the default profile.



Configure Profiles

Profiles stores runtime parameters. Oracle GoldenGate is shipped with a default profile, named Default. Explicit linking of profiles to jobs and compare pairs is optional, because jobs implicitly inherit runtime comparison parameters from the default profile.

For more control and more effective comparison activities, either modify the default profile parameters or create a new profile which inherits the default parameters setting. Then, assign the non-default profile to jobs and compare pairs.

From the navigation panel, click on Profile Configuration. Click on New to add new profile, Edit to modify existing profile and Delete to remove existing profile. Figure 6-26 shows the profile configuration page.

Profile Settings

The profile has four categories of parameters:

1. General
2. Sorting Method
3. Initial Compare

4. Confirm Out-Of-Sync

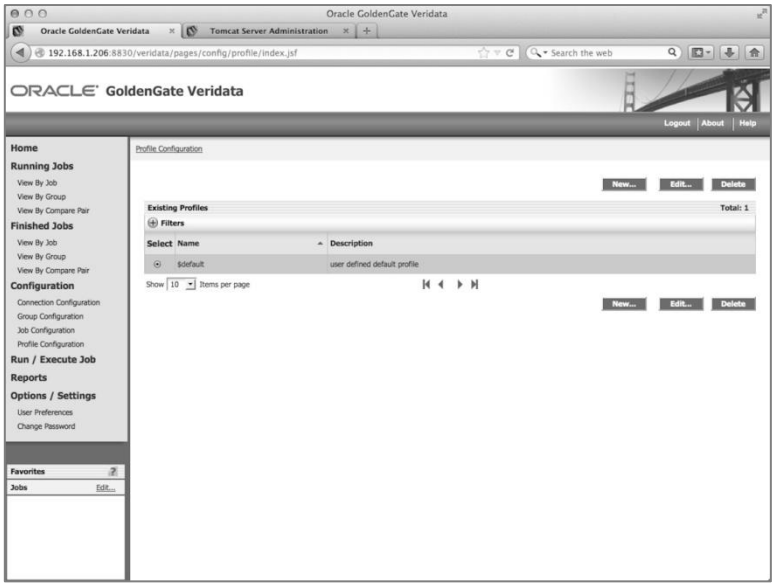
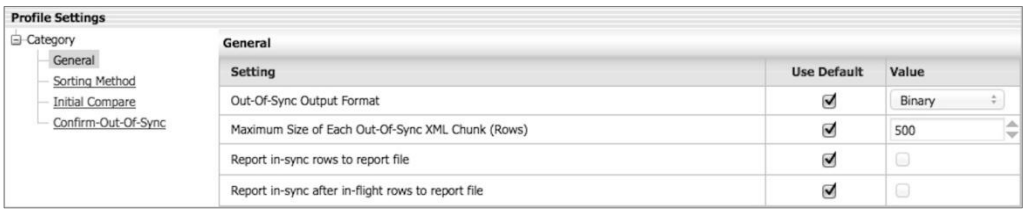


Figure 6-26: Configure Profiles Page

General Parameters

The general parameters default values should remain unchanged. For users interested to view out-of-sync rows in XML format, set the value for the parameter 'Out-Of-Sync Output Format' to XML by performing the following steps:

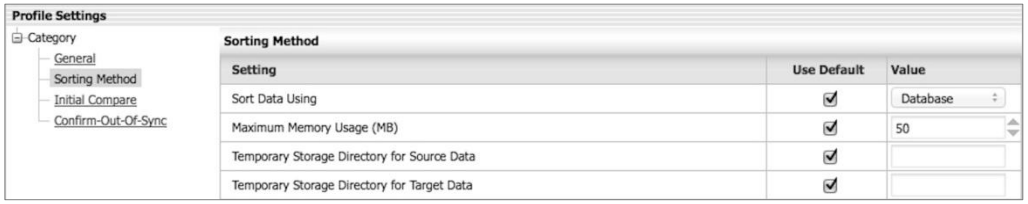
1. Uncheck Use Default
2. Select XML from the list of values
3. Click Save



Out-of-sync files are stored under /shared/data of Oracle GoldenGate Veridata server root directory. Then, sub-directories are created using the name of the compare group.

Sorting Method Parameters

The transactions volume, number of columns and rows of the source and target database tables, has direct impact on setting these parameters. To increase the performance of compare pairs processing, increase the value for the parameter 'Maximum Memory Usage (MB)'. The default setting of 50MB is fairly low for large compare pairs.



Since it is not possible to dynamically change memory usage for each table, use a value that will meet the majority of the compare pairs. Execute the following PL/SQL to report the average table size, which will be used as the value for the parameter.

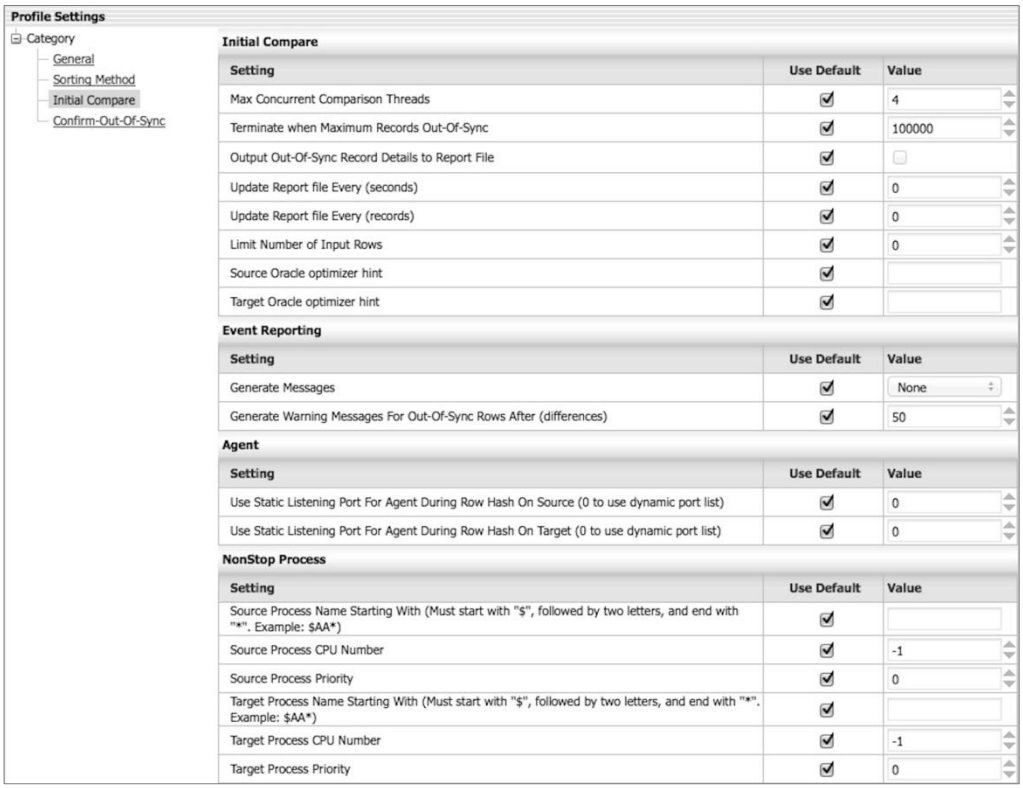
```
SQL> DECLARE
  2    CURSOR c1 IS SELECT table_name, sum(bytes)/1024/1024 bytes
  3                 FROM (SELECT segment_name table_name, bytes
  4                       FROM user_segments
  5                  WHERE  segment_type = 'TABLE' and
  6                         segment_name IN ('POLICIES', 'CUSTOMERS'))
  7                 GROUP BY table_name;
  8    vAverageSize NUMBER:=0;
  9    vtable_name  VARCHAR2(30);
 10    vbytes       NUMBER;
 11    vcnt         NUMBER:=0;
 12  BEGIN
 13    OPEN c1;
 14    LOOP
 15       FETCH c1 into vtable_name, vbytes;
 16       EXIT WHEN c1%NOTFOUND;
 17       vcnt := vcnt+1;
 18       DBMS_OUTPUT.PUT_LINE(vtable_name||' size='||vbytes||' MB');
 19       vAverageSize :=  vAverageSize + vbytes;
 20    END LOOP;
 21    DBMS_OUTPUT.PUT_LINE('The average size='||vAverageSize/vcnt||' MB');
 22  END;
 23  /
POLICIES size=.0625 MB
CUSTOMERS size=.0625 MB
The average size=.0625 MB

PL/SQL procedure successfully completed.

SQL>
```

Initial Compare Parameters

The initial compare parameters are categorized by initial compare, event reporting and agent. Setting the parameter 'Terminate when Maximum Records Out-Of-Sync' to lower values eliminates unnecessary overhead of continued comparison activities.



Confirm Out-Of-Sync Parameters

The out-of-sync profile parameters match the initial compare parameters, typically with lower values settings for parameters related to resources utilization. When changing the below parameters, perform adequate testing outside of Oracle GoldenGate. When the result is satisfactory, apply the values to profile parameters.

- Max Concurrent Comparison Threads
- Limit Number of Input Rows
- Source Oracle optimizer hint/Target Oracle optimizer hint

Execute Jobs

Jobs are executed to determine the compare pairs status within a compare group. From Oracle GoldenGate Veridata web interface navigation panel, click on Run / Execute Job option. The run configuration page has the list of configured jobs. To override the compare group runtime parameter, select the job profile from the available list of configured user-defined profiles. Figure 6-27 shows previous jobs completion status.

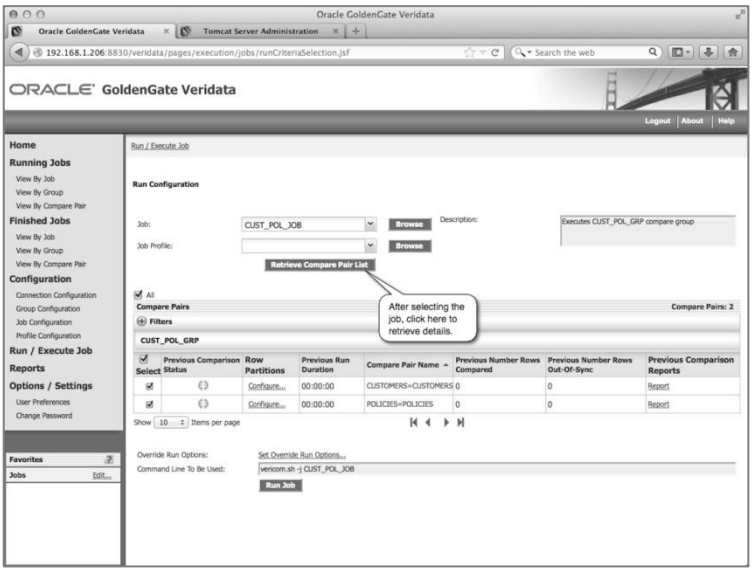


Figure 6-27: Run Job for a Compare Group

Row Partition

Row Partition technique filters the compare pair input data. It narrows down the initial compare by configuring row partition for each compare pair using a valid SQL predicate clause. By default, Oracle GoldenGate Veridata do not apply row partition.

Because the boundary of row partition is within a table only, apply the same row partition clause to all related tables for a consistent out-of-sync report. For example, if we need to limit rows compare to customer for CUST_NO > 100, apply the predicate to CUSTOMERS=CUSTOMERS compare pair and the referenced tables compare pair POLICIES=POLICIES. This yields consistent views of the out-of-sync report. Figure 6-28 shows adding SQL predicate to CUSTOMERS=CUSTOMERS compare pair. Click on New to add new SQL predicate statement. Click on Edit to modify existing SQL predicate statement. Click on Remove to delete existing SQL predicate statement.

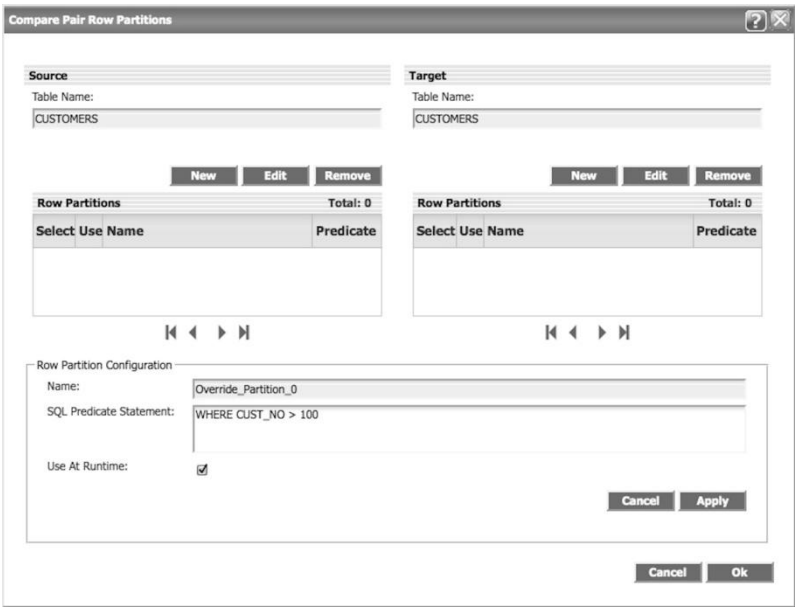


Figure 6-28: Configure Row Partition

Out-Of-Sync Reports

Oracle GoldenGate Veridata stores out-of-sync output data on the directory /shared/data/oos/<compare group name sub-directory> under the Oracle GoldenGate root directory.

The list of out-of-sync reported uses the compare group name.

```
$ ls -al
total 24
drwxr-x---   6 oracle oinstall 4096 Dec 11 23:01 .
drwx------   5 oracle oinstall 4096 Dec  8 18:32 ..
drwxr-x---   6 oracle oinstall 4096 Dec 16 00:31 CUST_JOB
drwxr-x---   4 oracle oinstall 4096 Dec 11 23:01 CUST_POL_GOB1
drwxr-x---  11 oracle oinstall 4096 Dec 10 23:05 CUST_POL_JOB
drwxr-x--- 175 oracle oinstall 4096 Dec 19 18:48 LOOKUP_DATA_JOB
$
```

To view the out-of-sync summary report from Oracle GoldenGate Veridata web interface navigation panel, under Finished Jobs, click View By Compare Pair option. Compare pairs that have out-of-sync has a link on OOS Rows column. Figure 6-29 shows the report output.

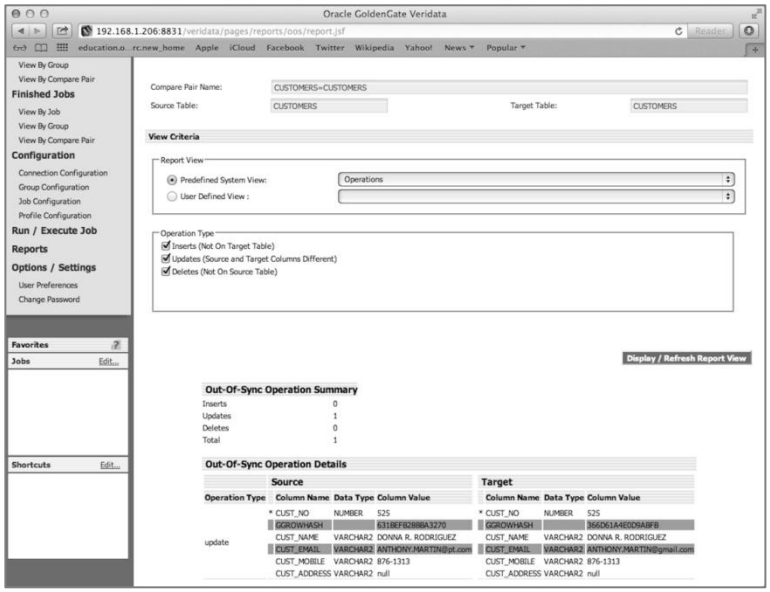


Figure 6-29: Out-Of-Sync Report Summary

The out-of-sync report summary view is customized by report type and operation types. For a tabular summary only, set the Preferred System View to Differences By Columns. The out-of-sync report view is customizable by operation of type:

- Inserts (Not On Target Table)
- Updates (Source and Target Column Different)
- Deletes (Not On Source Table)

The current release of Oracle GoldenGate Veridata functionality is limited to row- to-row compare only of the source and target tables. It does not provide a repair technique for rows differences. The data repair is implemented outside of Oracle GoldenGate Veridata. This chapter discusses Oracle database LogMiner as data repair techniques. Depending on two factors, the degree of out-of-sync compare pair and the causes of out-of-sync, it may be necessary to programmatically develop the repair utility. This requires configuring database links between the source and target system. Also, Oracle GoldenGate Extract and Replicat with intended settings and transformations enables handling out-of-sync from source to target and vice versa.

Oracle GoldenGate Reports

For executed jobs, Oracle GoldenGate Veridata generates various reports for analysis and troubleshooting. From the Oracle GoldenGate Veridata web interface navigation panel, click on Reports. Figure 6-30 shows the Oracle GoldenGate Veridata reports page.

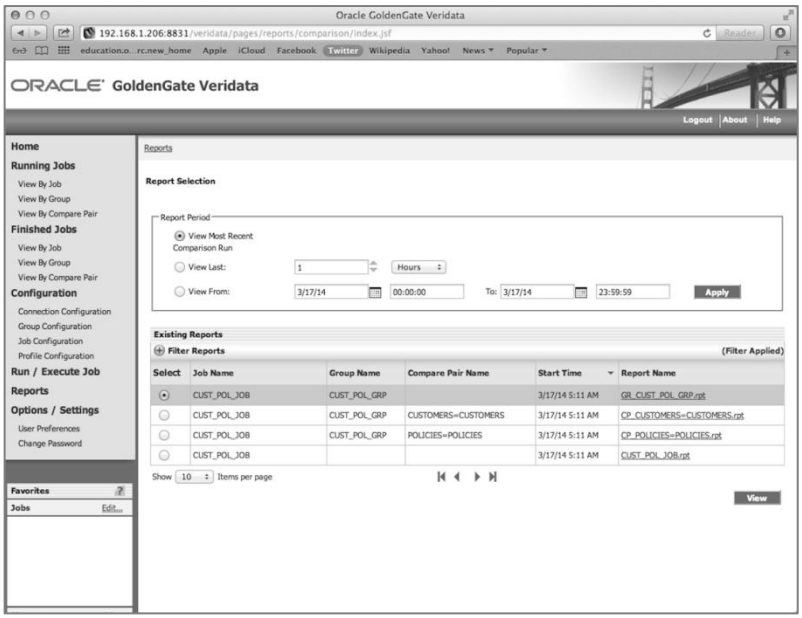


Figure 6-30: Compare Group Reports

Click on Report Name. The report outlines runtime parameters values and a summary for out-of-sync, if any. The report has the details for:

- Profile Applied
- Profile Applied General Parameters
- Sorting Method Parameters
- Initial Compare Parameters
- Confirm Out-Of-Sync Parameters
- Source and Target Agents
- Performance Statistics
- Summaries

Sections of the compare pair DISCOUNT_TYPES=DISCOUNT_TYPES report follows. The report name is constructed in the form <compare group name>.rpt.

****

```
                    Oracle GoldenGate Veridata
            11.2.1.0  OGGVDT_11.2.1.0.0_PLATFORMS_121108.1625

Copyright (C) 2004, 2012, Oracle and/or its affiliates. All rights reserved.

                    Starting at 2013-12-20 10:48:06

************************************************************************

OS          : Linux
Version     : 2.6.18-164.el5xen
Node        : gss-target
Machine     : amd64
Process id  : 27538, Thread id : 38
Java version: 1.6.0_15
Java vendor : Sun Microsystems Inc.

Compare Pair: DISCOUNT_TYPES=DISCOUNT_TYPES
Run ID      : (1235,1,1)

Compare Parameters:

    Source connection: GG_S1E2
    Source host      : 192.168.1.205  (oracle)
    Target connection: GG_T1E2
    Target host      : 192.168.1.206  (oracle)
    Source table     : OSM$REPAPI.DISCOUNT_TYPES
    Target table     : OSM$REPAPI.DISCOUNT_TYPES

Profile: $default

General:
************************************************************************
          Out-Of-Sync Output Format:* xml
   Maximum Size of Each Out-Of-Sync
                   XML Chunk (Rows): 500
 Report in-sync rows to report file: false
     Report in-sync after in-flight
                rows to report file: false

Sorting Method:
************************************************************************
                     Sort Data Using: database
           Maximum Memory Usage (MB): 50
    Temporary Storage Directory for
                         Source Data:
    Temporary Storage Directory for
                         Target Data:

Initial Compare:
************************************************************************
  Max Concurrent Comparison Threads: 4
     Terminate when Maximum Records
                         Out-Of-Sync: 100000
  Output Out-Of-Sync Record Details
                      to Report File: false
 Update Report file Every (seconds): 0
 Update Report file Every (records): 0
         Limit Number of Input Rows: 0
```

```
       Source Oracle optimizer hint:
       Target Oracle optimizer hint:

Initial Compare (Event Reporting):
**************************************************************************
                  Generate Messages: none
      Generate Warning Messages For
            Out-Of-Sync Rows After
                      (differences): 50

Initial Compare (Agent):
**************************************************************************
      Use Static Listening Port For
 Agent During Row Hash On Source (0
          to use dynamic port list): 0
      Use Static Listening Port For
 Agent During Row Hash On Target (0
          to use dynamic port list): 0

…
Confirm-Out-Of-Sync:
**************************************************************************
   Perform Confirm Out-Of-Sync Step: true
      Run Concurrently With Initial
                           Compare: true
      Delay Confirm-Out-Of-Sync By
                         (seconds): 0
     Terminate when Maximum Records
                       Out-Of-Sync: 100000
  Output Out-Of-Sync Record Details
                     to Report File: false
 Update Report file Every (seconds): 0
 Update Report file Every (records): 0
       Source Oracle optimizer hint:
       Target Oracle optimizer hint:

Confirm-Out-Of-Sync (Event Reporting):
**************************************************************************
                  Generate Messages: none
      Generate Warning Messages For
            Out-Of-Sync Rows After
                      (differences): 50

Confirm-Out-Of-Sync (Agent):
**************************************************************************
      Use Static Listening Port For
 Agent During Row Hash On Source (0
          to use dynamic port list): 0
      Use Static Listening Port For
 Agent During Row Hash On Target (0
          to use dynamic port list): 0

Starting  Veriagt session for source rowhash at  2013-12-20 10:48:06.

Initial Compare source Agent Information:
               JDBC URL: jdbc:oracle:thin:@192.168.1.205:1521:S1E2
             Session ID: 848
               SortNull: high
        NLS_CHARACTERSET: AL32UTF8
```

```
Starting  Veriagt session for target rowhash at  2013-12-20 10:48:06.

Initial Compare target Agent Information:
               JDBC URL: jdbc:oracle:thin:@192.168.1.206:1521:t1e2
             Session ID: 816
               SortNull: high
       NLS_CHARACTERSET: AL32UTF8

Performance Statistics for Source Rowhash at 2013-12-20 10:48:07.

    rows                : 3
    duration (secs)     : 00:00:00
    rows/sec            : 200
    row bytes           : 103
    row bytes/sec       : 6866
    bytes/row           : 34
    rows skipped        : 0
    blocks skipped      : 0
    time fetching       : 00:00:00
    time until first row: 00:00:00
    ipc msgs            : 8
    ipc bytes           : 2364
    bytes/msg           : 9

Comparing first row at 2013-12-20 10:48:07

*** Summarizing Initial Row Comparison Step ***

    Time                 : 2013-12-20 10:48:07
    Elapsed Time         : 00:00:00
    Comparisons performed: 3
    Rows per second      : 130
    Rows out-of-sync     : 0
    inserts              : 0
    updates              : 0
    deletes              : 0

Performance Statistics for Target Rowhash at 2013-12-20 10:48:07.

    rows                : 3
    duration (secs)     : 00:00:00
    rows/sec            : 28
    row bytes           : 103
    row bytes/sec       : 980
    bytes/row           : 34
    rows skipped        : 0
    blocks skipped      : 0
    time fetching       : 00:00:00
    time until first row: 00:00:00
    ipc msgs            : 8
    ipc bytes           : 2364
    bytes/msg           : 9



*** Summarizing Confirm Out-of-Sync Step ***
```

```
    Time                 : 2013-12-20 10:48:07
    Elapsed Time         : 00:00:00
    Rows confirmed       : 0
    Rows per second      : 0

*** Summarizing Persistently Out-of-Sync Rows ***

    Rows out-of-sync                 : 0
    inserts                          : 0
    updates                          : 0
    deletes                          : 0

*** Summarizing In-Sync-after-In-Flight Rows ***

    Rows in-sync-after-in-flight     : 0
    inserts                          : 0
    updates                          : 0
    deletes                          : 0

*** Summarizing Still-Changing Rows ***

    Rows in-flight                   : 0
    inserts                          : 0
    updates                          : 0
    deletes                          : 0


tables in sync: source OSM$REPAPI.DISCOUNT_TYPES, target
OSM$REPAPI.DISCOUNT_TYPES

Comparison terminated normally.
```

Veridata Command-Line Utility

While the majority of Oracle GoldenGate Veridata tasks and activities are performed using the web interface, the command-line interface is available for running non-configuration tasks.

Let us start by finding the version for your Veridata command-line. Login to Oracle GoldenGate Veridata server and enter the command below.

```
$ ./vericom.sh -v
VERIDATA_HOME /u01/app/Oracle_GoldenGate_Veridata
JRE_HOME: /u01/app/Oracle_GoldenGate_Veridata/jre
JAVA_OPTS: -Xmx1024m -Djava.awt.headless=true
Oracle GoldenGate Veridata Command Line Interface - Version 11.2.1.0
OGGVDT_11.2.1.0.0_PLATFORMS_121108.1625
$
```

Jobs created using Oracle GoldenGate Veridata web interface are executable from the command-line. Specify the –j flag followed by the job name.

```
$ ./vericom.sh -j LOOKUP_DATA_JOB
```

```
VERIDATA_HOME /u01/app/Oracle_GoldenGate_Veridata
JRE_HOME: /u01/app/Oracle_GoldenGate_Veridata/jre
JAVA_OPTS: -Xmx1024m -Djava.awt.headless=true
Oracle GoldenGate Veridata Command Line Interface - Version 11.2.1.0
OGGVDT_11.2.1.0.0_PLATFORMS_121108.1625
Job LOOKUP_DATA_JOB started
Run ID: (1037,0,0)
Number of Compare Pairs: 2
Number of Compare Pairs With Errors: 0
Number of Compare Pairs With OOS: 0
Number of Compare Pairs With No OOS: 2
Number of Compare Pairs Cancelled: 0
Job Report Filename:
/u01/app/Oracle_GoldenGate_Veridata/shared/data/rpt/LOOKUP_DATA_JOB/00001037/LOO
KUP_DATA_JOB.rpt
Run ID: (1037,0,0)
Job Completion Status: IN SYNC
$
```

Unattended Execution

One of the applications of VERICOM is to enable unattended execution of Oracle Veridata jobs. Data comparison is triggered using the operating system or the database scheduler. Since Oracle Veridata repository is stored on an Oracle database, use DBMS_SCHEDULER to execute Veridata jobs based on pre-determined intervals.

The following steps use DBMS_SCHEDULER to create a database job that executes Oracle GoldenGate Veridata job, LOOKUP_DATA_JOB.

Start by creating the program. This is an optional component, but promotes flexibility when working with DBMS_SCHEDULER. It separates the program from job definition.

```
T1E2>BEGIN
  2         DBMS_SCHEDULER.CREATE_PROGRAM (
  3           program_name=>'COMPARE_PAIRS',
  4           program_type=>'EXECUTABLE',
  5           program_action=>'/home/oracle/lookup_dat.sh',
  6           enabled=>TRUE,
  7           comments=>'Verify lookup tables');
  8  END;
  9  /

PL/SQL procedure successfully completed.

T1E2>
```

The PROGRAM_ACTION parameter specifies the shell script directory and name. The script invokes VERICOM using the flag –j followed by the job name. The shell script contains the two lines below with the correct operating system to execute permission settings.

```
#!/bin/bash
/u01/app/Oracle_GoldenGate_Veridata/vericom.sh -j LOOKUP_DATA_JOB
```

Next, create the schedule which is also optional and can be part of the job definition. Creating a schedule promotes flexibility when working with DBMS_SCHEDULER.

```
T1E2>BEGIN
  2   DBMS_SCHEDULER.CREATE_SCHEDULE (
  3      schedule_name=>'RUN_EVERY_1_HOUR',
  4      repeat_interval=>'FREQ=HOURLY; INTERVAL=1',
  5      comments=>'Run the jobs every hourly');
  6   END;
  7  /

PL/SQL procedure successfully completed.

T1E2>
```

The last step is to create the database job that will execute the Oracle GoldenGate Veridata job.

```
T1E2>BEGIN
  2    DBMS_SCHEDULER.CREATE_JOB (
  3       job_name=>'RUN_VD_COMPARE_LD',
  4       program_name=>'COMPARE_PAIRS',
  5       schedule_name=>'RUN_EVERY_1_HOUR',
  6       comments=>'Execute Veridata compare pair job every 1 hour',
  7       enabled=>TRUE);
  8  END;
  9  /

PL/SQL procedure successfully completed.

T1E2>
```

Manually run and verify that the job is executed without errors. Successful execution returns a status code of 0.

```
T1E2>BEGIN
  2     DBMS_SCHEDULER.RUN_JOB('RUN_VD_COMPARE_LD');
  3  END;
  4  /

 PL/SQL procedure successfully completed.

T1E2>SELECT status, error#, SUBSTR(additional_info, 1, 10) INFO
  2  FROM   dba_scheduler_job_run_details
  3  WHERE  job_name='RUN_VD_COMPARE_LD';


STATUS                         ERROR#     INFO
------------------------------ ---------- ------------------------
SUCCEEDED                               0
```

```
T1E2>
```

Verify the job compare status from within the Oracle Veridata web interface. Figure 6-31 shows the result of the compare pairs is in-sync.

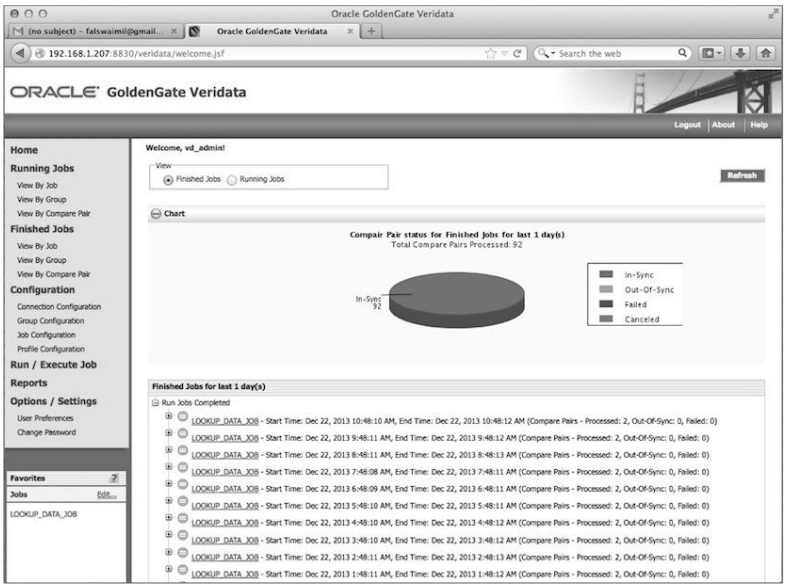


Figure 6-31: Data Compare Results for the Unattended Job

Veridata Server Parameters

Oracle GoldenGate Veridata server runtime uses server-side parameters. Some of the parameters define the instance properties, while others allocate resources for optimizing data compare activities. Oracle GoldenGate Veridata installation creates the parameter file veridata.cfg. The file defines and briefly describes each parameter. To find the location of the parameter file veridata.cfg, navigate to /shared/conf under the Oracle GoldenGate Veridata root directory.

```
# SERVER PARAMETERS
server.veridata_data                    /u01/app/Oracle_GoldenGate_Veridata/
                                        shared/data
server.persistence_db_type              ORACLE_OCI
server.persistence_db_instance          VDDB
server.persistence_db_name
server.persistence_db_port
server.persistence_db_username          vd_admin
server.persistence.oracle.tns_admin     /u01/app/oracle/product/11.2.0/
                                        dbhome_1/network/admin
server.meta_session_handle_timeout      900
```

```
server.max_concurrent_jobs                100
server.max_concurrent_comparison_threads  default
server.max_sort_memory                    default
server.max_comparison_sort_memory         default
server.sort_waiting_threshold             default
server.concurrent.writers                 default
server.concurrent.readers                 default
server.number_sort_threads                default

# WEB PARAMETERS
web.host                                  localhost
web.OOS_User_Templates                    /u01/app/Oracle_GoldenGate_Veridata/
                                          shared/templates
web.persistence_db_url                    jdbc:oracle:thin:@VDDB
```

Data Repair Techniques

Oracle GoldenGate Veridata performs comparisons on data sets to identify data discrepancies. However, data discrepancies repairs are handled outside of Oracle GoldenGate Veridata. Depending on the extent of data discrepancies, identified in the out-of-sync (OOS) report, the repairs techniques are developed using any of the following options:

- Oracle GoldenGate
- Oracle Database Utilities
- In-House Developed Utilities
- Third-Party Utilities

The majority of the data discrepancies are resolved using either Oracle GoldenGate or Oracle Database utilities. However, it is case-specific for selecting and applying the most appropriate options. For example, for consistent pattern of data discrepancies, the use of in-house developed interactive utility is recommended. It is likely to use the out-of-sync (OOS) XML files as input.

When using Oracle GoldenGate to repair data discrepancies related to DELETE and INSERT DML operations, configure Oracle GoldenGate to capture and deliver from the source to target database and vice versa. The purpose of this configuration is for data repair only and not for an online synchronization. This kind of repair should adapt a one-time remote task configuration of the initial-load extract process in which the Extract communicates directly with the Replicat over TCP/IP to apply changes. For data discrepancies related to UPDATE DML operations, configure Oracle GoldenGate to capture and deliver from source to target database and vice versa for an online replication. Consider the following when using Oracle GoldenGate to develop the data repair:

- Limit the Extract process capture using the option SQLPREDICATE of the TABLE clause for capturing the intended row(s) only.
- The Extract-Replicat direction depends on the operation type and target data source.
- Temporary disabling of foreign key constraints may be required for UPDATE DML statements.

Refer to chapter 3 and 4 for more details regarding initial-load techniques and configuring Oracle GoldenGate online synchronization. The Oracle GoldenGate Logdump utility supports data repair techniques. It searches trail files for detailed analysis, advanced filtering, and extracting columns values. Refer to chapter 14 for more details regarding using the Oracle GoldenGate Logdump utility.

The following section will be dedicated to demonstrate the Oracle Database LogMiner utility for developing data repair to handle out-of-sync rows.

Oracle Database LogMiner Utility

LogMiner is a database utility for mining and analysis of the transactions logs stored as database archived log files. The utility independently extracts and rebuilds INSERT and DELETE DML transactions without the need to restore and perform point-in-time database recovery. Historically, LogMiner has been used as the primary tool for repairing logical data corruptions by applying SQL undo (SQL_UNDO) to the target database.

The following sections illustrate the use of Oracle database LogMiner utility as data repair technique by applying SQL_UNDO and SQL_REDO transactions to the target data source.

Create the LogMiner External Data Dictionary

The purpose of the file-based data dictionary is for mapping and translating obscure database objects codes to their corresponding data dictionary names. For example, the table object ID translates to table name and user ID translates to username. The next two steps create the file-based data dictionary.

Step 1: Configure the database parameter UTL_FILE_DEST. This enables PL/SQL write access to the directory specified by the parameter. The parameter is static and requires a database restart.

```
SQL> ALTER SYSTEM SET utl_file_dir='/home/oracle' SCOPE=spfile;

System altered.

SQL> conn sys/oracle_4U as sysdba
```

```
Connected.
SQL> STARTUP FORCE
ORACLE instance started.

Total System Global Area   843456512 bytes
Fixed Size                 2233040 bytes
Variable Size              637537584 bytes
Database Buffers           201326592 bytes
Redo Buffers               2359296 bytes
Database mounted.
Database opened.
SQL>
```

Step 2: Use the DBMS_LOGMNR_D.BUILD procedure to create the file-based data dictionary. The input parameters are the name of the data dictionary file and directory location on the database server as specified by the parameter UTL_FILE_DEST. Ascertain the appropriate write/read operating system file permission settings on the directory specified.

```
SQL> CONN system/oracle_4U
Connected.
SQL>
SQL> BEGIN
  2     DBMS_LOGMNR_D.BUILD (
  3         dictionary_filename => 'T1E2lmdict.ora',
  4         dictionary_location => '/home/oracle');
  5  END;
  6  /

PL/SQL procedure successfully completed.

SQL>
```

Build the LogMiner Archived Files Queue

Now, build the list of archived log files using DBMS_LOGMNR.ADD_LOGFILE procedure. Use the option DBMS_LOGMNR.NEW to create the list and add the first archived log file to the queue. Then, use the DBMS_LOGMNR.ADDFILE for the parameter option to add subsequent archived files (transactions logs).

```
SQL> BEGIN
  2     DBMS_LOGMNR.ADD_LOGFILE (
  3          options=>DBMS_LOGMNR.NEW,
  4          logfilename =>
'/d01/app/oracleSE/fast_recovery_area/T1E2/archivelog/2014_02_15/o1_mf_1_114_9hx
p1rny_.arc');
  5  END;
  6  /

PL/SQL procedure successfully completed.

SQL> BEGIN
  2     DBMS_LOGMNR.ADD_LOGFILE (
```

```
  3          options=>DBMS_LOGMNR.ADDFILE,
  4          logfilename =>
'/d01/app/oracleSE/fast_recovery_area/T1E2/archivelog/2014_02_15/o1_mf_1_115_9hz
k9do8_.arc');
  5  END;
  6  /

PL/SQL procedure successfully completed.

SQL>
```

Start the LogMiner Engine

Start the LogMiner session using DBMS_LOGMNR.START_LOGMNR procedure. The input of the procedure is the file-based data dictionary. This session reads from the file-based data dictionary and writes to v$logmnr_contents dynamic view.

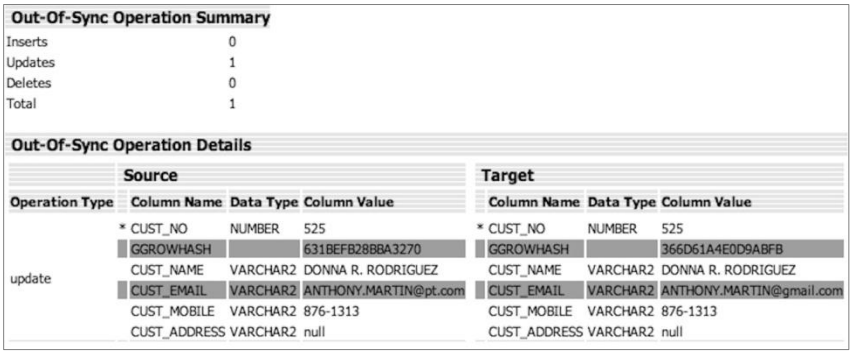
```
SQL> BEGIN
  2     DBMS_LOGMNR.START_LOGMNR (
  3         dictfilename => '/home/oracle/T1E2lmdict.ora');
  4  END;
  5  /

PL/SQL procedure successfully completed.

SQL>
```

Retrieve SQL_UNDO and SQL_REDO Transactions

From the same terminal session while the LogMiner session is active, query V$LOGMNR_CONTENT dynamic view to obtain SQL_UNDO and SQL_REDO DML transactions. Use Oracle GoldenGate Out-Of-Sync report details to narrow down the query output.



Out-Of-Sync Operation Summary

Inserts	0
Updates	1
Deletes	0
Total	1

Out-Of-Sync Operation Details

	Source			Target		
Operation Type	Column Name	Data Type	Column Value	Column Name	Data Type	Column Value
update	* CUST_NO	NUMBER	525	* CUST_NO	NUMBER	525
	GGROWHASH		631BEFB28BBA3270	GGROWHASH		366D61A4E0D9ABFB
	CUST_NAME	VARCHAR2	DONNA R. RODRIGUEZ	CUST_NAME	VARCHAR2	DONNA R. RODRIGUEZ
	CUST_EMAIL	VARCHAR2	ANTHONY.MARTIN@pt.com	CUST_EMAIL	VARCHAR2	ANTHONY.MARTIN@gmail.com
	CUST_MOBILE	VARCHAR2	876-1313	CUST_MOBILE	VARCHAR2	876-1313
	CUST_ADDRESS	VARCHAR2	null	CUST_ADDRESS	VARCHAR2	null

The query uses the CUST_EMAIL value in the WHERE clause to retrieve the SQL_UNDO (DELETE) and SQL_REDO (INSERT) statements.

```
SQL> SELECT sql_redo, sql_undo
  2  FROM   v$logmnr_contents
  3  WHERE  sql_undo like ' %DONNA R. RODRIGUEZ%';


SQL_REDO
--------------------------------------------------------------------------------
SQL_UNDO
--------------------------------------------------------------------------------
insert into "OSM$REPAPI"."CUSTOMERS"("CUST_NO","CUST_NAME","CUST_EMAIL","CUST_MO
BILE","CUST_ADDRESS") values ('525','DONNA R. RODRIGUEZ','ANTHONY.MARTIN@pt.com'
,'876-1313',NULL);
delete from "OSM$REPAPI"."CUSTOMERS" where "CUST_NO" = '525' and "CUST_NAME" = '
DONNA R. RODRIGUEZ' and "CUST_EMAIL" = 'ANTHONY.MARTIN@pt.com' and "CUST_MOBILE"
 = '876-1313' and "CUST_ADDRESS" IS NULL and ROWID = 'AAASepAAEAAAAFoABy';

SQL>
```

A possible solution for data repair is to re-apply the transaction by performing SQL_UNDO followed by SQL_REDO on S1E2 (source) database as illustrated next:

```
SQL> delete from "OSM$REPAPI"."CUSTOMERS" where "CUST_NO" = '525' and
"CUST_NAME" = 'DONNA R. RODRIGUEZ' and "CUST_EMAIL" = 'ANTHONY.MARTIN@gmail.com'
and
"CUST_MOBILE" = '876-1313' and "CUST_ADDRESS" IS NULL and ROWID =
'AAASepAAEAAAAFoABy'
/

1 row deleted.

SQL> commit;

Commit complete.

SQL> insert into
"OSM$REPAPI"."CUSTOMERS"("CUST_NO","CUST_NAME","CUST_EMAIL","CUST_MOBILE","CUST_
ADDRESS") values
('525','DONNA R. RODRIGUEZ','ANTHONY.MARTIN@gmail.com','876-1313',NULL)
/

1 row created.

SQL> commit;

Commit complete.

SQL>
```

End the LogMiner Session

Terminate LogMiner session by executing DBMS_LOGMNR.END_LOGMNR procedure. This stops the LogMiner internal engine.

```
SQL> BEGIN
  2     DBMS_LOGMNR.END_LOGMNR;
```

```
  3  END;
  4  /

PL/SQL procedure successfully completed.

SQL>
```

Figure 6-32 shows the compare pair is in-sync after applying the DML (SQL_UNDO and SQL_REDO) from the LogMiner session.

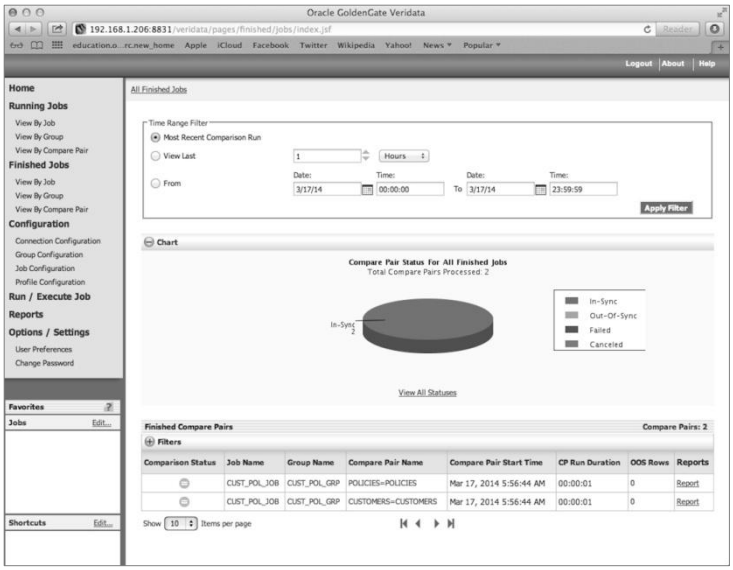


Figure 6-32: Compare Pair In-Sync

Visit http://www.learn-ogg.com for details, demonstrations, and downloads regarding data repair utilities and techniques.

Summary

Business continuity relies on data accuracy. It is crucial to discover, as early as possible, any kind of data discrepancies when using Oracle GoldenGate for high availability, disaster recovery and near zero-downtime migration, and upgrade configurations. In general, a bi-directional topology needs an online comparison tool to detect out-of-sync data sets so that a repair solution is developed and adapted quickly.

This chapter has covered the details of implementing Oracle GoldenGate Veridata for data comparison and introduced one of the database data repair utilities, the Oracle LogMiner utility.

This implementation requires performing the following activities:

- Configure Veridata agents for data fetching from source and target database
- Configure Oracle GoldenGate Veridata Database Repository
- Install Oracle GoldenGate Veridata agents and server components
- Perform Tomcat web server fundamental tasks
- Configure Veridata components, connections, groups, jobs, and profiles
- Run compare jobs and view out-of-sync (OOS) tables
- Configure and execute Oracle LogMiner for data repair

The use of integrated systems requires using certified components only. Oracle GoldenGate Veridata server and agents must be certified on the host operating system and database. Visit http://www.learn-ogg.com for current certifications.

Since Oracle GoldenGate Veridata does not carry out-of-sync data repair, the chapter covered Oracle database LogMiner as data repair techniques to complement Veridata. Given out-of-sync data is available and well organized in XML format, in-house developed utilities should be considered to handle consistent out-of-sync patterns.

The next chapter is a core function of Oracle GoldenGate. It is the near zero-downtime migration and upgrade technique. Oracle GoldenGate Veridata plays a crucial role in verifying no data discrepancies. It verifies the whole data sets of the two databases are identical before the database administrator disconnects users from the current system and moves them to the target system.

Near Zero-Downtime Database Migration and Upgrade

APPENDIX D

Overview

The need to continue processing transactions while performing major tasks on the 24/7 production database server is becoming an obligation for Oracle installations running core business applications and mission critical systems. The objective of this chapter is to cover the techniques of using Oracle GoldenGate and Oracle Database technologies to perform an end-to-end database migration and upgrade with a flexible failback option — leaving no impact on business continuity to achieve near zero-downtime of database availability.

For mission critical systems, an Oracle Database upgrade is strategic. The upgrade project plan must be developed in accordance to Oracle's software policy life cycle. Database administrators must be aware of Oracle policy milestones and plan their upgrade project, accordingly. This keeps their database releases current with Oracle supported platforms for applying fixes and avoiding the addition of premium support services fees. Nevertheless, the upgraded database takes advantage of Oracle software new release features and innovations which transparently enhances the database's functionality, performance, and security.

The near zero-downtime migration and upgrade activities are divided into a series of related phases. Each of the phases is composed of (7) tasks for completion. The list of phases and their brief description are:

1. Deploy Oracle GoldenGate
2. Online Clone of the Production Database
3. Migrate to Automatic Storage Management
4. Upgrade to Oracle Database 12c
5. Applications Regressions Testing
6. Synchronize the Target Database
7. Data Verification

Though a merge of Phase 2 and 3 is possible, this chapter illustrates them as two separate phases. The order of Phase 3 and 4 is exchangeable. This chapter adapts the order listed above — performing migration first then upgrade of the database. However, Phase 5 is application specific and, but discussed briefly.

Deploy Oracle GoldenGate

Oracle GoldenGate is deployed as unidirectional topology — data flows from the source system (production) to the target system. The primary extracts create local trail files. The secondary extracts (data pumps) ship the transactions to the target system. No transformations are performed on either extracts. The manager process on the source system is configured to optimize the storage level by purging local trail files upon being successfully shipped to the target system. The remote trail files are maintained on the target system until the database upgrade is completed and ready to synchronize the target database — aiming for a sub-second latency in preparation of the application transition phase.

Online Clone of the Production Database

Cloning the database is the key for a near zero-downtime database upgrade project. It serves two purposes:

1. As an initial-load technique from an active database
2. Eliminates the various types of downtimes

The database accumulative downtime is affected by two time factors: the timeframe of the database upgrade assistant (DBUA) to successfully upgrade the database and the timeframe of the applications' functional and technical testing. By cloning the database, it avoids the DBUA shutting down the production database required during the database dictionary upgrade. At the same time, it develops a highly reliable and compatible platform for applications testing. The clone is performed using the Oracle Recovery Manager (RMAN) command-line with the Linux Network File System (NFS) and is configured to allow file sharing by RMAN between the source and target systems.

Migrate to Automatic Storage Management

It seems Oracle Automatic Storage Management (ASM) is a strategic direction for Oracle database storage. The migration from the file system to the ASM requires the provision of ASM disks by the storage administrator. For the purpose of the implementation, ASM disks are simulated by creating a raw files type using the Linux operating system "dd" command. This is an optional phase but highly recommended to be considered and can be executed using either Oracle Recovery Manager (RMAN) or Oracle Enterprise Manager Database control. We will use RMAN to complete the migration from a file system to an automatic storage management (ASM) environment.

Upgrade to Oracle Database 12c

The upgrade to Oracle Database 12c is executed on the cloned database. Oracle Database 12c binaries are installed on a separate software location (home).
Prerequisites are verified in preparation of launching the Database Upgrade Assistant (DBUA). DBUA has two upgrade techniques — Graphical User Interface (GUI) and command-line. We will be using the recommended approach, GUI. The database is restarted to operate on archive log mode and the database flashback is enabled.

Applications Regressions Testing

There are two types of testing — technical and functional. For technical testing, adaption of the Oracle Real Applications Testing (RAT) Suite produces highly accurate and quick results. The functional testing is more time consuming. This is due to a complete transaction cycle testing followed by reports generation. It is preferred to conduct both tests on the cloned database to produce highly realistic results. The cloned database flashback feature is enabled for rewinding the database to remove data entered during the functional testing phase.

Synchronize the Target Database

It is necessary to flashback the target database after the functional testing is completed. The flashback rewinds the database to the system change number calculated by the RMAN clone command. The next step is to apply the remote trail files by starting the Replicat processes from the system change number calculated by RMAN. This avoids the overhead of data collision, if enabled, and Replicat terminates with errors due to duplicate records. Depending on the size of the remote trail, the number of Replicat configured and the current transaction volume, the timeframe to reach the sub-second delay varies. This is necessary to achieve before the application transition phase begins.

Data Verification

Now that the Replicats have applied the transactions to the target database, it is time to verify data integrity and accuracy. Configure and deploy Oracle GoldenGate Veridata for data sets confirmation. The applications servers must be disconnected for a short period of time to allow the last streams of transactions to be applied to the target database, and this where the near zero-downtime occurs. Execute Oracle GoldenGate Veridata jobs to confirm all data sets are in "in-sync status", then route the application servers to the new database.

Technical Architecture

Near zero-downtime migrations and upgrades from Oracle Database 11g to Oracle Database 12c is depicted in Figure 7-1. There are several major components to develop and execute for each phase. The Oracle technologies used to complete the migration and upgrade are:

- Oracle GoldenGate 12c
- Oracle GoldenGate 11g Veridata
- Oracle Database Automatic Storage Management (ASM)
- Oracle Database Recovery Manager (RMAN)
- Oracle Database Configuration Assistant (DBCA)
- Oracle Database Upgrade Assistant (DBUA)
- Oracle Database Flashback Technology
- Oracle Database Real Application Testing (RAT)

Keep in mind, depending on the scope of the project, some of the above components may become optional or adapted using alternative technologies. Since this chapter covers Oracle-to-Oracle Database upgrade, only the above list of Oracle components are adapted.

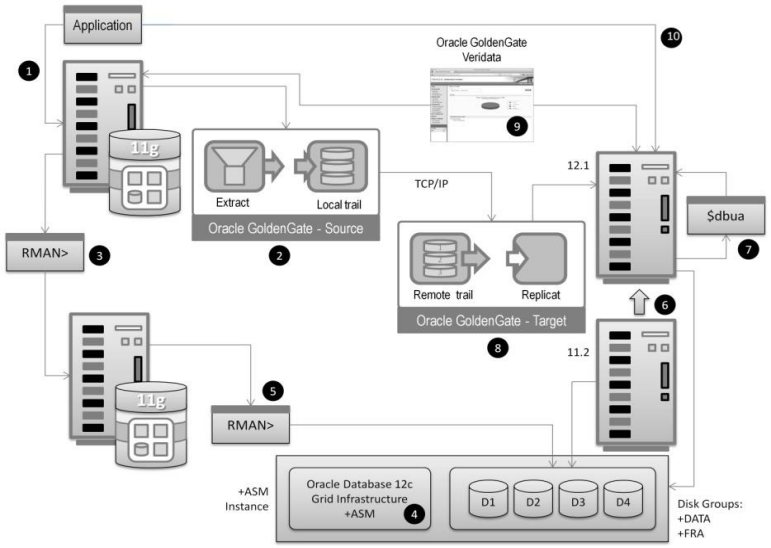


Figure 7-1: Near Zero-Downtime Migration and Upgrade Architecture

Figure 7-1 illustrates highly available applications running on the production system (Source – S1E2 database). Oracle GoldenGate unidirectional configuration is deployed. The primary extracts capture the transaction logs and the data pumps send data to the

remote system (Target). Oracle Database 12c Automatic Storage Management (ASM) instance manages database storage (Disk Groups). Online cloning of the production database leaves this database intact and avoids downtimes during the database data dictionary upgrade. Oracle Database 12c software is installed on the target system and the upgrade of the database is executed using DBUA.
The Replicat applies the remote trail files starting from the System Change Number (SCN) identified by the Oracle Recovery Manager (RMAN) "DUPLICATE" command. Oracle GoldenGate Veridata ascertains the source and target database are identical before diverting the applications from the source to the target database.

Transactions Workload Simulator

The ultimate goal of this chapter is to implement near zero-downtime migration and upgrade using Oracle GoldenGate while users are concurrently processing business transactions. Hence the transactions simulator, TRNAS_LOAD, will remain running on the production system (S1E2 instance) until the upgrade is completed then interrupted and redirected to the target system (T1E2 instance). Oracle GoldenGate is deployed to capture and ship transactions to the target system in preparation of being applied after successfully upgrading Oracle Database 11g to Oracle Database 12c.

Start by opening a terminal window and execute the following anonymous PL/SQL block on the source system (production). *Do not close this terminal window. It must remain open for the duration of the migration and upgrade tasks.*

```
SQL> CONN osm$repapi/oracle
Connected.
SQL> DESC trans_load
PROCEDURE trans_load
 Argument Name                  Type                    In/Out Default?
 ------------------------------ ----------------------- ------ --------
 PLOAD_LEVEL                    NUMBER                  IN     DEFAULT
 PSTARTFROM                     NUMBER                  IN     DEFAULT

SQL> BEGIN
  2     FOR i IN 100001..1000000 LOOP
  3         TRANS_LOAD(1, i);
  4         DBMS_LOCK.SLEEP(2);
  5     END LOOP;
  6  END;
  7  /
```

Using the procedure DBMS_LOCK.SLEEP(2) allows an elapsed time of a two seconds delay for every transaction. The above transaction volume should last for the duration of the upgrade. Our objective is to ultimately interrupt the load simulator TRANS_LOAD explicitly and move it to the migrated and upgraded environment.

Phase I: Deploy Oracle GoldenGate

Successfully deploying Oracle GoldenGate is a key component of near zero-downtime migration and upgrade. While the applications continue to deliver business transactions, Oracle GoldenGate captures the transaction logs and sends data to the target system. For applications with complex database design such as enterprise ERP where it is challenging to maintain the order of foreign keys referential integrity constraints, it is required that the target database foreign key constraints remain disabled during the synchronization process and successfully enabled before the applications are redirected to the new database. The disable/enable foreign key constraints step is developed using database scripts generated from the database dictionary.

Figure 7-2 shows Oracle GoldenGate deployed as a unidirectional classical capture configuration. Another alternative is to employ Oracle GoldenGate integrated capture where Oracle GoldenGate is installable on a downstream system. This avoids additional overhead to the production server as a result of transaction logs capture processing and local trail storage. Refer to Chapter 12 for details regarding Oracle GoldenGate integrated capture implementation.

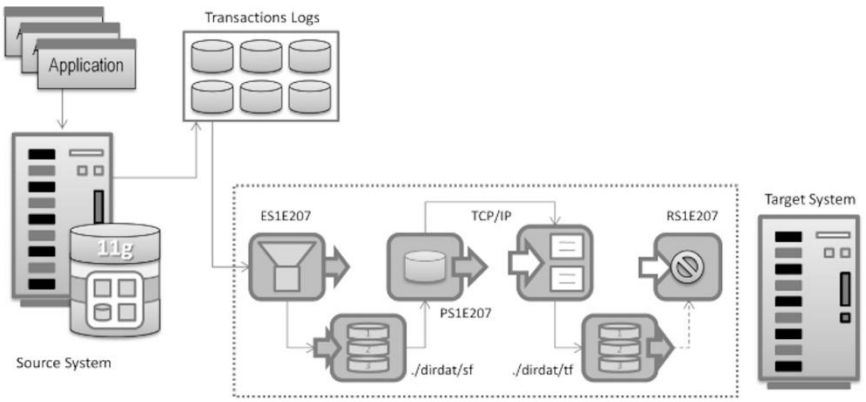


Figure 7-2: Oracle GoldenGate Unidirectional Setup

The configuration is composed of one or more primary extracts and one or more secondary extracts (data pumps). The primary extracts read the transaction logs for the configured list of database tables and create the corresponding local trails. The data pump extracts, over TCP/IP, ship the data to the remote system which creates corresponding remote trails. The Replicat processes remain stopped until the target database (cloned) is upgraded from Oracle Database 11g to Oracle Database 12c. Though the use of data pumps is optional, it is highly recommended for near zero-

downtime implementation to eliminate production server overhead in the event of network unavailability to the target system. Phase I consists of the following tasks:

1. Oracle GoldenGate Software Installation
2. Preparation of the Source Database
3. Enable Table-Level Logging
4. Configure and Test Capture, Route, and Delivery
5. Start Primary Extract (ES1E207)
6. Start Data Pump Extract (PS1E207)

Oracle GoldenGate Software Installation

Successful installation of Oracle GoldenGate on the source (production) and target system must be completed to proceed further. Consider the platform architecture.
This chapter uses Oracle GoldenGate 12c for Oracle Database 11g 64-bit. The same bundle of Oracle GoldenGate 12c is used for Oracle Database 12c 64-bit. Refer to Chapter 2 to learn about installing Oracle GoldenGate 12c.

Preparation of the Source Database

Prior to configuring Oracle GoldenGate Extracts and Replicats, the following database related tasks must be completed:

- Enable Database Supplemental Logging
- Create Oracle GoldenGate Database User

Database Supplemental Logging

Since the database supplemental logging is not enabled by default when the database is created, it is necessary to enable the minimum database supplemental logging –manually. The database supplemental logging records the primary key information for update and delete operations. Oracle GoldenGate locates the target row to update or delete on the target database using the primary key.

Query the data dictionary dynamic view V$DATABASE to find out the level of supplemental logging.

```
S1E2> SELECT supplemental log data min "Minimum",
   2         supplemental_log_data_pk  "Primary key",
   3         supplemental log data ui  "Unique Key",
   4         supplemental_log_data_fk  "Foregin Key",
   5         supplemental_log_data_all "All"
   6  FROM   v$database;
```

```
Minimum  Pri Uni For All
-------- --- --- --- ---
NO       NO  NO  NO  NO

S1E2>
```

Refer to Chapter 1 for more details regarding supplemental logging.

Oracle GoldenGate Database User

The use of a dedicated database user for Oracle GoldenGate is highly recommended. Avoid using an existing active database user. This user requires database privileges listed on Table 7-1.

Database Privilege	Source Database	Target Database
CREATE SESSION, ALTER SESSION	X	X
RESOURCE	X	X
CONNECT	X	X
SELECT ANY DICTIONARY	X	X
FLASHBACK ANY TABLE or FLASHBACK ON<owner.table>	X	
SELECT ANY TABLE or SELECT ON <owner.table>	X	X
INSERT, UPDATE, DELETE ON <target.table>		X
CREATE TABLE		X
EXECUTE on DBMS_FLASHBACK package	X	

Table 7-1: Database Privileges

Refer to Chapter 2 for more details regarding creating an Oracle GoldenGate database user.

Enable Table-Level Logging

Table-Level is added from Oracle GoldenGate GGSCI command-line interface.
The command ADD TRANDATA adds table-level logging using different techniques:

- Specify the schema and table name
- Specify the schema followed by the Oracle GoldenGate wildcard character
- Execute ADD TRANDATA using an obey file

The ADD TRANDATA command requires login to the database from within GGSCI.

```
GGSCI (ggs-source) 1> DBLOGIN USERID ggs_admin, PASSWORD oracle
Successfully logged into database.

GGSCI (ggs-source) 2>
```

Refer to Chapter 2 for more details regarding adding table-level logging.

Configure and Test Capture, Route, and Delivery

The source database has been successfully prepared for Oracle GoldenGate capture. The configuration uses a data pump process to send data to the target system.
This enables the capture process to continue operating in the event that the target system becomes unreachable for a considerable timeframe. It also offloads the primary extract from TCP/IP communications with the target's server collector.
The primary process becomes dedicated to capturing and creating the local trail only. The data pump process continues resending data up on the restoration of the network.

The next three sections implement the configuration. The delivery process remains stopped until the database upgrade is successfully completed and is ready to apply transactions to the target database.

- Configure the Primary Extract
- Configure the Secondary Extract
- Configure the Replicat

Begin by analyzing the database schemas and their tables. This enables the capture of schema by schema, mapping extract to schema and identifying unsupported data types. The following configuration uses the schema OSM$REPAPI.

```
SQL> SELECT table_name
  2  FROM   user_tables;

TABLE_NAME
------------------------------
RANDOM_DOMAIN
RANDOM_FIRST
RANDOM_MIDDLE
RANDOM_LAST
POLICY_TYPES
DISCOUNT_TYPES
CUSTOMERS
POLICIES
OSM_DOCUMENTS

9 rows selected.

SQL>
```

Configure the Primary Extract

The primary extract reads the transaction logs (captures) and writes to the local trail. The extract begins capture from the time it is successfully created or altered to begin just before starting the database cloning task. Then, it continues to capture while the database upgrade is in progress.

Prepare the primary extract parameter file. It identifies the local trail and the list of target tables. For near zero-downtime implementation, the inclusion of all tables is likely required. However, to exclude specific tables, use the parameter TABLEEXECLUDE followed by the list of tables to include. The TABLE parameter uses the Oracle GoldenGate wildcard pattern represented by OSM$REPAPI.* to include the remaining tables.

```
EXTRACT es1e207
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, PASSWORD oracle
EXTTRAIL ./dirdat/sf
TABLEEXCLUDE osm$repapi.OSM_DOCUMENTS;
TABLE osm$repapi.*;
```

Next, create the primary extract ES1E207. Since it is a single-instance database, the parameter THREADS 1 is specified. The extract starts transaction logs capture upon being successfully created, but it may be altered to begin capture just before the database cloning begins.

```
GGSCI (ggs-source) 4> ADD EXTRACT es1e207, TRANLOG, BEGIN now, THREADS 1
EXTRACT added.
```

Associate the primary extract ES1E207 to its local trail. Specify the maximum size of the local trail file. The default size is 100MB.

```
GGSCI (ggs-source) 6> ADD EXTTRAIL ./dirdat/sf, EXTRACT es1e207, MEGABYTES 16
EXTTRAIL added.
```

Configure the Secondary Extract

It is highly recommended to configure the secondary extracts – data pumps. Prepare the parameter file. Since no transformation is required, specify the parameter PASSTHRU mode. The target system is specified by the parameter RMTHOST, followed by the Oracle GoldenGate manager listening port. Use the same TABLEEXECLUDE and TABLE parameters to identify the list of tables.

```
EXTRACT ps1e207
USERID ggs_admin, PASSWORD oracle
PASSTHRU
RMTHOST 192.168.1.209, MGRPORT 7812
RMTTRAIL ./dirdat/tf
TABLEEXCLUDE osm$repapi.OSM_DOCUMENTS;
TABLE osm$repapi.*;
```

Create the data pump extract. The data pump reads the local trail and sends data to the remote system.

```
GGSCI (ggs-source) 5> ADD EXTRACT ps1e207, EXTTRAILSOURCE ./dirdat/sf
EXTRACT added.
```

Associate the data pump extract with its remote trail. Specify the maximum size of the local trail file. The default size is 100MB. To reattempt applying the transaction to the target database, do not configure purge of the remote trail files until the database upgrade is completed.

```
GGSCI (ggs-source) 6> ADD RMTTRAIL ./dirdat/tf, EXTRACT ps1e207, MEGABYTES 16
RMTTRAIL added.
```

Configure the Replicat

Prepare the Replicat parameter file. Since the Replicat requires a database table for a checkpoint, an initial database is created for testing the Replicat. Since database cloning is treated as the technique for initial data load, do not include the HANDLECOLLISIONS parameter. Rather, alter the Replicat to begin applying transactions using the System Change Number (SCN) of the database cloning. This saves unnecessary data collision overhead.

```
REPLICAT rs1e207
SETENV (ORACLE_SID="T1C2")
SETENV (ORACLE_HOME="/u01/app/oracle/product/11.2.0/dbhome_1")
SETENV (NLS_LANG="AMERICAN_AMERICA.AL32UTF8")
USERID ggs_admin, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/rs1e207.dsc, PURGE
MAPEXCLUDE osm$repapi.OSM_DOCUMENTS;
MAP osm$repapi., TARGET osm$repapi.*;
```

```
GGSCI (ggs-target) 3> ADD REPLICAT rs1e207, EXTTRAIL ./dirdat/tf
REPLICAT added.
```

Next, start the primary and secondary extracts. The Replicat remains stopped until the database upgrade is successfully completed.

Start the Primary Extract, ES1E207

Now, start the primary extract. Make the status as RUNNING and it updates the checkpoint timestamp.

```
GGSCI (ggs-source) 31> START EXTRACT es1e207

Sending START request to MANAGER ...
EXTRACT ES1E207 starting

GGSCI (ggs-source) 32> INFO EXTRACT es1e207

EXTRACT    ES1E207   Last Started 2014-04-11 11:24   Status RUNNING
Checkpoint Lag       00:00:02 (updated 00:00:09 ago)
Process ID           32142
Log Read Checkpoint  Oracle Redo Logs
                     2014-04-11 11:25:11  Thread 1, Seqno 92, RBA 1688972
                     SCN 0.3623058 (3623058)
```

Start the Data Pump Extract, PS1E207

Then, start the data pump extract. Make the status as RUNNING and it updates the checkpoint timestamp.

```
GGSCI (ggs-source) 33> START EXTRACT ps1e207

Sending START request to MANAGER ...
EXTRACT PS1E207 starting

GGSCI (ggs-source) 34> INFO EXTRACT ps1e207

EXTRACT    PS1E207   Last Started 2014-04-11 11:25   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:01:31 ago)
Process ID           32216
Log Read Checkpoint  File ./dirdat/sf000000
                     2014-04-11 11:24:22.000000  RBA 491227
```

Verify the Remote Trail

Use the Oracle GoldenGate Logdump utility to ascertain that the transactions are captured and sent to the target system.

```
$ ls -al
total 476
drwxr-x---  2 oracle oinstall   4096 Apr 11 17:35 .
drwxr-xr-x 28 oracle oinstall   4096 Apr  1 23:42 ..
-rw-r-----  1 oracle oinstall 474974 Apr 11 17:37 tf000000
$
```

The following displays transaction details using the Oracle GoldenGate Logdump utility. Perform the same verification on several remote trail files.

Open the Current Remote Trail File

Open the trail file. Use the Logdump utility to open one trail file at a time.

```
Logdump 3 >OPEN ./dirdat/tf000000
Current LogTrail is /u01/app/ggs/12.1.0/gghome_1/dirdat/tf000000
```

Display Transactions Summary

Display summary details. Each trail file indicates the total number of records and each type of DML size.

```
Logdump 4 >COUNT
LogTrail /u01/app/ggs/12.1.0/gghome_1/dirdat/tf000000 has 2017 records
Total Data Bytes            244700
  Avg Bytes/Record             121
Insert                        2016
Others                           1
After Images                  2016

Average of 2017 Transactions
    Bytes/Trans .....         169
    Records/Trans ...           1
    Files/Trans .....           1

Logdump 5 >
```

Table-Level Transaction Details

This is an important Logdump utility command. It shows specifics for each table in terms of DML transaction details.

```
Logdump 6 >COUNT, DETAIL
LogTrail /u01/app/ggs/12.1.0/gghome_1/dirdat/tf000000 has 2017 records
Total Data Bytes            244700
  Avg Bytes/Record             121
Insert                        2016
Others                           1
After Images                  2016

Average of 2017 Transactions
    Bytes/Trans .....         169
    Records/Trans ...           1
    Files/Trans .....           1

*FileHeader*                                    Partition 0
Total Data Bytes              1456
  Avg Bytes/Record            1456
```

```
Others                          1

OSM$REPAPI.CUSTOMERS                                Partition 4
Total Data Bytes            97326
  Avg Bytes/Record             96
Insert                       1008
After Images                 1008

OSM$REPAPI.POLICIES                                 Partition 4
Total Data Bytes           145918
  Avg Bytes/Record            144
Insert                       1008
After Images                 1008
Logdump 7 >
```

Refer to Chapter 14 for more details regarding the Oracle GoldenGate Logdump utility.

Phase II: Online Clone of the Source Database

Achieving near zero-downtime upgrade requires cloning the source database, which is an image copy of the database at a timestamp referred to as the System Change Number (SCN). The primary purpose of cloning the source database is to avoid production database shutdown by the Database Upgrade Assistant (DBUA) during the database data dictionary upgrade. There are several techniques for cloning the database. However, this chapter employs the Oracle Recovery Manager (RMAN) DUPLICATE command to clone the database S1E2 and create the target database T1C1. The database clone is an online operation. It does not incur the source database shutdown.

The adaption of RMAN to clone the source database allows the customization of the target database physical structure, such as:

- Exclusion of read-only tablespaces.
- Exclusion of named tablespaces; such as testing environment tablespaces, or inclusion of list of named tablespaces.
- Open the target database on RESTRICTED SESSION; the default, the target database is opened on READ/WRITE mode.
- Clone without applying archived redo log files.

Figure 7-3 represents a clone database from the source to target system.

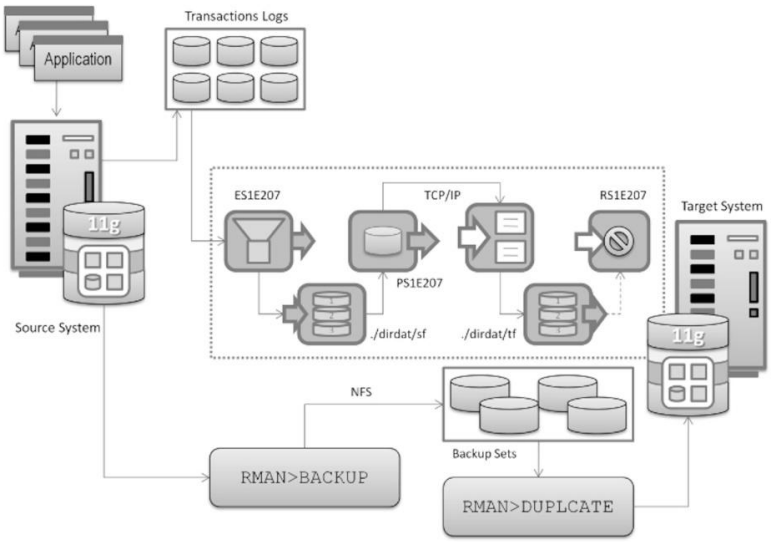


Figure 7-3: Database Cloning to Target System using RMAN

The cloned database uses a different unique Database Identifier (DBID) from the source database. Cloning the database is performed either from a backup or from an active database instance. It is created on the same system or remotely. The cloned database is one of the initial data load techniques. It includes the complete contents of the database. There are four techniques for cloning the database:

1. From an active database connected to the source and target instances
2. From a backup connected to the source and target instances
3. From a backup connected to the target instance and recovery catalog
4. From a backup connected to the target instance

This chapter uses technique 2 : from a backup connected to the source and target instances. The database and archive log files source database backup is completed using RMAN. Using Linux File System (NFS), the backup directory is shared on the remote system.

When database cloning is performed outside of this context, the source database/ instance is referred to as the 'Target' and the target database/instance is referred to as the 'Auxiliary'.

Another option of cloning the database is using Oracle Enterprise Manager Database Control (OEM). This is a Graphical User Interface (GUI) and commonly used by less experienced Database Administrators (DBA). Figure 7-4 represents cloning the database using OEM.

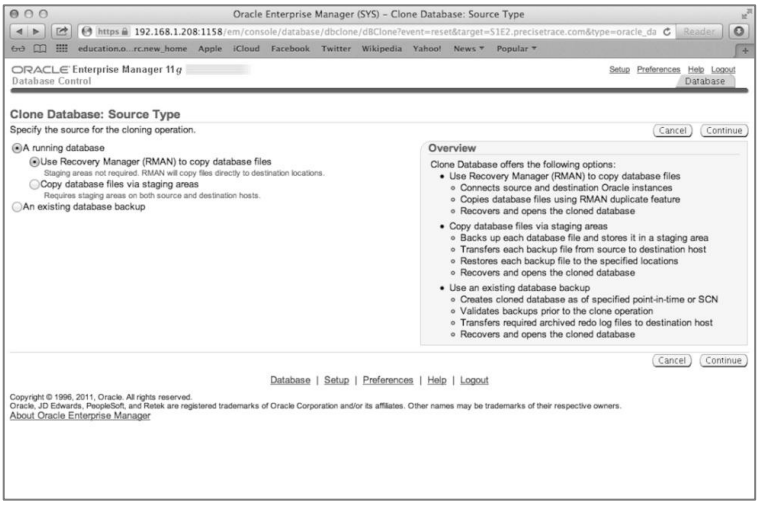


Figure 7-4: Database Cloning using Oracle Enterprise Manager (OEM)

Since the cloned database is created on a different system, configuring the Linux Network File System server and client is illustrated first.

Install Oracle Database 11g

The database clone on a separate system requires the installation of Oracle Database 11g on the target system. The cloned database delivers three functions:

- Initial Data Load from an Active Source Database
- Migration and Upgrade Platform
- Application Testing Platform

Install Oracle Database 11g software only. Do not attempt to create the target database, as it will be created as a cloned database. Figure 7-5 shows Oracle Database 11g Universal Installer. In Step 3 of 10, select the 'install database software only'.

The installation of Oracle Database 11g requires adjusting the Operating System Kernel parameters as per the installation's guide. Or simply copy across the Kernel parameter file of the source system. Then activate the new parameters using the Operating System command – sysctl, as shown below.

```
# scp /etc/sysctl.conf root@192.168.1.209:/etc
# /sbin/sysctl -p
```

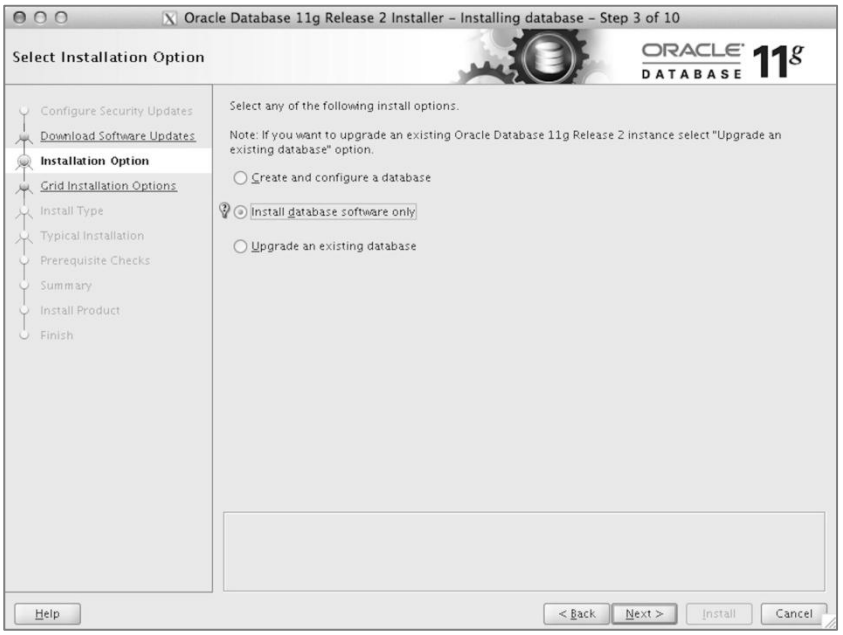


Figure 7-5: Oracle Database 11g Installation

Configure Linux Network File System

The purpose of configuring the Network File System (NFS) is to allow the target system to share directories (mount points) on the source system as if it is stored on the local system storage. This enables quick access to the backup files created by RMAN on the source system database when creating a backup-based clone database.

Considering the source system IP address is 192.168.1.213/192.168.1.208 and the target system is 192.168.1.214/192.168.1.209, execute the steps as follows:

Source System (Server)

Login as root user and execute the steps below.

- Create the shared directory used for RMAN backup files destination.

```
# mkdir -p /home/oracle/backup
# cd /home/oracle
# chown -R oracle:oinstall backup
```

- Allow the target system to mount the directory by modifying the file '/etc/exports.' Add the below line indicating the client IP address and allowed mount options.

```
/home/oracle/backup    192.168.1.214 (rw, sync)
```

- Restart NFS services to activate the configuration.

```
# /sbin/service nfs restart
Shutting down NFS mountd:                                  [  OK  ]
Shutting down NFS daemon:                                  [  OK  ]
Shutting down NFS quotas:                                  [  OK  ]
Shutting down NFS services:                                [  OK  ]
Starting NFS services:                                     [  OK  ]
Starting NFS quotas:                                       [  OK  ]
Starting NFS daemon:                                       [  OK  ]
Starting NFS mountd:                                       [  OK  ]
# /sbin/service portmap restart
Stopping portmap:                                          [  OK  ]
Starting portmap:                                          [  OK  ]
#
```

- Verify NFS configuration settings.

```
# /usr/sbin/showmount -e 192.168.1.213
```

Target System (Client)

Login as root user and execute the steps below.

- Create the directory used for RMAN backup files destination

```
# mkdir -p /home/oracle/backup
# cd /home/oracle
# chown -R oracle:oinstall backup
```

- Mount the directory '/home/oracle/backup' on the source system. Specify the IP address of the server followed by the source and target shared directory.

```
# mount -t nfs 192.168.1.213:/home/oracle/backup /home/oracle/backup
```

- Run RMAN backup command to test NFS setup by specifying the backup location as the NFS mounted directory '/home/oracle/backup.'

```
$ rman

Recovery Manager: Release 11.2.0.3.0 - Production on Sat Apr 12 09:48:44
2014

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights
reserved.
```

```
RMAN> CONNECT TARGET

connected to target database: S1E2 (DBID=4279993164)

RMAN> BACKUP TABLESPACE users
2> FORMAT '/home/oracle/backup/user_%U.dbf';

Starting backup at 12-APR-14
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00004
name=/u01/app/oracle/oradata/S1E2/users01.dbf
channel ORA_DISK_1: starting piece 1 at 12-APR-14
channel ORA_DISK_1: finished piece 1 at 12-APR-14
piece handle=/home/oracle/backup/user_03p5hkkf_1_1.dbf
tag=TAG20140412T095111 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:12
Finished backup at 12-APR-14

RMAN>
```

- Verify RMAN backup is created on the target system. Change the permission on the backup directory to avoid permission denied by the operating system during the restore operation.

```
$ cd /home/oracle/backup
$ ls -al
total 100404
drwxrwxrwx 2    500      501      4096 Apr 12 08:51 .
drwx------ 4 oracle oinstall      4096 Apr 12 00:56 ..
-rw-r----- 1    500      501 102694912 Apr 12 08:51 user_03p5hkkf_1_1.dbf
$ chown oracle:oinstall *
```

Creating a Backup-Based Clone Database

A backup-based clone database uses an online backup of the database files and archived log files. Since the target database (clone) is located on a separate system, override the default backup destination, Fast Recovery Area (FRA). Use the FORMAT parameter to specify a shared directory configured using the Network File System (NFS). The NFS shared directory ensures the backup is immediately available for the RMAN DUPLICATE command to clone the database.

The following RMAN script performs an online backup of the database and archived log files. It begins by allocating three channels to enable backup parallelism.
The script begins backup of the database files, followed by the archived log files.

Both backups are written to the destination '/home/oracle/backup' accessible by the target system as an NFS mounted directory. Upon successful completion of the backup, the channels allocated are released.

Connect using the TARGET option which designates the source database. Verify the database name and DBID returned.

```
RMAN> CONNECT TARGET

connected to target database: S1E2 (DBID=4279993164)

RMAN> RUN {
2> ALLOCATE CHANNEL c1 TYPE DISK;
3> ALLOCATE CHANNEL c2 TYPE DISK;
4> ALLOCATE CHANNEL c3 TYPE DISK;
5>    BACKUP DATABASE INCLUDE CURRENT CONTROLFILE
6>    FORMAT '/home/oracle/backup/db_bck_%U.dbf';
7>    SQL 'ALTER SYSTEM SWITCH LOGFILE';
8>    BACKUP ARCHIVELOG ALL
9>    FORMAT '/home/oracle/backup/db_bck_%U.arc';
10> RELEASE CHANNEL c1;
11> RELEASE CHANNEL c2;
12> RELEASE CHANNEL c3;
13> }
```

From the target system, verify that the created backup and backup files owner, group, and permission are correct. Otherwise, manually execute the operating system shell commands 'chown' and 'chmod' for correcting the backup files settings.

```
# cd /home/oracle/backup
# chown oracle:oinstall *; chmod 777 *
```

Backup File Name Format

The backup files created must be unique, an error is returned when the Recovery Manager (RMAN) tries to overwrite an existing file. The above script uses the substitution variable %U, this ascertains unique backup files names are generated. Other combination of substitution variables are available, but must be combined carefully to produce unique files names. The substitution variable %U combines the substitution variables %u, %p and %c.

As shown in Figure 7-6, the clone database is divided into 3 related phases:

1. Configure the Target Instance
2. Start the Target Instance in NOMOUNT Mode
3. Perform Clone Database using RMAN

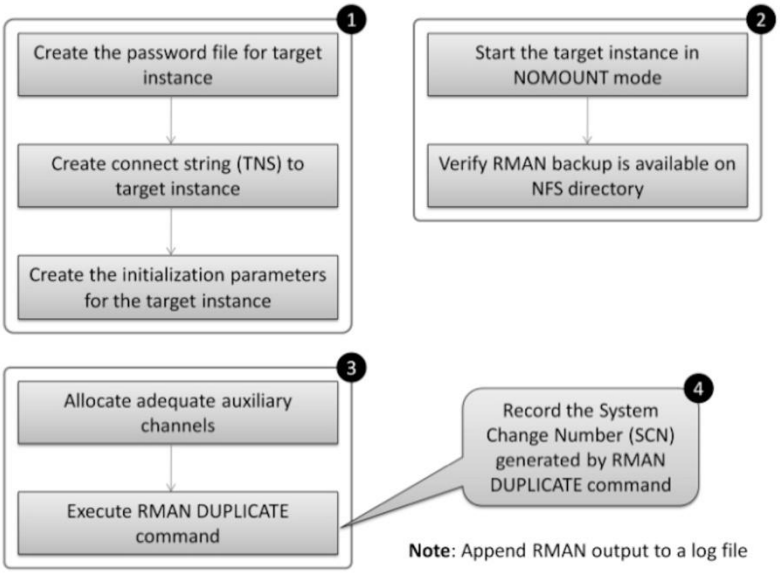


Figure 7-6: Clone Database from Backup

Configure the Target Instance

The configuration prepares the target instance for database cloning. Use the Oracle password utility ORAPWD to create the password file, the Oracle NETMGR, or NETCA utility to create the TNS database connect string and create the parameter file, 'initS1E2.ora'.

- Create the password file on the default location $ORACLE_HOME/dbs

```
$ pwd
/u01/app/oracle/product/11.2.0/dbhome_1/dbs
$ orapwd file=orapwT1C1.ora password=oracle_4U force=y
$ ls
init.ora  orapwT1C1.ora
```

- Create the listener on the target system then create the database connect strings from source to target and vice versa. Test the database connect strings using Oracle TNSPING utility.

  Source to Target Connect String Testing

```
$ tnsping T1C1
```

```
TNS Ping Utility for Linux: Version 11.2.0.3.0 - Production on 18-APR-2014
03:08:06

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 = 192.168.1.209)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
T1C1.precisetrace.com)))
OK (40 msec)
[oracle@ggs-source ~]$
```

Target to Source Connect String Testing

```
$ tnsping S1E2

TNS Ping Utility for Linux: Version 11.2.0.3.0 - Production on 18-APR-2014
10:09:57

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 = 192.168.1.208)(PORT = 1521))) (CONNECT_DATA = (SERVICE_NAME =
S1E2.precisetrace.com)))
OK (10 msec)
$
```

- Create the target parameter file from the source database instance then modify it to the suite of the target database.

```
SQL> CONN / AS SYSDBA
Connected.

SQL> CREATE pfile FROM spfile;

File created.

SQL>

*.audit_file_dest='/u01/app/oracle/admin/T1C1/adump'
*.audit_trail='db'
*.compatible='11.2.0.0.0'
*.control_files='/u01/app/oracle/oradata/T1C1/control01.ctl','/u01/app/oracl
e/fast_recovery_area/T1C1/control02.ctl'
*.db_block_size=8192
*.db_domain='precisetrace.com'
*.db_name='T1C1'
*.db_recovery_file_dest='/u01/app/oracle/fast_recovery_area'
*.db_recovery_file_dest_size=4322230272
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=T1C1XDB)'
*.log_archive_format='%t_%s_%r.dbf'
```

```
*.memory_target=2147483648
*.open_cursors=300
*.processes=150
*.remote_login_passwordfile='EXCLUSIVE'
*.undo_tablespace='UNDOTBS1'
*.db_file_name_convert='/u01/app/oracle/oradata/S1E2/',
                       '/u01/app/oracle/oradata/T1C1/'
*.log_file_name_convert='/u01/app/oracle/oradata/S1E2/',
                       '/u01/app/oracle/oradata/T1C1/'
```

Modify then save the file as initT1C1.ora and copy it over to the target system.

```
$ scp initT1C1.ora
oracle@192.168.1.209:/u01/app/oracle/product/11.2.0/dbhome_1/dbs
```

Create the directory structure as needed for the database instance T1C1.

```
$ cd $ORACLE_BASE/admin
$ mkdir -p T1C1/cdump T1C1/udump T1C1/adump
$ cd T1C1
$ ls
adump  cdump  udump
$ cd $ORACLE_BASE/oradata
$ mkdir T1C1
$ cd $ORACLE_BASE
$ cd fast_recovery_area/
$ mkdir T1C1
$
```

Startup the Target Instance

Start the target instance using the parameter file 'initT1C1.ora'. The instance is started in NOMOUNT mode. Then, create the server parameter file (spfileT1C1.ora). Shutdown and startup the instance in MOUNT mode using the parameter file spfileT1C1.ora.

```
$ export ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
$ export ORACLE_SID=T1C1
$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Fri Apr 18 09:38:42 2014

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> CONN / AS SYSDBA
Connected to an idle instance.
SQL> STARTUP NOMOUNT
pfile='/u01/app/oracle/product/11.2.0/dbhome_1/dbs/initT1C1.ora'
ORACLE instance started.

Total System Global Area   551165952 bytes
Fixed Size                   2230232 bytes
Variable Size              331352104 bytes
Database Buffers           213909504 bytes
Redo Buffers                 3674112 bytes
SQL>
```

```
SQL> CREATE spfile FROM PFILE;

File created.

SQL>
```

Perform Clone Database using RMAN

Now, we are ready to perform the database clone. While logged in to the target system, connect to the target instance (T1C1) and remotely connect to the source instance (S1E2). The RMAN DUPLICATE command runs on verbose mode.
To avoid 'permission denied error' returning from the operating system while RMAN restores backup files, prior to starting the command ascertain the correct permission settings on the target system NFS directory '/home/oracle/backup.'

```
$ export ORACLE_HOME=/u01/app/oracle/product/11.2.0/dbhome_1
$ cd $ORACLE_HOME/bin
$ ./rman

Recovery Manager: Release 11.2.0.3.0 - Production on Fri Apr 18 10:14:59 2014

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

RMAN> CONNECT AUXILIARY sys/oracle_4U

connected to auxiliary database: T1C1 (not mounted)

RMAN> CONNECT TARGET sys/oracle_4U@S1E2

connected to target database: S1E2 (DBID=4279993164)

RMAN>

RMAN> RUN {
2> DUPLICATE TARGET DATABASE TO T1C1
3> BACKUP LOCATION '/home/oracle/backup'
4> NOFILENAMECHECK;
5> }
```

The above RMAN DUPLICATE script overrides the default backup location, the Fast Recovery Area (FRA) by specifying the non-default destination using the parameter BACKUP LOCATION. The parameter NOFILENAMECHECK prevents RMAN from checking whether source database data files with the same name as clone database data files are in use. Specify this option when the source and clone database data files and redo log files have the same names. This option is specified because the clone database is on a different system with the same disk configuration, file system structure, and file names as the source database system. Without the parameter NOFILENAMECHECK, the command will terminate with an error.

If you receive the error message ' RMAN-06025: no backup of archived log for thread 1 with sequence 255 and starting SCN of 8375157 found to restore', then use the NOREDO option as illustrated below.

```
RMAN> RUN {
2> DUPLICATE TARGET DATABASE TO T1C1
3> NOREDO
4> BACKUP LOCATION '/home/oracle/backup'
5> NOFILENAMECHECK;
6> }
```

Target Replicat Starting SCN

The above RMAN DUPLICATE command determined the System Change Number (SCN) for performing point in-time recovery and opened the target database in RESETLOGS mode.

The following section of the RMAN DUPLICATE command log shows the SCN 14274705. This will be the application point of the Replicat for synchronizing the target database – T1C1.

```
contents of Memory Script:
{
   set until scn 14274705;
   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";
   restore
   clone database
   ;
}
```

Post Database Clone

Modify the file '/etc/oratab' to include the clone database reference. To avoid a re-clone database in the event of failure, start a consistent database backup using RMAN. First, shutdown the database using the IMMEDIATE option, then startup in MOUNT state.

```
RMAN> STARTUP MOUNT

connected to target database (not started)
Oracle instance started
database mounted
```

```
Total System Global Area     551165952 bytes

Fixed Size                     2230232 bytes
Variable Size                369100840 bytes
Database Buffers             176160768 bytes
Redo Buffers                   3674112 bytes

RMAN> BACKUP DATABASE;
```

Verify the Clone Database Structure

Since the RMAN DUPLICATE command did not exclude any of the source database tablespaces after the clone database operation was successfully completed, the clone database (target) must have identical physical data file structures.

The following is the list of the clone database data files, control files, online redo log files, list of tables owned by OSM$REPAPI, and the current row counts of the table POLICIES.

```
$ . oraenv
ORACLE_SID = [oracle] ? T1C1
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus /nolog

SQL*Plus: Release 11.2.0.3.0 Production on Fri Apr 18 18:51:14 2014

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

SQL>CONN / AS SYSDBA
Connected.
SQL> SELECT file_name FROM dba_data_files UNION
  2  SELECT file_name FROM dba_temp_files UNION
  3  SELECT name      FROM v$controlfile  UNION
  4  SELECT member    FROM v$logfile;

FILE_NAME
--------------------------------------------------------------------------------
/u01/app/oracle/fast_recovery_area/T1C1/control02.ctl
/u01/app/oracle/oradata/T1C1/control01.ctl
/u01/app/oracle/oradata/T1C1/redo01.log
/u01/app/oracle/oradata/T1C1/redo02.log
/u01/app/oracle/oradata/T1C1/redo03.log
/u01/app/oracle/oradata/T1C1/sysaux01.dbf
/u01/app/oracle/oradata/T1C1/system01.dbf
/u01/app/oracle/oradata/T1C1/temp01.dbf
/u01/app/oracle/oradata/T1C1/undotbs01.dbf
/u01/app/oracle/oradata/T1C1/users01.dbf

10 rows selected.

SQL> SELECT table_name
  2  FROM   dba_tables
  3  WHERE  owner='OSM$REPAPI';
```

```
TABLE_NAME
------------------------------
RANDOM_DOMAIN
RANDOM_FIRST
RANDOM_MIDDLE
RANDOM_LAST
POLICY_TYPES
DISCOUNT_TYPES
CUSTOMERS
POLICIES
OSM_DOCUMENTS

9 rows selected.

SQL> SELECT count(*) FROM osm$repapi.policies;

  COUNT(*)
----------
    921740

SQL>
```

Phase III: Migrate to the Automatic Storage Management

The migration from the file system to the Automatic Storage Management (ASM) allows the upgraded database to take advantage of high availability features delivered by ASM, which are necessary for supporting 24/7 systems. After the migration is successfully completed, the database continues to function as it was before the migration. However, ASM enables the Database Administrator (DBA) to manage the database storage more effectively by performing online operations that may require database shutdown on a non-ASM environment. An example of an online operation delivered by ASM is adding and removing disk storage while the database is operational which avoids user interruption by keeping the database open during the operation execution.

Before migration from the file system to the ASM, perform the following configurations:

- Create, configure, and provision ASM disks
- Create and configure ASM disk groups

Figure 7-7 depicts ASM architecture and migrating database data file from file system to ASM using RMAN. The high-level component of ASM is the disk group, which logically groups ASM disks – the physical disk partitions provisioned for ASM. A single disk group may be referenced by one or more databases.

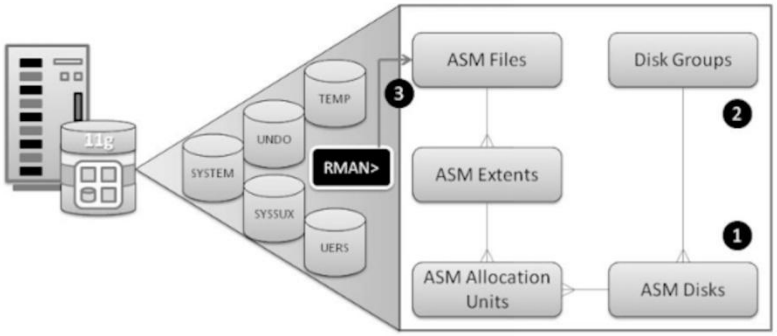


Figure 7-7: Automatic Storage Management Architecture

Oracle ASM disks are the storage devices associated with an Oracle ASM disk group. There are several forms of an ASM disk : an entire physical disk, combination of one or more disk partitions, a Logical Unit Number (LUN), a logical volume (LV), or network file system. Each ASM disk is composed of ASM allocation units, the smallest contiguous amount of disk space that ASM allocates. When you create an ASM disk group, the ASM allocation unit of 1, 2, 4, 8, 16, 32, or 64 MB is specified as a disk group attribute. The default is 1MB. ASM allocation units form an ASM extent associated with an ASM file. An Oracle ASM extent is the raw storage which holds the contents of an Oracle ASM file. An Oracle ASM file consists of one or more extents tracked using an extent map table.

Installation of Oracle 12c Grid Infrastructure

Before migration to the ASM, it is necessary to install Oracle 12c Grid Infrastructure and configure the disk groups. The installation requires using a dedicated operating system user that is different from Oracle Database 12c. Figure 7-8 shows the installation options.

The Oracle Universal Installer for Oracle 12c Grid Infrastructure has four installation options:

- Install and Configure Oracle Grid Infrastructure for a Cluster
- Install and Configure Oracle Grid Infrastructure for a Standalone Server
- Upgrade Oracle Grid Infrastructure or Oracle Automatic Storage Management
- Install Oracle Grid Infrastructure Software Only

Since the ASM disks are not provisioned yet, select the option 'Install Oracle Grid Infrastructure Software Only'. After the ASM disks are provisioned, then use Oracle Automatic Storage Management Configuration Assistant (ASMCA) to create,

start the ASM instance +ASM and create and configure the disk groups +DATA and +FRA.

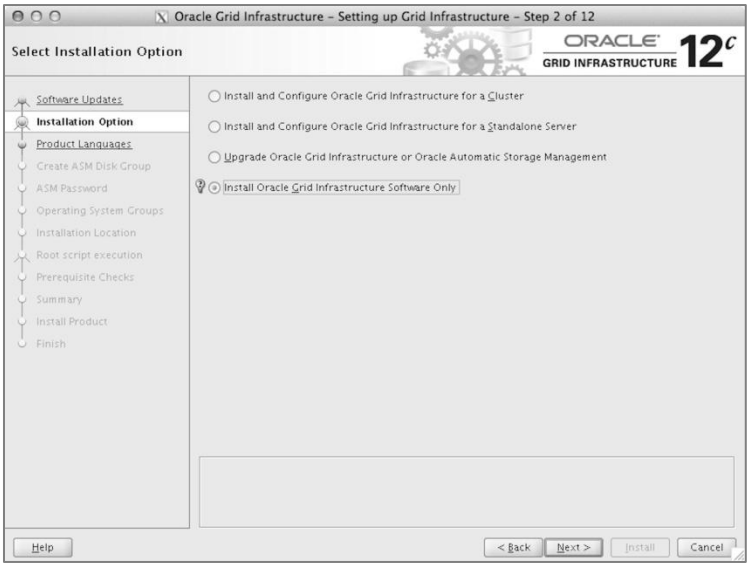


Figure 7-8: Oracle Grid Infrastructure 12c

For details regarding creating and providing ASM disks, refer to the section 'Building Virtual File System for ASM Disks.'

Working with ASM Instance

An ASM Instance is installed on Oracle Database 12c server. There must be one and only one ASM instance running on the system. The ASM instance must be started in order for the database instance to access data files. Connect as SYSASM to start the ASM instance, which mount the list of created and configured disk groups.

```
$ . oraenv
ORACLE_SID = [+ASM] ? +ASM
The Oracle base remains unchanged with value /u02/app/grid
$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Wed Apr 30 21:40:21 2014

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> CONN / AS SYSASM
Connected to an idle instance.
SQL> STARTUP
ASM instance started

Total System Global Area   1135747072 bytes
```

```
Fixed Size                  2297344 bytes
Variable Size            1108283904 bytes
ASM Cache                  25165824 bytes
ASM diskgroups mounted
SQL>
```

The ASM Instance uses fewer parameters than the database instance. The only mandatory parameter-value is INSTANCE_TYPE=ASM. View the ASM instance parameters related to list the auto-mounted disk groups and ASM disks storage devices.

```
SQL> SHOW PARAMETERS disk

NAME                                 TYPE           VALUE
-------------------------------   -----------    ----------------------
asm_diskgroups                       string         DATA, FRA
asm_diskstring                       string         /dev/x*
SQL>
```

This instance works in conjunction with the database instance to handle database file management. When the database instance submits the following CREATE TABLESPACE statement:

```
SQL> CREATE TABLESPACE edge
  2  DATAFILE '+DATA' SIZE 50M AUTOEXTEND on;

Tablespace created.

SQL> SELECT file_name
  2  FROM   dba_data_files
  3  WHERE  tablespace_name='EDGE';

FILE_NAME
-----------------------------------------------------------------
+DATA/t1c1/datafile/edge.267.846273245

SQL>
```

The tablespace data file is created as an ASM file as follows:

1. The database requests creating the data file on the specified disk group, +DATA.
2. The ASM Instance allocates the storage based on the Allocation Unit (AU) of the disk group. ASM spreads the allocation units evenly across all ASM disks, within the disk group specified, to balance load and avoid hot spots. It also mirrors allocation units for the disk group with redundant normal or high attribute.
3. The database background process ASMB receives the extent map for the new file. The extent map is a table that maps data extents in a file to allocation units on disk. This allows the database instance access to the data.
4. The file is now open and the database process starts formatting the file.
5. The ASM Instance confirms the file creation and marks the file as created.
6. The file is closed and reopened by the database instance.

Database Disk Groups

There are two disk groups created; +DATA and +FRA. Creating disk groups requires ASM disks. The +DATA disk group hosts all types of database data files and the +FRA disk group is dedicated for the Fast Recovery Area (FRA) files only.

Figure 7-9 is the Automatic Storage Management Configuration Assistant (ASMCA). It manages disk groups. Use ASMCA to create/alter/drop disk groups, add disks to a disk group, drop disks from a disk group, or mount /dismount disk groups.

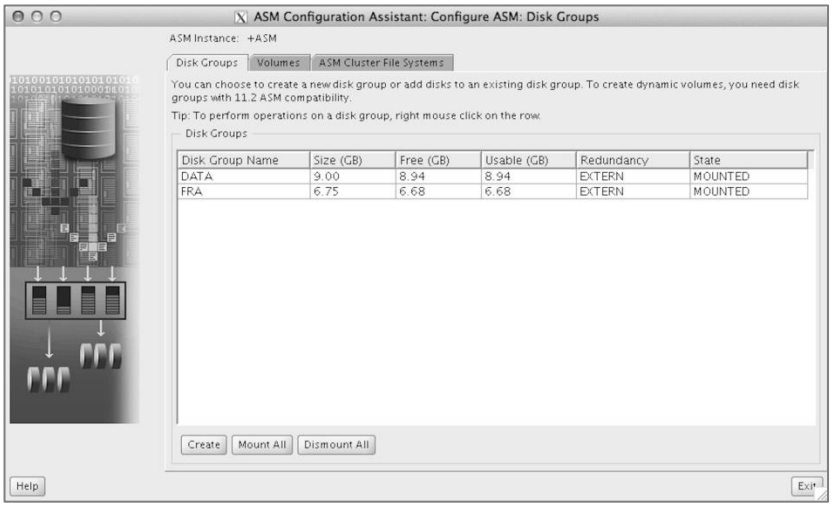


Figure 7-9: Automatic Storage Management Configuration Assistant (ASMCA)

ASM Instance has its own command-line – ASMCMD. ASMCMD adapts Linux-like file system management commands. The following illustrates the copy of the archived log file from ASM to file system using the copy command. The '+' character designates the ASM root directory. Use the change directory 'cd' command to navigate across the ASM directory structure.

```
$ . oraenv
ORACLE_SID = [grid] ? +ASM
The Oracle base has been set to /u02/app/grid
$ asmcmd
ASMCMD> ls
DATA/
FRA/
ASMCMD> cd +FRA/T1C1/ARCHIVELOG/2014_05_02
ASMCMD> ls
thread_1_seq_24.257.846500305
ASMCMD> cp * /home/grid/arc
copying +FRA/T1C1/ARCHIVELOG/2014_05_02/thread_1_seq_24.257.846500305 ->
/home/grid/arc/thread_1_seq_24.257.846500305
```

```
ASMCMD>
```

Testing the creation of tablespace in +DATA disk group. The next CREATE TABLESPACE command creates the tablespace EDGE. The data file is auto extended with an initial size of 100MB.

```
$ . oraenv
ORACLE_SID = [+ASM] ? T1C2
The Oracle base has been changed from /u02/app/grid to /u01/app/oracle
$ sqlplus /nolog

SQL*Plus: Release 11.2.0.3.0 Production on Tue Apr 22 18:34:38 2014

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

SQL> CONN / AS SYSDBA
Connected.
SQL> CREATE TABLESPACE edge
  2  DATAFILE '+DATA';

Tablespace created.

SQL> SELECT tablespace_name
  2  FROM   dba_tablespaces;

TABLESPACE_NAME
------------------------------
SYSTEM
SYSAUX
UNDOTBS1
TEMP
USERS
EDGE

6 rows selected.

SQL> DROP TABLESPACE edge;

Tablespace dropped.

SQL>
```

The above commands were done on a test database – T1C2. It is necessary to create a test database of type Automatic Storage Management (ASM). The test confirms Oracle Grid Infrastructure 12c is properly installed and disk groups are configured for the migration. View the +ASM instance alert file. Investigate and resolve errors found before proceeding from Migration to the Automatic Storage Management. Use DBCA to drop the test database. This will free disk groups space.

This next section outlines performing the migration to the Automatic Storage Management (ASM).

Migration to the Automatic Storage Management

Migration of the clone database storage from the file system to the Automatic Storage Management (ASM) is executed using either the Oracle Enterprise Manager (OEM) 'Migrate Database to ASM' wizard or the Oracle Recovery Manager (RMAN) DUPLICATE command. Very Large Database (VLDB) environments benefit using RMAN for migrating the database gradually. For example, processing the migration based on the disk storage availability and the ownership of tablespaces by applications allows faster and manageable restoration in case of failure. The RMAN DUPLICATE command supports a database clone from an active database. This option minimizes the amount of trail files to synchronize the target database after successfully upgrading the database to Oracle Database 12c. Figure 7-10 represents the migration workflow using the RMAN DUPLICATE command, which requires preparation of the disk groups +DATA and +FRA for the database file storage.

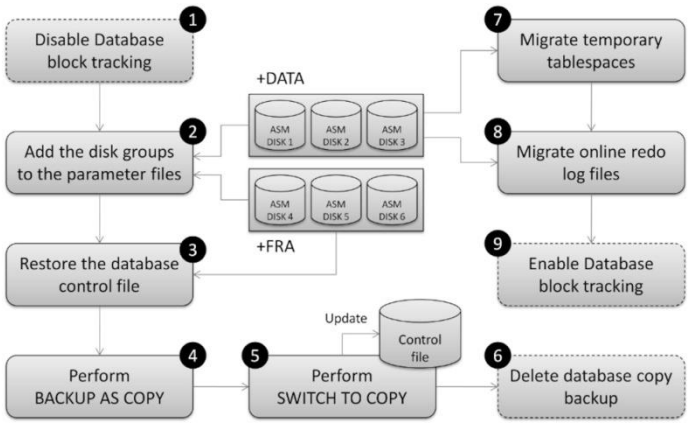


Figure 7-10: Migration Workflow using RMAN

The following steps are executed on the clone database using SQL*Plus and Oracle Recovery Manager (RMAN).

1. Disable the database block change tracking process
2. Modify the parameter file by adding the ASM disk groups
3. Migrate the control file
4. Migrate the data files
5. Migrate the temporary tablespaces
6. Migrate the online redo log files groups
7. Enable the database block change tracking process

Disable Block Change Tracking

When the block change tracking is enabled, for each database block that is changed, the background process CTWR writes the physical address to an external file – outside of the database. The purpose is to support RMAN incremental backups. During the migration to the ASM, if the database block change tracking process is enabled, then it must be disabled prior to starting the migration steps. To determine if the process is enabled or disabled, query the database dynamic view V$BLOCK_CHANGE_TRACKING. If the status column returns 'ENABLED', then disable it, otherwise proceed to the next step.

```
SQL> SELECT status, filename FROM v$block_change_tracking;

STATUS     FILENAME
---------- ------------------------------------------------------------
ENABLED    /u01/app/oracle/…/T1C1/changetracking/o1_mf_9ozlbm2w_.chg

SQL> ALTER DATABASE DISABLE BLOCK CHANGE TRACKING;

Database altered.

SQL>
```

Define Disk Groups

Migration to the Automatic Storage Management (ASM) requires the ASM disk groups to be created and mounted. This migration uses two disk groups, +DATA and +FRA. The disk group +DATA stores the database data files, redo log files, and control files. The disk group +FRA is the location of the database Fast Recovery Area (FRA), the default location for RMAN backup files, flashback log files, and the mirror copy of the control file and online redo log files.

- Determine the name and status of +DATA and +FRA disk groups by querying the ASM dynamic view V$ASM_DISKGROUP.

```
SQL> SELECT name, state
  2  FROM   v$asm_diskgroup;

NAME                           STATE
------------------------------ -----------
FRA                            MOUNTED
DATA                           MOUNTED

SQL>
```

- Create the parameter file 'initT1C1.ora' from the server parameter file 'spfileT1C1.ora'. Use IMMEDIATE shutdown option to force checkpoint and closes the database files.

```
SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> CONN / AS SYSDBA
Connected to an idle instance.
SQL> CREATE pfile FROM spfile;

File created.

SQL>
```

- Add and/or modify the parameters file by adding the disk groups +DATA and +FRA as the locations of the control files, data files, and online redo log files. Table 7-2 lists the modified parameters.

Parameter	Description
CONTROL_FILES	The location of Oracle managed control files.
DB_CREATE_ONLINE_LOG_DEST_1	The location of the first member of the online redo log files and control file.
DB_CREATE_FILE_DEST	The location of the data file
DB_RECOVERY_FILE_DEST	The location of the Fast Recovery Area (FRA).

Table 7-2: Automatic Storage Management File Locations

```
*.audit_file_dest='/u01/app/oracle/admin/T1C1/adump'
*.audit_trail='db'
*.compatible='11.2.0.0.0'
*.control_files=(+DATA, +FRA)
*.db_create_online_log_dest_1=+DATA
*.db_create_file_dest=+DATA
*.db_recovery_file_dest=+FRA
*.db_recovery_file_dest_size=4322230272
*.db_block_size=8192
*.db_domain='precisetrace.com'
*.db_name='T1C1'
*.db_recovery_file_dest_size=4322230272
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=T1C1XDB)'
*.log_archive_format='%t_%s_%r.dbf'
*.memory_target=2147483648
*.open_cursors=300
*.processes=150
*.remote_login_passwordfile='EXCLUSIVE'
*.undo_tablespace='UNDOTBS1'
```

- Create the server parameter file 'spfileT1C1.ora' from the modified parameter file 'initT1C1.ora.' The two parameter files are currently stored on the default location - $ORACLE_HOME/dbs. After the migration to the ASM is completed, use the

Oracle supplied PL/SQL package DBMS_FILE_TRANSFER to move the server parameter file to ASM storage.

```
SQL> CREATE spfile FROM pfile;

File created.

SQL>
```

Control File Migration

The migration begins by restoring an image copy of the current control file to the location(s) specified by the parameter CONTROL_FILES. Start the Oracle Recovery Manager (RMAN), connect to the target database, start the instance in NOMOUNT mode and perform the RMAN RESTORE command of the control file.

- Start Oracle Recovery Manager (RMAN) and start the instance in NOMOUNT mode:

```
$ rman

Recovery Manager: Release 11.2.0.3.0 - Production on Tue Apr 22 19:35:20
2014

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights
reserved.

RMAN> CONNECT TARGET

connected to target database (not started)

RMAN> STARTUP NOMOUNT

Oracle instance started

Total System Global Area     551165952 bytes

Fixed Size                     2230232 bytes
Variable Size                331352104 bytes
Database Buffers             213909504 bytes
Redo Buffers                   3674112 bytes

RMAN>
```

- Based on the current location of the control file identified from 'initT1C1.ora', use Oracle Recovery Manager (RMAN) RESTORE CONTROLFILE command to restore a copy of the control file to +DATA and +FRA.

```
RMAN> RESTORE CONTROLFILE FROM
2> '/u01/app/oracle/oradata/T1C1/control01.ctl';

Starting restore at 22-APR-14
```

```
using channel ORA_DISK_1

channel ORA_DISK_1: copied control file copy
output file name=+DATA/t1c1/controlfile/current.256.845581127
output file name=+FRA/t1c1/controlfile/current.257.845581127
Finished restore at 22-APR-14

RMAN>
```

Data File Migration

After the control is successfully restored, the next step is to migrate the database permanently and undo data files. One method to achieve this is by creating a backup copy of the database, which is a consistent backup of the database data files. The file location is explicitly identified by the FORMAT parameter as +DATA disk group. Using the RMAN COPY command creates an image copy of the data files to the location specified by the FORMAT parameter.

- Optionally, enable Recovery Manager (RMAN) parallelism. This enhances the data file copy operations by allocating multiple channels. A parallelism of 6 allocates six RMAN database channels to create the database image copy.

```
RMAN> STARTUP MOUNT

database is already started
database mounted
released channel: ORA_DISK_1

RMAN> CONFIGURE DEVICE TYPE DISK PARALLELISM 6;

new RMAN configuration parameters:
CONFIGURE DEVICE TYPE DISK PARALLELISM 6 BACKUP TYPE TO BACKUPSET;
new RMAN configuration parameters are successfully stored
```

- Create an image copy of the database data files to the +DATA disk group. The database is in MOUNT mode to create consistent image backup files. The COPY command excludes the files belonging to the database temporary tablespace, which will migrate separately.

```
RMAN> BACKUP AS COPY DATABASE FORMAT '+DATA';

Starting backup at 22-APR-14
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=26 device type=DISK
allocated channel: ORA_DISK_2
channel ORA_DISK_2: SID=27 device type=DISK
allocated channel: ORA_DISK_3
channel ORA_DISK_3: SID=28 device type=DISK
allocated channel: ORA_DISK_4
channel ORA_DISK_4: SID=29 device type=DISK
allocated channel: ORA_DISK_5
```

```
channel ORA_DISK_5: SID=30 device type=DISK
allocated channel: ORA_DISK_6
channel ORA_DISK_6: SID=31 device type=DISK
channel ORA_DISK_1: starting datafile copy
input datafile file number=00002
name=/u01/app/oracle/oradata/T1C1/sysaux01.dbf
channel ORA_DISK_2: starting datafile copy
input datafile file number=00001
name=/u01/app/oracle/oradata/T1C1/system01.dbf
channel ORA_DISK_3: starting datafile copy
input datafile file number=00004
name=/u01/app/oracle/oradata/T1C1/users01.dbf
channel ORA_DISK_4: starting datafile copy
input datafile file number=00003
name=/u01/app/oracle/oradata/T1C1/undotbs01.dbf
output file name=+DATA/t1c1/datafile/users.259.845581373
tag=TAG20140422T194250 RECID=7 STAMP=845581412
channel ORA_DISK_3: datafile copy complete, elapsed time: 00:00:51
output file name=+DATA/t1c1/datafile/undotbs1.260.845581377
tag=TAG20140422T194250 RECID=8 STAMP=845581424
channel ORA_DISK_4: datafile copy complete, elapsed time: 00:00:57
output file name=+DATA/t1c1/datafile/system.258.845581371
tag=TAG20140422T194250 RECID=9 STAMP=845581560
channel ORA_DISK_2: datafile copy complete, elapsed time: 00:03:12
output file name=+DATA/t1c1/datafile/sysaux.257.845581371
tag=TAG20140422T194250 RECID=10 STAMP=845581564
channel ORA_DISK_1: datafile copy complete, elapsed time: 00:03:24
Finished backup at 22-APR-14

Starting Control File and SPFILE Autobackup at 22-APR-14
piece handle=+FRA/t1c1/autobackup/2014_04_22/s_845580320.258.845581577
comment=NONE
Finished Control File and SPFILE Autobackup at 22-APR-14

RMAN>
```

- Now, the control and data files are moved to Automatic Storage Management (ASM). The next step is to update the control files with the new locations of the database data files then open the database in normal mode. RMAN SWITCH command updates the file names for a database, tablespace, or data file to the latest image copies available for the specified files or database.

```
RMAN> SWITCH DATABASE TO COPY;

using target database control file instead of recovery catalog
datafile 1 switched to datafile copy
"+DATA/t1c1/datafile/system.258.845581371"
datafile 2 switched to datafile copy
"+DATA/t1c1/datafile/sysaux.257.845581371"
datafile 3 switched to datafile copy
"+DATA/t1c1/datafile/undotbs1.260.845581377"
datafile 4 switched to datafile copy
"+DATA/t1c1/datafile/users.259.845581373"

RMAN> ALTER DATABASE OPEN;
```

```
database opened

RMAN> exit

Recovery Manager complete.
$
```

The database migration to the Automatic Storage Management (ASM) is partially completed and the database is open in read/write mode. The next steps manually migrate the remaining of the database – temporary and online redo log files. Migration of temporary and online redo log files is an in-place operation.
The database remains open while executing migration commands.

Temporary Files Migration

The Recovery Manager (RMAN) command BACKUP AS COPY DATABASE excludes the temporary tablespace files. It is performed by manually creating an interim temporary tablespace to allow the recreation of the existing temporary tablespace as Automatic Storage Management (ASM) type.

- Determine the name and size of the temporary files. Since the database is open, query the database data dictionary view DBA_TEMP_FILES.

```
SQL> CONN / AS SYSDBA
Connected.
SQL> SELECT file_name, bytes/1024/1204 MB
  2  FROM   dba_temp_files;

FILE_NAME                                                MB
-------------------------------------------------- -----------
/u01/app/oracle/oradata/T1C1/temp01.dbf                549.42

SQL>
```

- Create an interim temporary tablespace then alter the default database-wide temporary tablespace. Query the data dictionary DATABASE_PROPERTIES view to ascertain the default database-wide temporary tablespace.

```
SQL> CREATE TEMPORARY TABLESPACE temp_1
  2  TEMPFILE '/u01/app/oracle/oradata/T1C1/temp02.dbf' SIZE 550M;

Tablespace created.

SQL> ALTER DATABASE DEFAULT TEMPORARY TABLESPACE temp_1;

Database altered.
```

- Drop and recreate the original temporary tablespace. The new temporary tablespace is created as an Automatic Storage Management (ASM) file type.

Make the temporary tablespace size match the original size.

```
SQL> DROP TABLESPACE temp INCLUDING CONTENTS;

Tablespace dropped.

SQL> CREATE TEMPORARY TABLESPACE temp
  2  TEMPFILE '+DATA' SIZE 550M;

Tablespace created.

SQL> ALTER DATABASE DEFAULT TEMPORARY TABLESPACE temp;

Database altered.
```

- Drop the interim temporary tablespace and verify the temporary tablespace is created as an Automatic Storage Management (ASM) file type.

```
SQL> DROP TABLESPACE temp_1 INCLUDING CONTENTS;

Tablespace dropped.

SQL> SELECT file_name
  2  FROM   dba_temp_file;

FILE_NAME
--------------------------------------------------------------------------------
+DATA/t1c1/tempfile/temp.261.845582177

SQL>
```

Online Redo Log Files Migration

Since Oracle Recovery Manager (RMAN) does not backup online redo log files, the migration from a file system to Automatic Storage Management (ASM) type is performed manually. As illustrated below, it is an in-place operation and does not require shutdown of the database.

- List of the current online redo group members by a joint query of V$LOGFILE and V$LOG.

```
SQL> SELECT l.group#, lf.member, l.bytes/1024/1024
  2  FROM   v$logfile lf, v$log l
  3  WHERE  lf.group#=l.group#
  4  ORDER BY 1;

    GROUP# MEMBER                                    L.BYTES/1024/1024
---------- ----------------------------------------  -----------------
         1 /u01/app/oracle/oradata/T1C1/redo01.log                  50
         2 /u01/app/oracle/oradata/T1C1/redo02.log                  50
         3 /u01/app/oracle/oradata/T1C1/redo03.log                  50
```

```
SQL>
```

- Create new online redo log groups, each group composed of two members. The first member is stored on the +DATA disk group and the second member on the +FRA disk group. According to the above query, the next three groups to be created are 4, 5 and 6. Use the same returned size.

```
SQL> ALTER DATABASE ADD LOGFILE GROUP 4 ('+DATA', '+FRA') SIZE 50M;

Database altered.

SQL> ALTER DATABASE ADD LOGFILE GROUP 5 ('+DATA', '+FRA') SIZE 50M;

Database altered.

SQL> ALTER DATABASE ADD LOGFILE GROUP 6 ('+DATA', '+FRA') SIZE 50M;

Database altered.
```

- Drop the non-ASM online redo log groups. Start by dropping INACTIVE and UNUSED groups only. The query below shows group 1 status is CURRENT and cannot be dropped until several log switches are archived and it becomes INACTIVE.

```
SQL> SELECT group#, status FROM v$log;

    GROUP# STATUS
---------- ----------------
  1 CURRENT
  2 INACTIVE
  3 INACTIVE
  4 UNUSED
  5 UNUSED
  6 UNUSED

6 rows selected.

SQL> ALTER DATABASE DROP LOGFILE GROUP 2;

Database altered.

SQL> ALTER DATABASE DROP LOGFILE GROUP 3;

Database altered.
```

- To drop an active group, perform multiple log switches until the group status becomes INACTIVE.

```
SQL> SELECT group#, status FROM v$log;

    GROUP# STATUS
---------- ----------------
  1 CURRENT
  4 UNUSED
  5 UNUSED
  6 UNUSED
```

```
SQL> ALTER SYSTEM SWITCH LOGFILE;

System altered.

SQL> select GROUP#, STATUS from v$log;

    GROUP# STATUS
---------- ----------------
  1 INACTIVE
  4 CURRENT
  5 UNUSED
  6 UNUSED

SQL> ALTER DATABASE DROP LOGFILE GROUP 1;

Database altered.
```

- Verify that the online redo log file migrated to the Automatic Storage Management (ASM). As per the query output below, there are two members for each group stored on the +DATA and +FRA disk groups. Multiplexing the online redo log groups avoids single point of failure.

```
SQL> SELECT member
  2  FROM   v$logfile;

MEMBER
--------------------------------------------------------------------------------
+DATA/t1c1/onlinelog/group_4.262.845583489
+FRA/t1c1/onlinelog/group_4.262.845583491
+DATA/t1c1/onlinelog/group_5.263.845583507
+FRA/t1c1/onlinelog/group_5.263.845583509
+DATA/t1c1/onlinelog/group_6.264.845583523
+FRA/t1c1/onlinelog/group_6.264.845583525

6 rows selected.

SQL>
```

This ends the migration of the database from the file system to the Automatic Storage Management (ASM). The ASM instance must remain operational to provide support for the database instance access of database files.

Verify New Database Storage

Verify that the database storage has been migrated to the Automatic Storage Management (ASM) files type. Use the SQL*Plus or ASMCMD command-line to confirm the database files are successfully migrated to the ASM.

```
SQL> SELECT file_name FROM dba_data_files UNION
  2  SELECT file_name FROM dba_temp_files UNION
  3  SELECT member    FROM v$logfile      UNION
```

```
  4  SELECT name      FROM v$controlfile;

FILE_NAME
--------------------------------------------------------------------------------
+DATA/t1c1/controlfile/current.256.845581127
+DATA/t1c1/datafile/sysaux.257.845581371
+DATA/t1c1/datafile/system.258.845581371
+DATA/t1c1/datafile/undotbs1.260.845581377
+DATA/t1c1/datafile/users.259.845581373
+DATA/t1c1/onlinelog/group_4.262.845583489
+DATA/t1c1/onlinelog/group_5.263.845583507
+DATA/t1c1/onlinelog/group_6.264.845583523
+DATA/t1c1/tempfile/temp.261.845582177
+FRA/t1c1/controlfile/current.257.845581127
+FRA/t1c1/onlinelog/group_4.262.845583491

FILE_NAME
--------------------------------------------------------------------------------
+FRA/t1c1/onlinelog/group_5.263.845583509
+FRA/t1c1/onlinelog/group_6.264.845583525

13 rows selected.

SQL>
```

DBMS_FILE_TRANSFER Package

Another approach to migrate and to copy files from the file system to the ASM is to use the Oracle supplied PL/SQL package DBMS_FILE_TRANSFER. The example below illustrates the use of the package to migrate a temporary tablespace from the file system to the Automatic Storage Management (ASM) using the following steps:

- Create and provide access to the source and target directories – SRCDIR and TGTDIR. The source data file is located on the directory '/home/oracle.' The target directory is the +DATA disk group.

```
SQL> CREATE DIRECTORY srcdir as '/home/oracle';

Directory created.

SQL> GRANT all ON DIRECTORY srcdir to system;

Grant succeeded.

SQL> CREATE DIRECTORY tgtdir as '+DATA';

Directory created.

SQL> GRANT all ON DIRECTORY tgtdir to system;

Grant succeeded.

SQL>
```

- Execute DBMS_FILE_TRANSFER.COPY_FILE procedure

```
SQL> BEGIN
  2     DBMS_FILE_TRANSFER.COPY_FILE(
  3          'SRCDIR', 'temp_1.dbf', 'TGTDIR', 'temp_1.f');
  4  END;
  5  /

PL/SQL procedure successfully completed.

SQL>
```

- Update the database data dictionary with the new file location. The tablespace must be either offline or the database is on MOUNT mode. Obtain the name of the copied file by querying V$ASM_FILES.

```
SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> STARTUP MOUNT
ORACLE instance started.

Total System Global Area       551165952 bytes
Fixed Size                       2230232 bytes
Variable Size                  352323624 bytes
Database Buffers               192937984 bytes
Redo Buffers                     3674112 bytes
Database mounted.
SQL> ALTER DATABASE RENAME FILE '/home/oracle/temp_1.dbf'
  2  TO '+DATA/temp_1.f';

Database altered.

SQL> ALTER DATABASE OPEN;

Database altered.

SQL>
```

- Verify the new location of the migrated temporary file.

```
SQL> SELECT tablespace_name, file_name
  2  FROM   dba_temp_files;

TABLESPACE_NAME      FILE_NAME
-------------------- --------------------------------------------------
TEMP_2               +DATA/t1c1/tempfile/copy_file.267.846281147
TEMP_1               +DATA/temp_1.f
TEMP                 +DATA/t1c1/tempfile/temp.263.846267555

SQL>
```

The next two steps are optional, but highly recommended to account for the disk space used by unneeded database files.

Delete Database Files

After migrating the database to the Automatic Storage Management (ASM), it is recommended to remove all previous backups and data file copies. Removing files must be executed using Oracle Recovery Manager (RMAN) and not manually. RMAN removes the files from the FRA. Update the control file and data dictionary.

```
$ rman

Recovery Manager: Release 11.2.0.3.0 - Production on Tue Apr 22 20:30:41 2014

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

RMAN> CONNECT TARGET

connected to target database: T1C1 (DBID=3821873595)

RMAN> RUN {
2> DELETE COPY OF DATABASE;
3> }

using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=42 device type=DISK
allocated channel: ORA_DISK_2
channel ORA_DISK_2: SID=47 device type=DISK
allocated channel: ORA_DISK_3
channel ORA_DISK_3: SID=45 device type=DISK
allocated channel: ORA_DISK_4
channel ORA_DISK_4: SID=43 device type=DISK
allocated channel: ORA_DISK_5
channel ORA_DISK_5: SID=44 device type=DISK
allocated channel: ORA_DISK_6
channel ORA_DISK_6: SID=49 device type=DISK
List of Datafile Copies
=======================

Key     File S Completion Time Ckp SCN    Ckp Time
------- ---- - --------------- ---------- ---------------
11      1    A 22-APR-14       5993451    22-APR-14
        Name: /u01/app/oracle/oradata/T1C1/system01.dbf

12      2    A 22-APR-14       5993451    22-APR-14
        Name: /u01/app/oracle/oradata/T1C1/sysaux01.dbf

13      3    A 22-APR-14       5993451    22-APR-14
        Name: /u01/app/oracle/oradata/T1C1/undotbs01.dbf

14      4    A 22-APR-14       5993451    22-APR-14
        Name: /u01/app/oracle/oradata/T1C1/users01.dbf

Do you really want to delete the above objects (enter YES or NO)? YES
deleted datafile copy
datafile copy file name=/u01/app/oracle/oradata/T1C1/system01.dbf RECID=11
STAMP=845581916
deleted datafile copy
```

```
datafile copy file name=/u01/app/oracle/oradata/T1C1/sysaux01.dbf RECID=12
STAMP=845581916
deleted datafile copy
datafile copy file name=/u01/app/oracle/oradata/T1C1/undotbs01.dbf RECID=13
STAMP=845581916
deleted datafile copy
datafile copy file name=/u01/app/oracle/oradata/T1C1/users01.dbf RECID=14
STAMP=845581916
Deleted 4 objects

RMAN> exit

Recovery Manager complete.
```

Removal of Unneeded Files

After all database data files, online redo log files, and control files are migrated to the ASM, remove the old files from the file system. Be careful when executing this step. Make sure your current directory is '/u01/app/oracle/oradata/T1C1.'

```
$ cd /u01/app/oracle/oradata/T1C1/
$ ls -al
total 167092
drwxr-xr-x 2 oracle oinstall      4096 Apr 22 20:31 .
drwxr-x--- 3 oracle oinstall      4096 Apr 18 10:04 ..
-rw-r----- 1 oracle oinstall  10076160 Apr 22 19:25 control01.ctl
-rw-r----- 1 oracle oinstall  52429312 Apr 22 20:20 redo01.log
-rw-r----- 1 oracle oinstall  52429312 Apr 22 20:12 redo02.log
-rw-r----- 1 oracle oinstall  52429312 Apr 22 20:12 redo03.log
-rw-r----- 1 oracle oinstall  49291264 Apr 22 19:21 temp01.dbf
-rw-r----- 1 oracle oinstall 209723392 Apr 22 19:55 temp02.dbf
$ rm *.log *.dbf *.ctl
$
```

Enable Block Change Tracking

This step is performed for testing purposes only. The block change tracking process should remain disabled during the next phase – the database upgrade.

```
$ sqlplus /nolog

SQL*Plus: Release 11.2.0.3.0 Production on Tue Apr 22 20:37:31 2014

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

SQL> CONN / AS SYSDBA
Connected.

SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;

Database altered.

SQL> ALTER DATABASE DISABLE BLOCK CHANGE TRACKING;
```

```
Database altered.

SQL>
```

Whole Database Backup

The database has been migrated from the file system to the Automatic Storage Management (ASM), before starting the Phase IV - Upgrade to Oracle Database 12c. Use Oracle Recovery Manager (RMAN) to perform a whole database backup. This backup is available in the event of failure during the database upgrade.

```
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP on;

new RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters are successfully stored

RMAN> BACKUP DATABASE PLUS ARCHIVELOG TAG="Before_Upgrade";
```

Upon successful backup, the next step is to upgrade the clone database from Oracle Database 11g to Oracle Database 12c as illustrated in Phase IV.

Phase IV: Upgrade to Oracle Database 12c

There are several methods to upgrade the clone database to Oracle Database 12c. Our aim is to achieve a near zero-downtime database upgrade by keeping the source database up and running throughout the upgrade process. The database upgrade is performed using the Oracle Database Upgrade Assistant (DBUA) or manually using SQL upgrade commands. Oracle Corporation recommends using the DBUA Graphical User Interface (GUI) as illustrated in this section. Figure 7-11 shows the clone database upgrade.

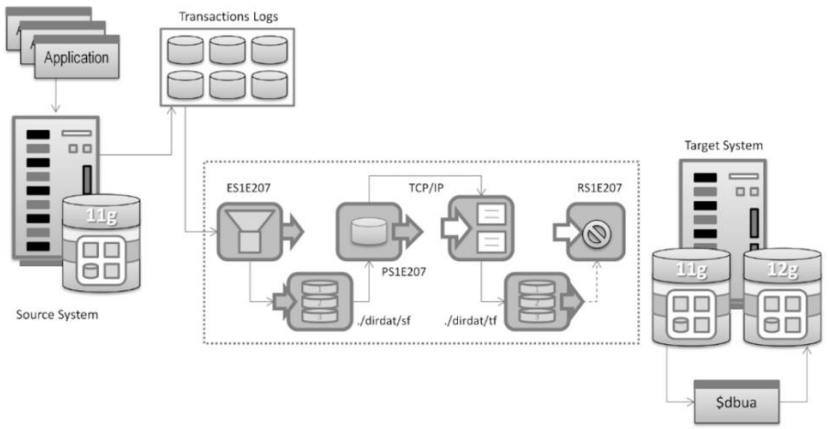


Figure 7-11: Clone Database Upgrade using the DBUA

Let us start by discussing the upgrade checklist. This ensures the database upgrade process is successfully executed.

Preparation Checklist

- Direct Upgrade Certification Path
- Delete the Database Recycle Bin
- Check for INVALID and Duplicate Objects in the SYS and SYSTEM Schemas
- Remove Outdated and Obsolete Parameters
- Check for INVALID Components
- Check for Mandatory Components
- Remove Obsolete Components
- Apply Patch Sets, Critical Patch Updates (CPUs), and Patches
- Minimum Disk Space Requirements
- Minimum Target Memory for SGA/PGA Settings

Direct Upgrade Certification Path

It is crucial to ascertain that the direct database upgrade is certified with Oracle Database 12c Release 1. For specific database releases, it may be required to perform an interim database upgrade to reach a certified source version for direct upgrade to Oracle Database 12c version 12.1.0.1.0. At the time of writing this book, the current direct upgrade path is shown in Table 7-1.

Source Database Release	Source Version	Direct Upgrade Support
Oracle Database 11g Release 2	11.2.0.2 and later	Yes
Oracle Database 11g Release 1	11.1.0.7	Yes
Oracle Database 10g Release 2	10.2.0.5	Yes

Table 7-1: Direct Upgrade Certification Path

Visit http://www.learn-ogg.com for the latest database upgrade certification path to Oracle Database 12c.

Delete the Database Recycle Bin

The recycle bin must be purged before starting the database upgrade. Use the command below to empty the database recycle bin for all users on the source database.

```
SQL> PURGE DBA_RECYCLEBIN;

DBA Recyclebin purged.
```

```
SQL>
```

Invalid and Duplicate Objects

Find out if there are invalid and duplicate objects owned by the SYS and SYSTEM. Find and fix invalid and duplicate objects as illustrated below.

```
SQL> SELECT DISTINCT object_name, object_type, owner
  2  FROM   dba_objects
  3  WHERE  status='INVALID'
  4  ORDER BY owner, object_name, object_type;

no rows selected

SQL>
```

Do not start the database upgrade when there are invalid data dictionary objects.
To recompile invalid objects, connect as SYSDBA and run the supplied script 'utlrp.sql' as shown below.

```
SQL> @?/rdbms/admin/utlrp.sql
```

Outdated Parameters

When Oracle develops a new database release, a parameter may become obsolete or depreciated. Obsolete parameters, underscore parameters and events must be removed prior to starting the database upgrade.
Obsolete parameters on Oracle Database 12c Release 1

- _app_ctx_vers
- _log_io_size

For a list of depreciated parameters, query the dynamic view V$PARAMETER as show below.

```
SQL> SELECT name
  2  FROM   v$parameter
  3  WHERE  isdeprecated ='TRUE'
  4  ORDER BY name;

NAME
--------------------------------------------------------------------------------
active_instance_count
background_dump_dest
buffer_pool_keep
buffer_pool_recycle
commit_write
cursor_space_for_time
fast_start_io_target
```

```
global_context_pool_size
instance_groups
lock_name_space
log_archive_local_first

NAME
--------------------------------------------------------------------------------
log_archive_start
max_enabled_roles
parallel_automatic_tuning
parallel_io_cap_enabled
parallel_server
parallel_server_instances
plsql_debug
plsql_v2_compatibility
remote_os_authent
resource_manager_cpu_allocation
sec_case_sensitive_logon

NAME
--------------------------------------------------------------------------------
serial_reuse
sql_trace
standby_archive_dest
user_dump_dest

26 rows selected.

SQL>
```

Invalid Components

Query DBA_REGISTRY to list the components' status. The status for all components must be valid before attempting the upgrade. Also, the DBUA pre-upgrade script reports installed components' status.

```
SQL> SELECT comp_name component, version, status
  2  FROM   dba_registry;

COMPONENT                                VERSION        STATUS
---------------------------------------- -------------  ------
OWB                                      11.2.0.3.0     VALID
Oracle Application Express               3.2.1.00.12    VALID
Oracle Enterprise Manager                11.2.0.3.0     VALID
OLAP Catalog                             11.2.0.3.0     VALID
Spatial                                  11.2.0.3.0     VALID
Oracle Multimedia                        11.2.0.3.0     VALID
Oracle XML Database                      11.2.0.3.0     VALID
Oracle Text                              11.2.0.3.0     VALID
Oracle Expression Filter                 11.2.0.3.0     VALID
Oracle Rules Manager                     11.2.0.3.0     VALID
Oracle Workspace Manager                 11.2.0.3.0     VALID
Oracle Database Catalog Views            11.2.0.3.0     VALID
Oracle Database Packages and Types       11.2.0.3.0     VALID
JServer JAVA Virtual Machine             11.2.0.3.0     VALID
Oracle XDK                               11.2.0.3.0     VALID
```

```
Oracle Database Java Packages             11.2.0.3.0     VALID
OLAP Analytic Workspace                   11.2.0.3.0     VALID
Oracle OLAP API                           11.2.0.3.0     VALID

18 rows selected.

SQL>
```

Mandatory Components

Oracle Database 12c Release 1 requires a few mandatory components. XDB (Oracle XML Database) is a mandatory component. It must be valid and upgraded successfully.

Obsolete Components

Oracle Database 12c Release 1 has made the following components obsolete:

- Enterprise Manager (EM) Database Control Repository
- UltraSearch

When performing upgrade using the DBUA, obsolete components are skipped. However, it is best to remove them from the clone database before starting the upgrade.

Database Patches

This is an imperative step and must be applied. The database patching is done to ensure having a certified source database version or to apply fixes necessary for a successful upgrade. Apply all patch sets, Patch Set Updates (PSUs) and patches to the clone database installation. By applying all patch types, it is likely that the database upgrade path will be confirmed as per the Oracle certification matrix.

Minimum Disk Space Requirements

The Database Upgrade Assistant (DBUA) reports the disk space usage and requirements. Ascertain how to meet the minimum recommended disk space for the ASM disk groups, flashback logs, and archived log files. The disk space requirements take into consideration the selected DBUA recovery option. For a clone database created from backup, it is not necessary to preserve any kind of DBUA option as the clone database may be recreated in case of a database upgrade failure and reattempt.

System Global Area Requirements

Allocate enough memory to the shared and Java pools. This is done by using the Automatic Shared Memory Management (ASMM) or manually. Failure to allocate the required memory causes the database upgrade to terminate with errors. During the upgrade phase, the data dictionary PL/SQL packages and Java classes are recompiled. However; some of the Java classes need to allocate enough memory to compile successfully. To avoid the upgrade terminates with failure due to unable to allocate memory allocation, set a value of 2GB or higher for the database parameter MEMORY_TARGET.

Now, let us perform the actual upgrade phases: Pre-Upgrade, Upgrade, and Post-Upgrade. Figure 7-12 presents the upgrade as a top-down approach highlighting the tasks for a successful database upgrade project.

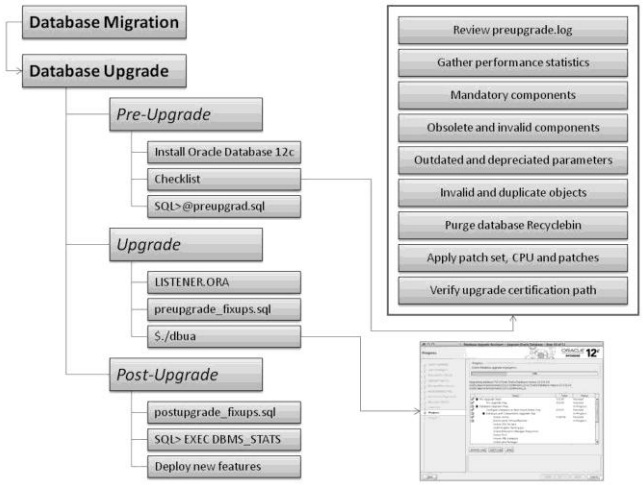


Figure 7-12: Database Upgrade Workflow

Database Pre-Upgrade

Before starting Oracle Database 12c Database Upgrade Assistant (DBUA), the pre-upgrade script (preupgrd.sql) must be executed. It quickly generates a pre-upgrade actions report. The pre-upgrade script is shipped with Oracle Database 12c and executed on the source database. It performs a sequence of checks against the database, but does not perform any action.

The pre-upgrade report includes the following sections:

- Renamed Parameters
- Obsolete and Depreciated Parameters
- Component List and Status
- Tablespace and their Space Usage
- Pre-Upgrade Checks and Required Actions
- Pre-Upgrade Recommendations

From the Oracle Database 12c software installation directory on the clone database server, follow the steps below to execute the pre-upgrade script.

```
$ cd /u01/app/oracle/product/12.1.0/db_1/rdbms/admin
$ sqlplus /nolog

SQL*Plus: Release 11.2.0.3.0 Production on Wed Apr 23 18:49:26 2014

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

SQL> CONN / AS SYSDBA
Connected.
SQL> @preupgrd.sql
Loading Pre-Upgrade Package...
Executing Pre-Upgrade Checks...
Pre-Upgrade Checks Complete.
      ************************************************************

Results of the checks are located at:
 /u01/app/oracle/cfgtoollogs/T1C1/preupgrade/preupgrade.log

Pre-Upgrade Fixup Script (run in source database environment):
 /u01/app/oracle/cfgtoollogs/T1C1/preupgrade/preupgrade_fixups.sql

Post-Upgrade Fixup Script (run shortly after upgrade):
 /u01/app/oracle/cfgtoollogs/T1C1/preupgrade/postupgrade_fixups.sql

      ************************************************************

      Fixup scripts must be reviewed prior to being executed.

      ************************************************************

      ************************************************************
               ====>> USER ACTION REQUIRED       <<====
      ************************************************************

 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.

      ************************************************************

SQL>
```

Since the clone database and Oracle Database 12c are on the same host, another listener (LISTENER2) is configured and started to support the upgraded database. Export the environment variables TNS_ADMIN and LD_LIBRARY_PATH as shown below.

```
$export TNS_ADMIN=/u01/app/oracle/product/12.1.0/db_1/network/admin
$cd /u01/app/oracle/product/12.1.0/db_1/bin
$./lsnrctl start LISTENER2
$export LD_LIBRARY_PATH=/u01/app/oracle/product/12.1.0/db_1/lib
```

Database Upgrade

The Near Zero-Downtime Database Upgrade requires the use of Oracle GoldenGate as part of the database upgrade technique. There are two methods for upgrading an Oracle Database:

- Oracle Database Upgrade Assistant
- Manual Upgrade using SQL

Oracle Corporation recommends using the Oracle Database Assistant (DBUA). The completion of the Database Pre-Upgrade Checklist ascertains the successful completion of the database upgrade using the Oracle Database Assistant (DBUA). The database upgrade timeframe is approximately 90-180 minutes regardless of the database size and it does not add, modify, or delete data.

Database Upgrade Assistant

The Oracle Database Upgrade Assistant (DBUA) is the Oracle recommended method for upgrading the database. There are two interface options to execute the DBUA:

- Graphical User Interface (GUI)
- Silent Interface

The Database Upgrade Assistant (DBUA) guides you through the upgrade steps by finding and implementing recommendations to ensure the database is successfully completed. The DBUA interface shows the progress of the database component and the completion status and simultaneously writes to the log files. DBUA alters and/or creates auxiliary files to support the upgraded database and provision database recoverability in the event of failure and fully supports the database running on Real Application Clusters. For a highly integrated and automated upgrade, the DBUA is available on silent mode from the command-line.

However, when the Database Administrator requires full control of the database upgrade runtime, the upgrade is done manually using SQL upgrade commands only. This method is more error prone when compared with the DBUA.

Starting Oracle Database Upgrade Assistant

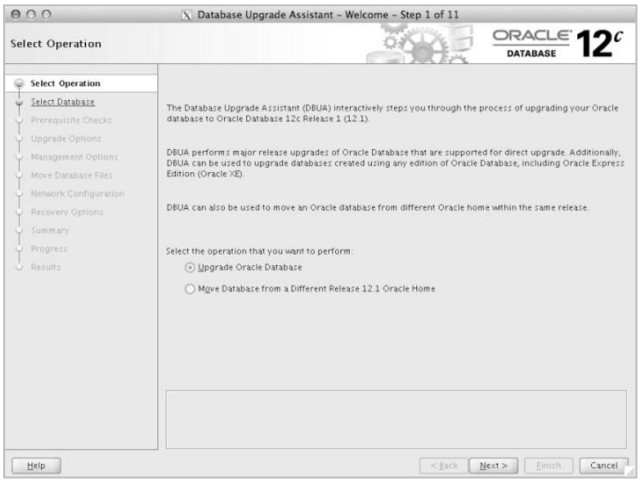
The Database Upgrade Assistant (DBUA) is started from Oracle Database 12c home by executing the following steps:

```
$ /u01/app/oracle/product/12.1.0/db_1/bin/dbua
```

Step 1 of 11: Select Operation

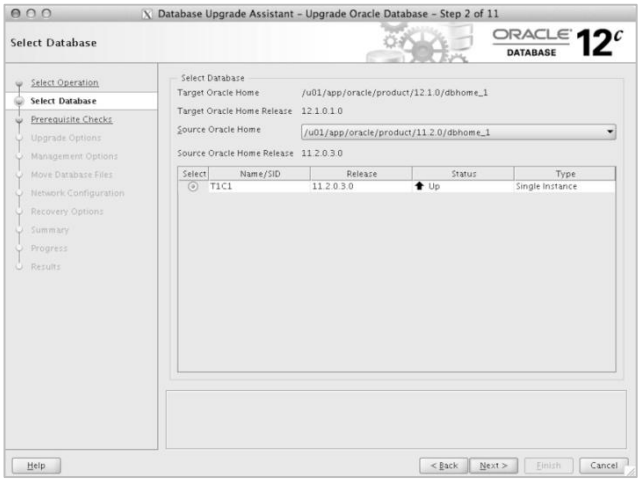
The Database Upgrade Assistant (DBUA) supports the following type of operations:

- Upgrade Oracle Database
- Merge Database from a Different Release 12.1 Oracle Home



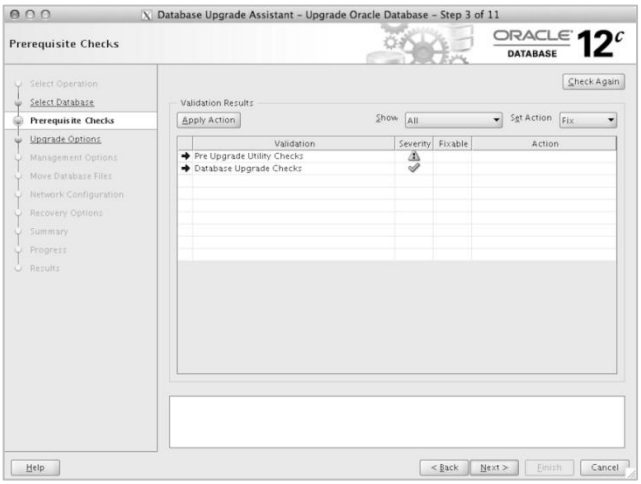
Step 2 of 11: Select Database

DBUA detects the source and target database software releases from file '/etc/oratab'. From the list of databases, select the database name for upgrade. The status of the database must be up.



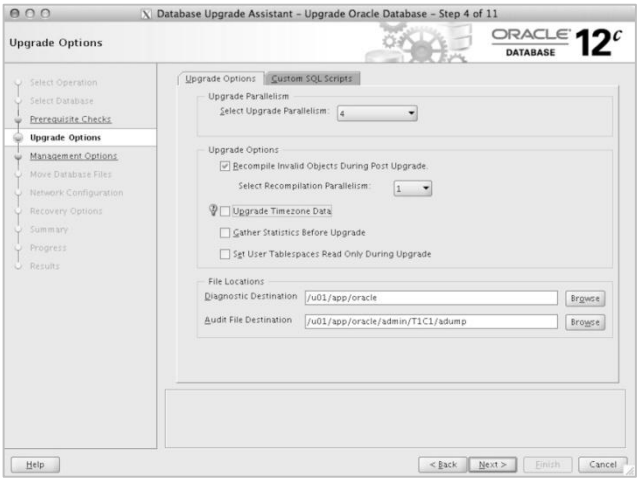
Step 3 of 11: Prerequisite Check

This step checks the prerequisite for the target database. Due to depreciated components and depreciated parameters, some alerts will be generated. Examine the alerts generated which may require taking actions before proceeding further.



Step 4 of 11: Upgrade Options

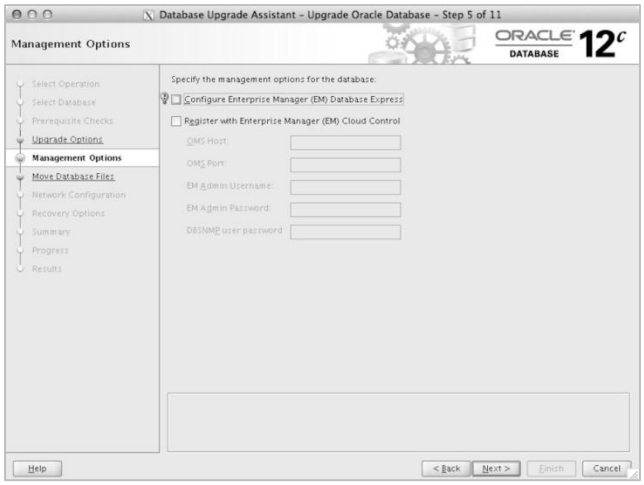
Select the database upgrade parallelism which is set to 4 by default. Parallelism enhances the database upgrade performance and reduces the downtime incurred during the database data dictionary upgrade. From the Upgrade Options, select 'Recompile Invalid Objects During Post Upgrade.' This eliminates the need of recompiling invalid objects manually by running the supplied script 'utlrp.sql'.



Step 5 of 11: Management Options

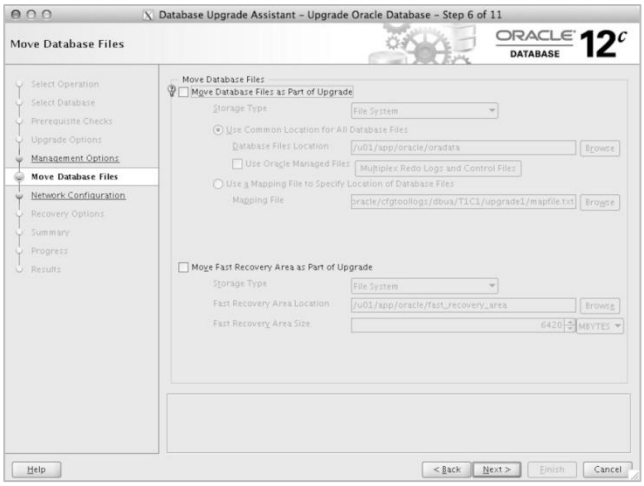
Oracle Enterprise Manager 11g Database Control is depreciated by Oracle Database 12c. There are two available management options:

- Enterprise Manager (EM) Database Express
- Enterprise Manager (EM) Cloud Control



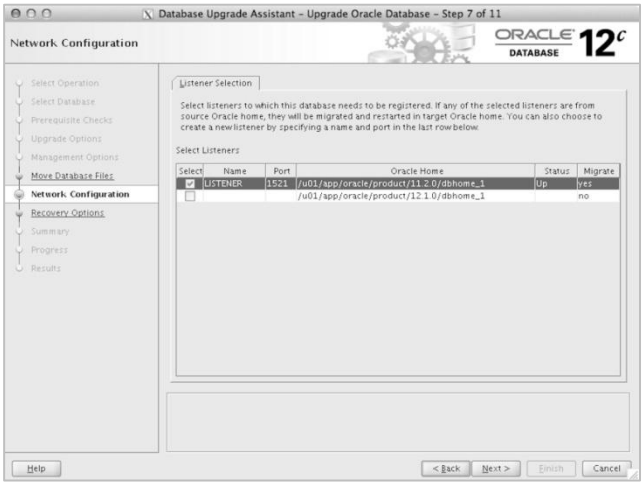
Step 6 of 11: Move Database Files

This enables you to move the database files and Fast Recovery Area (FRA) location. The storage types available are File System and Automatic Storage Management (ASM). For File System, you may select the Oracle Managed File (OMF) option.



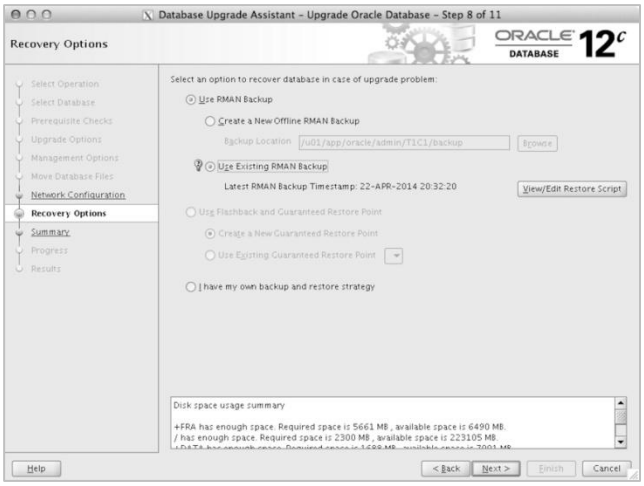
Step 7 of 11: Network Configuration

Select the listeners from the available list. The target database will be registered with the selected listeners. Or enter the listener name and the port number to be created by the DBUA.



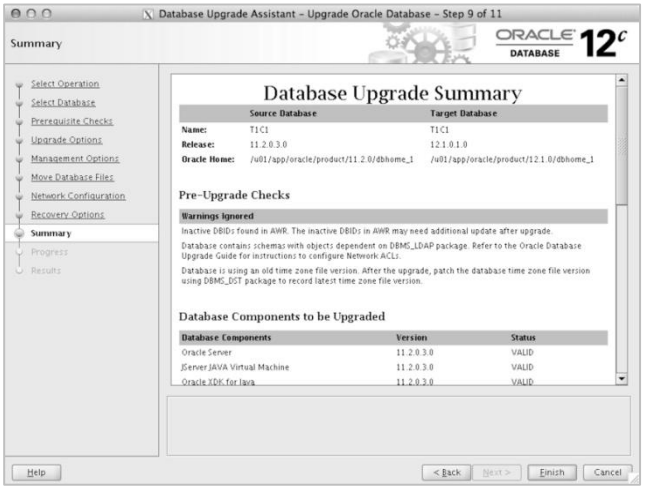
Step 8 of 11: Recovery Options

There are two recovery options: Use RMAN backup or Database Flashback. Since this is a clone database from an existing backup, select the option 'I have my own backup and restore strategy'. Consider the disk space requirements for +DATA, +FRA, flashback logs, and the archived log file. Disk space requirements must be met before proceeding further, if any.



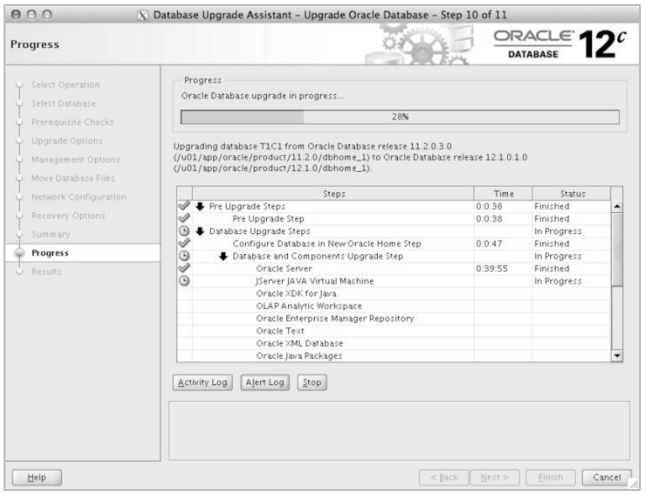
Step 9 of 11: Summary

Review the database upgrade summary sections. Verify the source and target database details – database name, release, and home. The Pre-Upgrade Checks section lists the issues found and the corresponding actions. The summary includes database parameters' old and changed values as recommended by the pre-upgrade script.
Click 'Finish' to begin the database upgrade.



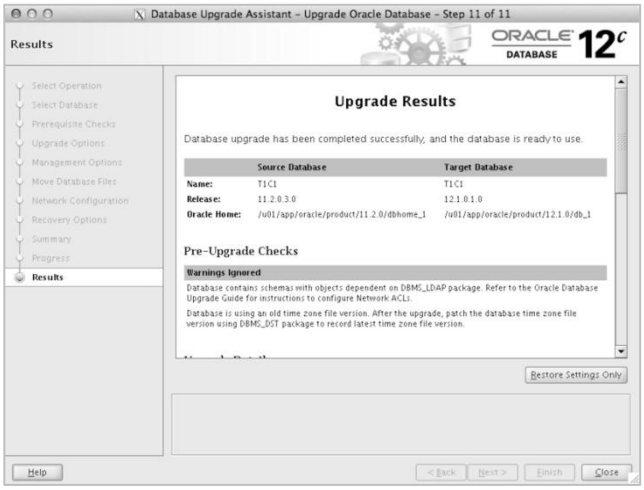
Step 10 of 11: Progress

Depending on the number of components to upgrade, the timeframe varies. An average of 90-120 minutes is common to complete the upgrade for all components.



Step 11 of 11: Upgrade Result

Review the upgrade results. For each step, DBUA generates a log file. Review the log files for recommendations as a post-upgrade step.



Verify Database Upgrade

Congratulations! The database upgrade is successfully completed. The target database instance is started and the database is opened. Query the dynamic view V$VERSION to confirm the database core components versions.

```
$ . oraenv
ORACLE_SID = [T1C1] ? T1C1
The Oracle base remains unchanged with value /u01/app/oracle
$ sqlplus /nolog

SQL*Plus: Release 12.1.0.1.0 Production on Sun Apr 27 19:36:50 2014

Copyright (c) 1982, 2013, Oracle.  All rights reserved.

SQL> CONN / AS SYSDBA
Connected.
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>
```

Silent Mode Upgrade

Another method of performing the database upgrade is running the DBUA on silent mode. The main reason for running the DBUA on silent mode is the absence of X Windows Manager. It also provides more control over the DBUA GUI runtime. The following script illustrates running the DBUA on silent mode:

```
$ /u01/app/oracle/product/12.1.0/db_1/bin/dbua -silent \
> -sid T1C1 \
> -oracleHome /u01/app/oracle/product/11.2.0.3/dbhome_1 \
> -diagnosticDest /u01/app/oracle \
> -autoextendFiles \
> -recompile_invalid_objects true \
> -degree_of_parallelism 2 \
> -upgradeTimezone \
> -emConfiguration NONE \
> -keepHiddenParams \
> -gatheringStatistics \
> -createGRP true \
> -upgrade_parallelism 2
```

The timeframe for the database upgrade using the DBUA silent mode is the same as the DBUA GUI mode. Performing the pre-upgrade checklist ensures database components are upgraded successfully.

Manual Upgrade

The manual upgrade provides more control when performing the database upgrade. The steps for performing a manual database upgrade using SQL command are highlighted below.

- On the source database, run the pre-upgrade script tool shipped with Oracle Database 12c. The script file is preupgrd.sql.
- Create a whole database backup using RMAN. This serves as the database recovery solution in the event of failure.
- The installation of Oracle Database 12c is already completed. Prepare the new Oracle Home database auxiliary files, parameters, password file and network files.
- Shutdown the source database.
- While still connected as SYSDBA, execute the SQL upgrade command STARTUP UPGRADE.
- Restart the database instance.
- Run the post-upgrade script 'utlu121s.sql'
- Run the post-upgrade script 'catupst.sql,' which performs the remaining database upgrade operations.

Database Post-Upgrade

The post-upgrade configures files related to Oracle Database 12c. When using Oracle Database Configuration Assistant (DBCA), the database post-upgrade changes are minimal. The following is the list of post-upgrade changes:

- Update the Operating System environment variables
- Update '/etc/oratab' for Manual Upgrade
- Update Client Files
- Upgrade Recovery Manager Catalog, if used
- Enable Database Vault, if used.
- Deploy Oracle Database 12c new features, gradually.
- Enable Unified Auditing

Enable Flashback Database

Using the Flashback Database quickly brings the database to an earlier point in time by undoing changes that occurred since that time. The advantage of the database Flashback is its fast operation compared to the traditional RMAN restore-recover operations to perform point-time database recovery.

The Flashback Database step supports rewinding the database after Phase V – Application Regression Testing is completed. The purpose is to reestablish the target database without the need to re-clone the database. Follow the steps below to configure and enable Flashback Database.

Though there are several Flashback Database options which are: Timestamp, System Change Number (SCN), Restore Point and Flashback Sequence Number, use of the SCN or restore point is preferred. Begin by obtaining the current database System Change Number (SCN). This will be used on the subsequent step to flashback the database.

```
SQL> SELECT current_scn
  2  FROM   v$database;

CURRENT_SCN
-----------
    9294723

SQL>
```

Enabling the Flashback Database starts a background process RVWR to write the Flashback logs to the Fast Recovery Area (FRA). The amount of Flashback is set to 129600 minutes (90 days) to target, but not to guarantee enough timeframe to complete

the applications' functional testing. The Fast Recovery Area (FRA) size is proportional to the retention target.

```
SQL> ALTER SYSTEM SET DB_FLASHBACK_RETENTION_TARGET=129600
  2  SCOPE=both;

System altered.

SQL> ALTER DATABASE FLASHBACK on;

Database altered.

SQL> SELECT flashback_on
  2  FROM   v$database;

FLASHBACK_ON
------------------
YES

SQL>
```

Setting the parameter DB_FLASHBACK_RETENTION_TARET is optional. It specifies an upper limit, in minutes, on how far back the database can be flashed back. The default value is 1440 minutes (1 day). When setting the value for the parameter DB_FLASHBACK_RETENTION_TARET, consider two important elements – the timeframe to complete the testing and the size of the Fast Recovery Area (FRA) to sustain the accumulative database flashback logs.

Keep monitoring the storage requirement and usage while the Flashback Databse is enabled. Use Oracle Enterprise Manager or perform queries against the database dynamic views:

V$FLASHBACK_DATABASE_LOG
V$FLASH_RECOVERY_AREA_USAGE

Phase V: Applications Regression Testing

Before the transition to the new system, functional and technical testing must be thoroughly conducted. The purpose of various types of testing is to uncover issues that may arise due to changes and enhancements to the database management system internals. A very common area to address is the behavior of the application with the new database optimizer. Oracle Corporation has addressed this issue using the Oracle Real Application Testing suite, which consists of two solutions:

- Oracle Database Replay
- SQL Performance Analyzer

Oracle Database Replay

Oracle Database Replay is the database upgrade companion tool. The Database Replay enables users to perform real-time, real-world testing by capturing actual database workloads on the production system and replaying them on the clone database system. The test on the clone database can be performed with production characteristics including timing and transactions concurrency. The main feature of Database Replay is it provides analysis and reporting to highlight potential problems such as slow completion of a report on the new environment. The Oracle Database Replay has two interfaces – Oracle Enterprise Manager and Oracle PL/SQL supplied packages. Figure 7-13 shows the Oracle Database Replay workflow, which is composed of three tasks:

- Capture Workload. Captures a workload from the production (source) environment. The capture is done during normal business hours to create a real-world baseline.
- Pre-Process Workload. This task prepares the capture files for replay on the target system. The workload capture files must be moved to the target system before the next task, when the pre-process is done on the source system.
- Replay Workload. This task replays the pre-processed workload on the clone database.

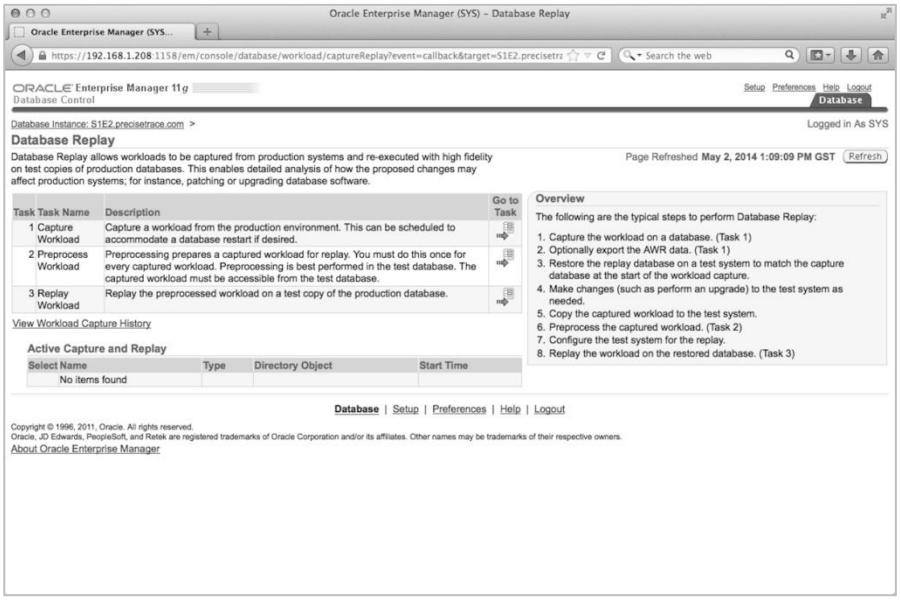


Figure 7-13: Oracle Database Replay

SQL Performance Analyzer

The SQL Performance Analyzer (SPA) can be used to quickly and accurately predict and prevent potential performance problems for any database changes that alter the original sequence of execution plan steps. The change can be any of the following:

- Database Upgrade
- Operating System Upgrade
- Database Migration
- Deploying New Application
- Database Schema Changes / Adding Indexes
- Statistics Gathering
- Database Parameter Changes
- Hardware and Operating System Changes

Figure 7-14 depicts the Oracle SQL Performance Analyzer (SPA). The workflow run is either a predetermined or guided workflow. Select the workflow that corresponds to the upgrade path or guided workflow.

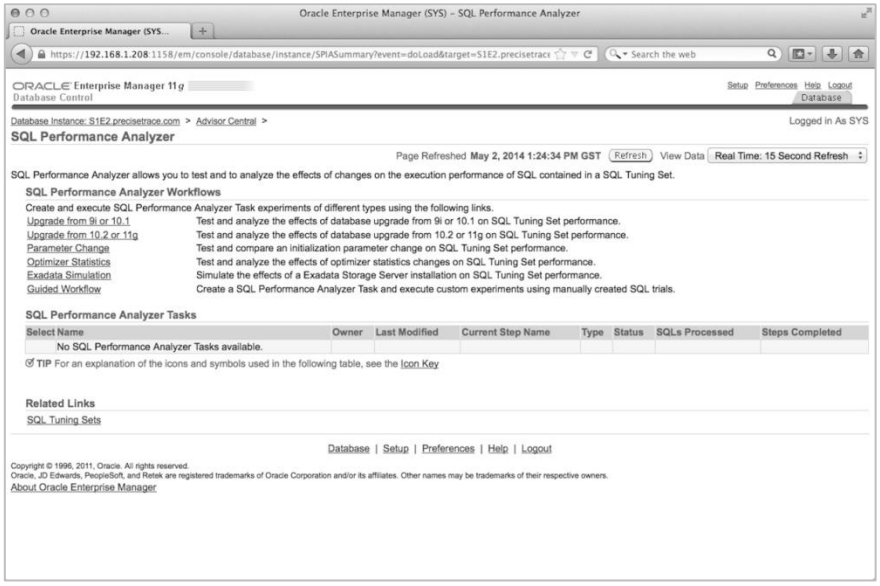


Fig 7-14: Oracle SQL Performance Analyzer (SPA)

This is an overview of Oracle Database Real Application Testing Suite solutions. Use either Oracle Database Replay or SQL Performance Analyzer. Phase V – Applications Regressions Testing is out of this book's scope, as our objective is limited to

implementation of using Oracle GoldenGate 12c for Near Zero-Downtime Migration and Upgrade of the Oracle Database only.

Perform Flashback Database

After completing the functional and technical testing, the database must be restored to the state just after the upgrade is completed. The following procedure illustrates performing the Flashback Database.

```
SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> STARTUP MOUNT
ORACLE instance started.

Total System Global Area   551165952 bytes
Fixed Size                   2230232 bytes
Variable Size              373295144 bytes
Database Buffers           171966464 bytes
Redo Buffers                 3674112 bytes
Database mounted.
SQL> FLASHBACK DATABASE TO SCN 9294723;
SQL> ALTER DATABASE OPEN RESETLOGS;

Database altered.

SQL>
```

Query the dynamic view V$SESSION_LONGOPS to monitor the progress of the Flashback Database. Another method is to monitor the instance alert file.

```
$ cd /u01/app/oracle/diag/rdbms/t1c1/T1C1/trace
$ tail -f alert_T1C1.log
Flashback Restore Start
Flashback Restore Complete
Flashback Media Recovery Start
Serial Media Recovery started
Recovery of Online Redo Log: Thread 1 Group 6 Seq 3 Reading mem 0
  Mem# 0: +DATA/t1c1/onlinelog/group_6.266.846267799
  Mem# 1: +FRA/t1c1/onlinelog/group_6.260.846267803
Incomplete Recovery applied until change 9319751 time 05/03/2014 07:23:44
Flashback Media Recovery Complete
Completed: flashback database to scn 9319750
```

Disable Flashback Database

Now, disable Flashback Database. The disable command terminates the Flashback background process RVWR and deletes existing Flashback logs stored at Fast Recovery Area (FRA).

```
SQL> ALTER DATABASE FLASHBACK off;

Database altered.

SQL> SELECT flashback_on FROM v$database;

FLASHBACK_ON
------------------
NO

SQL>
```

Phase VI: Apply Target Database Synchronization

Simultaneously, while the clone database is going through an upgrade to Oracle Database 12c, Oracle GoldenGate is running on the source system (S1E2). It is continuously capturing and sending transactions and trail files to the target system (T1C1). After successfully completing the database upgrade, the target database synchronization starts from the System Change Number (SCN) identified by the RMAN DUPLICATE command. This saves the Replicat processing time for reading, applying, and handling conflicts of unnecessary transactions. It is recommended that the target Oracle GoldenGate manager process is configured without the trail files purge option. Remote purge of trail files will be configured after the migration of users, when it is certain that the trail files are no longer needed. Figure 7-15 depicts the synchronization of the target database.

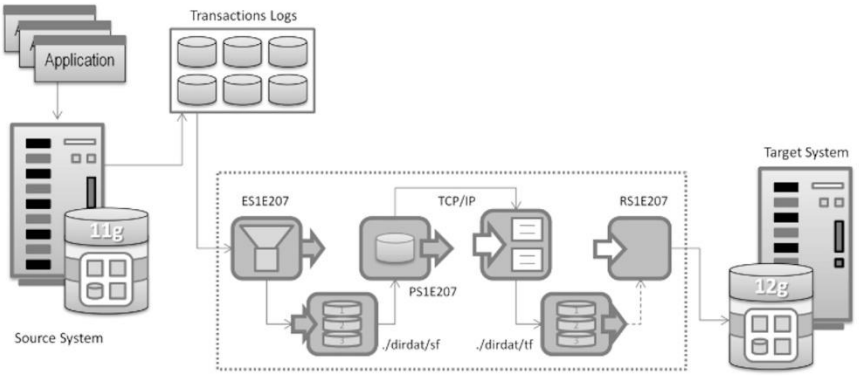


Figure 7-15: Synchronization of the Target Database

The Replicat process applies transactions to the target database in the order of the committed System Change Number (SCN), preserving data integrity of the target database while the applications continue to run on the source database. Implement parallel downstream Extract-Replicat processes carefully to avoid escalation of data integrity

issues. The list of tables must be only related to tables handled by the Extract-Replicat group.

Remote Trail Files

Remote trail files are shipped from the source to target system using the data pump process. They are accumulated and ultimately applied to synchronize the target database. The size of the remote trail is proportional to the source transactions volume and the timeframe to complete the database migration and upgrade. Verify and compare the trail files on the source and its availability on the target system.

```
$ cd /u01/app/ggs/12.1.0/gghome_1/dirdat
$ ls -ltr
total 395056
-rw-r----- 1 oracle oinstall   492190 Apr 11 11:24 tf000000
-rw-r----- 1 oracle oinstall 15999967 Apr 11 21:00 tf000001
-rw-r----- 1 oracle oinstall 15999941 Apr 12 06:37 tf000002
-rw-r----- 1 oracle oinstall 15999756 Apr 12 16:15 tf000003
-rw-r----- 1 oracle oinstall 15999820 Apr 14 21:09 tf000004
-rw-r----- 1 oracle oinstall 15999990 Apr 15 06:50 tf000005
-rw-r----- 1 oracle oinstall 15999742 Apr 15 16:32 tf000006
-rw-r----- 1 oracle oinstall 15999867 Apr 16 02:15 tf000007
-rw-r----- 1 oracle oinstall 15999954 Apr 16 11:59 tf000008
-rw-r----- 1 oracle oinstall 15999940 Apr 16 21:44 tf000009
-rw-r----- 1 oracle oinstall 15999999 Apr 17 07:30 tf000010
-rw-r----- 1 oracle oinstall 15999852 Apr 17 17:17 tf000011
-rw-r----- 1 oracle oinstall 15999867 Apr 18 03:05 tf000012
-rw-r----- 1 oracle oinstall 15999872 Apr 18 12:59 tf000013
-rw-r----- 1 oracle oinstall 15999870 Apr 18 22:51 tf000014
-rw-r----- 1 oracle oinstall 15999906 Apr 19 08:44 tf000015
-rw-r----- 1 oracle oinstall 15999879 Apr 19 18:36 tf000016
-rw-r----- 1 oracle oinstall 15999783 Apr 20 04:29 tf000017
-rw-r----- 1 oracle oinstall 15999984 Apr 20 14:24 tf000018
-rw-r----- 1 oracle oinstall 15999955 Apr 21 00:19 tf000019
-rw-r----- 1 oracle oinstall 15999811 Apr 21 10:16 tf000020
-rw-r----- 1 oracle oinstall 15999790 Apr 21 20:14 tf000021
-rw-r----- 1 oracle oinstall 15999876 Apr 22 06:13 tf000022
-rw-r----- 1 oracle oinstall 15999796 Apr 22 16:12 tf000023
-rw-r----- 1 oracle oinstall 15999887 Apr 23 02:13 tf000024
-rw-r----- 1 oracle oinstall 15999797 Apr 23 12:16 tf000025
-rw-r----- 1 oracle oinstall  3440714 Apr 23 14:25 tf000026
$
```

Checkpoint Database Table

Create the checkpoint table used by the Replicat process. The default checkpoint location is a database table as defined by the GLOBALS parameter files.

```
GGSCI (ggs-target) 9> DBLOGIN USERID ggs_admin, PASSWORD oracle
Successfully logged into database.

GGSCI (ggs-target) 10> ADD CHECKPOINTTABLE
```

```
No checkpoint table specified. Using GLOBALS specification
(ggs admin.chkpttab)...

Successfully created checkpoint table ggs_admin.chkpttab.

GGSCI (ggs-target) 11>
```

The Delivery Group

The delivery group is composed of the parameter file RS1E207.PRM, the Replicat process PS1E207, the remote trail ./dirdat/tf, and the checkpoint file ./dirchk/ES1E207.CPR. The delivery process uses the target database SQL native calls to apply transactions by reading the remote trail files.

Replicat Parameter File

Though the Replicat process starts from the System Change Number (SCN) identified by the RMAN DUPLICATE command, to prevent a duplicate records error message, the HANDLECOLISSIONS parameter is enabled.

```
GGSCI (apps-srv1) 3> EDIT PARAMS rs1e207
```

```
REPLICAT rs1e207
SETENV (ORACLE_SID="T1C1")
SETENV (ORACLE_HOME="/u01/app/oracle/product/12.1.0/db_1")
SETENV (NLS_LANG="AMERICAN_AMERICA.AL32UTF8")
USERID ggs_admin, PASSWORD oracle
ASSUMETARGETDEFS
HANDLECOLISSIONS
DISCARDFILE ./dirrpt/rs1e207.dsc, PURGE
MAPEXCLUDE osm$repapi.OSM_DOCUMENTS;
MAP osm$repapi.*, TARGET osm$repapi.*;
```

Define Replicat Process

The Replicat process reads from the ‘./dirdat/tf’ remote trail and applies transactions to the target database – T1C1.

```
GGSCI (ggs-target) 12> ADD REPLICAT rs1e207, EXTTRAIL ./dirdat/tf
REPLICAT added.

GGSCI (ggs-target) 13>
```

Remote Trail Settings

Verify the details of the remote trail. The Replicat process applies the remote trail files to the target database.

```
$ ./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) 2> info rmttrail ./dirdat/tf

       Extract Trail: ./dirdat/tf
             Extract: PS1E207
               Seqno: 1
                 RBA: 9045389
           File Size: 16M

GGSCI (ggs-source) 3>
```

The GGSCI INFO command shows the location, the Replicat name, the current sequence, the current RBA (Relative Byte Address), and the trail file size.

Start and Monitor Replicat Process

The Replicat is started after a specified System Change Number (SCN) using the option AFTERCSN of the START REPLICAT GGSCI command.

```
GGSCI (ggs-target) 16> INFO replicat RS1E207

REPLICAT   RS1E207   Initialized   2014-05-04 05:16   Status STOPPED
Checkpoint Lag       00:00:00 (updated 95:49:26 ago)
Log Read Checkpoint  File ./dirdat/tf000000
                     First Record  RBA 0


GGSCI (ggs-target) 17> START REPLICAT rs1e207 AFTERCSN 14274705

Sending START request to MANAGER ...
REPLICAT RS1E207 starting


GGSCI (ggs-target) 18> INFO ALL

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RS1E207     08:57:15      00:00:00

GGSCI (ggs-target) 19>
```

Perform quick basic test using SQL*Plus. This ascertains that the Replicat is applying transactions to the target database tables. Repeat to confirm rows count is changing.

```
SQL> CONN osm$repapi/oracle
Connected.
SQL> SELECT COUNT(*)
  2  FROM   customers;

  COUNT(*)
----------
   2553617

SQL>
```

At this point, the source and target database are synchronizing. The source transactions are applied in real-time to the target database achieving sub-seconds of latency. Now, the users migration to the new system occurs.

Phase VII: Data Verification

Prior to diverting applications from the source environment running Oracle Database 11g to the target environment running Oracle Database 12c, an important task must be performed and delivered with 100% accuracy - *Data Verification*. Data Verification compares data sets on the source and target databases and it is best performed using Oracle GoldenGate Veridata. Oracle GoldenGate Veridata assures data consistency without the need to rely on in-house developed SQL scripts or applications to compare table-to-table. This becomes a complex task for a very large database (VLDB) environment with a large number of tables. In such an environment, Oracle GoldenGate Veridata plays the major rule by automating the creation of compare pair groups using different methods such as wildcard patterns. This eliminates creating compare pairs explicitly.

Oracle GoldenGate Veridata

Oracle GoldenGate Veridata performs advanced data comparisons between compare pair groups of the source and target databases. The use of Oracle GoldenGate Veridata requires software installation and configuration of:

- Oracle GoldenGate Server Database Repository
- Oracle GoldenGate Server
- Oracle GoldenGate Veridata Agent (Java Agent) on the source and target systems

To perform comparison on data sets, Oracle GoldenGate Veridata requires the configuration of related components. Use the Oracle GoldenGate Veridata web-based administration tool to execute the following:

- Configure connections to source (S1E2) and target (T1C1) databases
- Configure groups
- Configure compare pair groups
- Configure compare jobs

The process of performing comparisons of data sets between the source and target databases is illustrated by presenting the following major steps:

- Compare Pair Groups Job Runtime
- Initial Compare
- Out-of-Sync Confirmation

For details, refer to Chapter 6 regarding implementing Oracle GoldenGate Veridata.

Compare Pair Groups Job Runtime

After configuring connections, compare groups, compare pair groups and compare jobs, the job is executed for the compare groups as shown in Figure 7-16. There are two compare groups: OSM_APPS for transactional compare pairs and OSM_LOOKUP for lookup compare pairs.

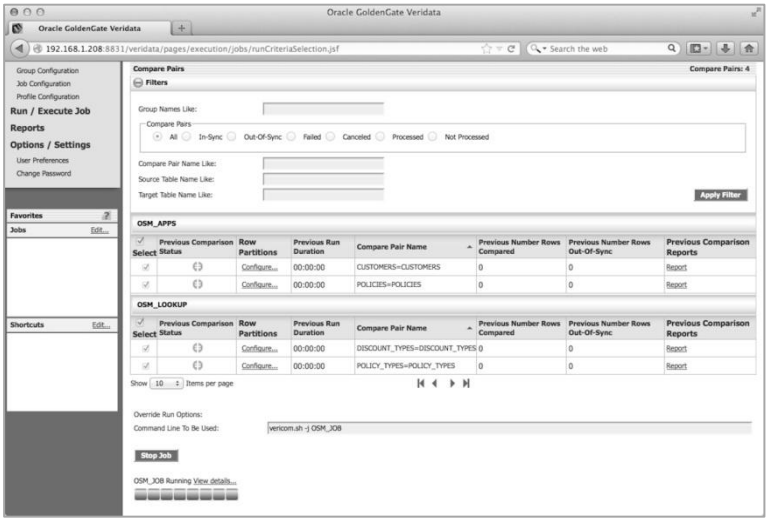


Figure 7-16: Compare Pair Group Job Runtime Progress

The compare groups and compare pairs configured are shown in Table 7-3.

Compare Group	Compare Pair
OSM_APPS	CUSTOMERS=CUSTOMERS POLICIES= POLICIES
OSM_LOOKUP	POLICY_TYPES= POLICY_TYPES DISCOUNT_TYPES= DISCOUNT_TYPES

Table 7-3: Compare Groups Configuration

Initial Compare

While Oracle GoldenGate is performing the Change Data Capture (CDC) on the source database — sending and applying the transactions to the target database — data deviation continues to exist. To ascertain accurate compare results,
the compare pair groups include all tables: transactional and lookup tables. Figure 7-17 shows the initial compare with out-of-sync results. This occurs because of the delay on applying the transactions to the target database.

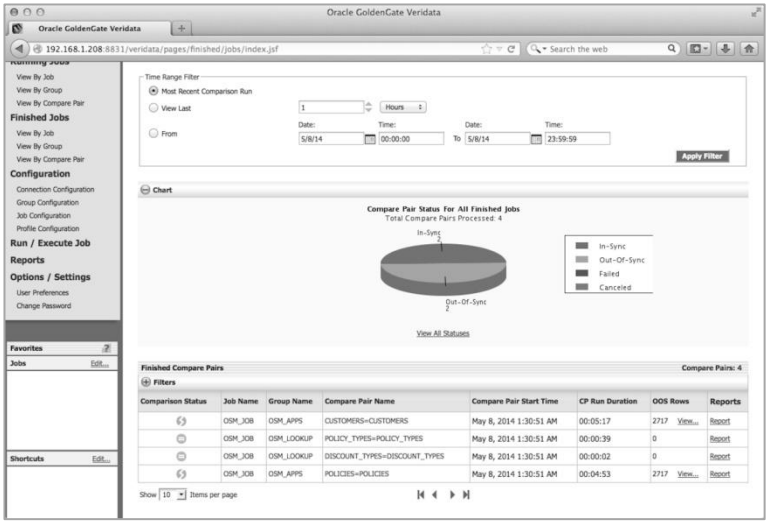


Figure 7-17: Initial Compare – Out of Sync

The OOS rows column indicates the extent of data mismatch, click on the View link to display the Out-of-Sync summary report. Out-of-Sync is expected to occur and
the extent depends on the current transactions volume of the source database.

Out-of-Sync Confirmation

For an enterprise running multiple application servers, the next step is to disconnect the application servers gradually during the minimum database workload.
The purpose of using this approach is to achieve sub-seconds latency between the source and target databases. Use Oracle GoldenGate GGSCI LAG command to determine the current lag of the Replicat process. When it comes within the expected upgrade downtime timeframe, it is time to disconnect the remaining application servers to allow data convergence. When the lag is zero, rerun the compare job to obtain the current Out-of-Sync result.

For our case, disconnecting the application server terminates the PL/SQL database application simulator by issuing the SQL KILL SESSION command.

```
SQL> SELECT s.sid, s.serial#
  2  FROM   v$session s, v$sqlarea a
  3  WHERE  s.username = 'OSM$REPAPI' AND
  4         s.sql_id = a.sql_id       AND
  5         UPPER(a.sql_text) like '%TRANS_LOAD%';

       SID    SERIAL#
---------- ----------
     19       1559

SQL> ALTER SYSTEM KILL SESSION '19, 1559' IMMEDIATE;

System altered.

SQL>
```

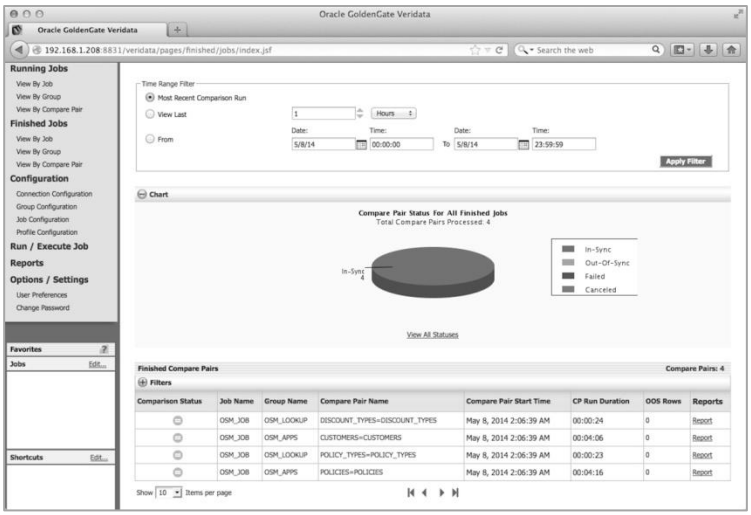


Fig 7-18: Synchronized Source and Target Databases

Figure 7-18 shows in-sync compare pair groups following termination of the PL/SQL transactions simulator. All transaction logs from the source database have been shipped and completely applied to the target database.

Achieving Near Zero-Downtime Optimization

The major factors impacting the database upgrade downtime are listed below. They are all related and listed in the order of higher-lower impact.

- Transactions volume on the source database during out-of-sync confirmation.

  High transactions volume affects the primary extracts capture from the transaction logs. Consider configuring parallelism of extracts, data pumps, and Replicat processes. Also, consider disabling applications access during low system activities to minimize the transaction logs.

- Extracts (primary and secondary) and Replicat lag.

  Within the downstream, there are three types of lags: The primary extract lag is the time difference of writing to the local trail and the transaction logs (online redo log file). The data pump extract lag is the time difference between writing to the remote trail and reading from the local trail. The Replicat lag is the time difference of applying transactions to the target database and reading from the remote trail. Fine tuning the lags reduces the downtime. Use the GGSCI LAG command to determine the current lag as shown below.

```
GGSCI (ggs-target) 5> LAG REPLICAT RS1E207

Sending GETLAG request to REPLICAT RS1E207 ...
No records yet processed.
At EOF, no more records to process.

GGSCI (ggs-target) 6>
```

- Application Servers Redirection Timeframe

  The redirection of Application Servers instances increases the downtime. An example is Oracle Application Express. The changes are made to DADS.CONF file to specify the new host name, listener port number, and SID. The preparation of redirection changes ahead of time and reduces the downtime as application servers are started immediately upon the confirmed synchronization of the source and target databases.

Removing Oracle GoldenGate Configurations

After the applications servers are diverted to the target system, perform the following tasks to remove Oracle GoldenGate instance configuration. However, do not de-install Oracle GoldenGate software and instance if there are more 24/7 databases to be upgraded. The following is the list of actions for removing the configurations:

- Stop, backup parameter files, and remove Extracts configurations on the source system.
- Stop the source instance manager.
- Delete the local trail files.
- Stop, backup parameter files, and remove Replicat configurations on the target system.
- Stop the target instance manager.
- Delete the remote trail files.
- Drop the checkpoint table.

Refer to Chapter 2 for details regarding Oracle GoldenGate 12c de-installation and removing configurations.

This concludes using Oracle GoldenGate for Near Zero-Downtime Migration and Upgrade. This chapter followed a phased approach which may vary by combining, adding, modifying, or eliminating specific tasks and/or phases. Ultimately, it is an outline for meeting the objective – Using Oracle GoldenGate for Near Zero-Downtime Migration and Upgrade.

Building Virtual File System for ASM Disks

Automatic Storage Management (ASM) employs provisioned disks for creating disk groups. The following steps instantiate the virtual disks used to create a virtual file system, without creating disks partitioning.

- Create zero-filled flat files

  Run the 'dd' command to create the zero-filled flat files. Optionally, in between each 'dd' command, the Operating System 'sync' command is executed to write buffered data to the disk, releasing memory for the next 'dd' command. The size of the file is specified using the 'count' attribute. The following example creates a zero-filled file of size 20GB.

  ```
  #dd if=/dev/zero of=/u01/asmdsks/_asm_file_vdisk_01 bs=1k count=20971520
  ```

- Attach a loopback device for each file

Use the Operating System utility 'losetup' to attach the first Linux loopback device with the flat file created. Ensure the loopback devices /dev/loop1 to /dev/loop7 are available. Test each loopback device availability; to test the first loopback device, type the following the command:

```
# /sbin/losetup /dev/loop1
```

- Link the virtual loopback device

  Use the command 'ln' to alias the loopback devices to be presented as virtual block devices.

- Create the ASM Disk

  The last step is to use Oracle ASM command 'oracleasm createdisk' to provision the files as virtual ASM disk.

Warning! This virtual disk setup is for training purposes. It should not be employed for a production environment. The script is designed to create up to 7 ASM disks only.

```
#----------------------------------------------------------------------
#-- DISCLAIMER:
#--
#-- This script is provided for educational purposes only.
#-- The script has been tested on an Oracle Databae 11g 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
echo `date`
echo "Enter non-existing ASM directory or press ENTER to exit, "
echo "Example: /u01/ASMDSKS: "; read vdir
if [ -d $vdir ]; then
   echo "Directory exit"; exit -1
elif [[ $vdir != /* ]]; then
   echo "Invalid directory, example /u01/ASMDSKS"; exit -1
else
  echo "Enter number of virtual disks followed by size, "
  echo "Example: 4 4194304: "; read vdsk vsize
  if [[ "$vdsk" = *[!2-7]* ]]; then
      echo "Invalid argument value, Example: 2 4096"; exit -1
  else
     mkdir -p $vdir
     for i in $(eval echo "{1..$vdsk}")
     do
        sync; echo 3 > /proc/sys/vm/drop_caches
        echo "Creating $vdir/_asm_file_vdisk_0$i"
```

```
        dd if=/dev/zero of=$vdir/_asm_file_vdisk_0$i\
           bs=1024 obs=1024 count=$vsize status=noxfer
        echo "De-attaching /dev/loop$i"
        /sbin/losetup -d /dev/loop$i
        /sbin/losetup /dev/loop$i $vdir/_asm_file_vdisk_0$i
        case "$i" in
           2) echo "Creating ASM_DISK0$i";
              ln -s /dev/loop$i /dev/xvdd;
              /usr/sbin/oracleasm createdisk ASM_DISK0$i /dev/xvdd;;
           3) echo "Creating ASM_DISK0$i";
              ln -s /dev/loop$i /dev/xvde;
              /usr/sbin/oracleasm createdisk ASM_DISK0$i /dev/xvde;;
           4) echo "Creating ASM_DISK0$i";
              ln -s /dev/loop$i /dev/xvdf;
              /usr/sbin/oracleasm createdisk ASM_DISK0$i /dev/xvdf;;
           5) echo "Creating ASM_DISK0$i";
              ln -s /dev/loop$i /dev/xvdg;
              /usr/sbin/oracleasm createdisk ASM_DISK0$i /dev/xvdg;;
           6) echo "Creating ASM_DISK0$i";
              ln -s /dev/loop$i /dev/xvdh;
              /usr/sbin/oracleasm createdisk ASM_DISK0$i /dev/xvdh;;
           7) echo "Creating ASM_DISK0$i";
              ln -s /dev/loop$i /dev/xvdi;
              /usr/sbin/oracleasm createdisk ASM_DISK0$i /dev/xvdi;;
        esac
     done
     echo "Scanning ASM disk..."
     echo ""
     /usr/sbin/oracleasm listdisks
  fi
fi
```

Summary

Employing Oracle GoldenGate for near zero-downtime migration and upgrade demonstrates a key application area of the software. Since Oracle GoldenGate is a multitasking software for enterprises running 24/7 systems, the software may be used to achieve near zero-downtime migration and upgrade while it is already installed and running on the system performing query offloading. This is an Oracle GoldenGate Return on Investment (ROI) strategy when deployed using best practices.

This chapter has demonstrated the migration and upgrade from Oracle Database 11g to Oracle Database 12c, the multitenant database, providing an innovative approach for database consolidation. The topics covered were:

- Deploy Oracle GoldenGate on Source and Target Databases
- Online Clone of the Database
- Migration to the Automatic Storage Management
- Upgrade from Oracle Database 11g to Oracle Database 12c
- Target Database Synchronization and Data Verification

Though the chapter has demonstrated the upgrade from Oracle Database 11g to Oracle Database 12c, when using Oracle GoldenGate, it enables a jumping upgrade approach. For example, direct upgrade from Oracle Database 9i to Oracle Database 12c. The jumping upgrade uses an initial-load from an active database rather than a database clone. Such an approach provides higher flexibility when compared with the traditional upgrade methods such as using Oracle Database Export and Import.

The chapter specified upgrading Oracle Database 11g to Oracle Database 12c using the DBUA Graphical User Interface (GUI) as it is the recommended technique for database upgrade. Other database upgrade techniques are DBUA on Silent Mode and Manual Upgrade using SQL commands. The DBUA Silent Mode provides a similar level of control as the DBUA GUI. However, a manual upgrade is executed step-by-step which promotes more control throughout the upgrade process.

It is critical to verify the data before the application transition occurs. Using Oracle GoldenGate Veridata delivers the best result in terms of accuracy and performance.

The next chapter covers Oracle GoldenGate heterogeneity between Oracle Database 12c and Oracle MySQL. This combination positions Oracle GoldenGate distinctively for data replication.

Oracle GoldenGate Advanced Integrations and Replications

APPENDIX

E

Overview

Enterprise Database Applications employ the database features for different purposes. Mission Critical Applications run on a deployed, clustered database environment using Oracle Real Application Clusters (RAC). For regulation and compliance, database applications connect to a primary database protected by one or more physical standby databases deployed using Oracle Data Guard (DG). Oracle GoldenGate highly supports these type of advanced environments.

The Oracle GoldenGate strategic direction has depreciated Oracle Streams. Customers are advised to migrate their data replication from Oracle Streams to Oracle GoldenGate. Oracle GoldenGate 11g release 2 for Oracle database supports of integrated capture for enabling downstream systems support and removing major limitations found in classic capture. For best scalability of large scale systems with massive transformation, intermediate systems promote data management scalability and eliminate overhead on the source and target systems.

The need to replicate Data Definition Language (DDL) is essential for dynamic in-house, developed database applications. Changes to the source database objects are replicated transparently — eliminating the need for the Database Administrators (DBAs) to manage actively changing database schemas.

This chapter covers advanced Oracle Database features requiring a high level of Oracle database administration skills. This chapter illustrates and explains the following:

- Oracle Data Guard (DG) Integration
- Migration from Oracle Streams to GoldenGate
- Integrated Capture for Oracle Database
- Intermediate Systems Support for Extreme Scalability
- Data Definition Language (DDL) Statements Replication

Although this chapter was developed using Oracle GoldenGate 12c for Oracle Database 12c, the same concepts and configuration parameters apply to Oracle Database 11g, with the exception of Oracle Database 12c Multi-Tenant.

Oracle Data Guard (DG) Integration

Oracle Data Guard is designed for deploying disaster recovery (DR) solutions. For some organizations, the availability of Data is very crucial for daily operations. Mirroring of Data using hardware is by far more costly and requires a greater level of experience to manage the different components. The Data Guard is an integrated component of the Oracle database and uses the Online Redo Log architecture to implement the "industry most reliable yet cost effective" disaster recovery solution for businesses. Depending on business requirements, Data Guard can be configured using different modes. Oracle Data Guard can be implemented with maximum protection, maximum availability, and maximum performance. A decision is made by evaluating Data availability and Data protection to decide on the Data Guard protection mode.

For systems requiring continuous availability, planned and unplanned downtimes are not an acceptable option. These systems need to be designed so that they are protected to be operational during either type of downtime situations to ensure the continuity of the business. Planned downtime occurs for a specific purpose during a specific time interval. Upgrading the system is a very common planned downtime. To minimize the downtime and the loss of data, production databases are protected by the complete infrastructure known as Oracle Data Guard. Unplanned downtime may occur as a result of instance failure, disk failure, data corruption, complete disaster, or human error. Based on the business requirement for high availability, it may be necessary to switch to a standby database to avoid such downtime intervals. Since failover to a standby database is a one-way operation and requires a rebuild afterwards, a decision has to be made whether to repair the production database or
to failover to the standby database. Depending on the type of failure, the proper recovery technique is used. For example, a user error can be fixed using "Flashback" without the need of a standby failover.

GoldenGate and Data Guard Architecture

One of the objectives for integrating an Oracle GoldenGate in a Data Guard environment is to spin off a read-write database instance used by other application systems. Figure 10-1 illustrates the installed components and the install locations of Oracle GoldenGate. The Oracle Database S1C1 is the primary database (source) which is protected by a physical standby database. The Data Guard broker performs an asynchronous redo transport to a maintained synchronized read-only remote database. Oracle GoldenGate captures from

the standby archived file and delivers to the remote database T1C1. This database is open on a read-write mode operating with the Flashback database enabled suitable for testing, reporting and other applications.

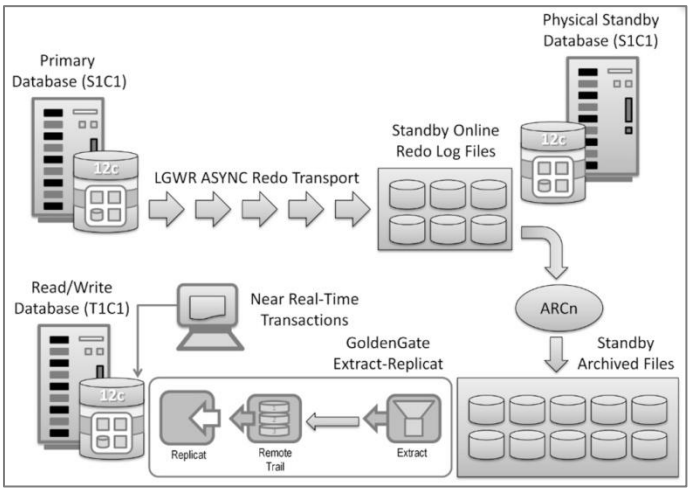


Figure 10-1: Oracle GoldenGate in a Data Guard Environment

Since the Extract ES1C101 is configured to operate on the "Archived Log Only" (ALO) mode, the target database remains behind the source database. When T1C1 is supporting non-critical application types, the capture is implemented to write directly to a remote trail – './dirdat/ta' rather than using a secondary capture (Data Pump). The Replicat RT1C101 applies the transactions to the target database using the native database interface. The use of a parallel Replicat is recommended to support heavy volume of archived redo log files. Refer to Chapter 1 for more details regarding implementing capture and delivery parallelism.

Creating Physical Standby Using RMAN

There are several methods for creating a physical standby database. The following sections illustrates the creation of a standby database using the Recovery Manager (RMAN). Use the following steps to create the physical standby database for the primary database S1C1.

- Enable and verify a database force logging option (Primary)
- Configure the standby redo log files (Primary)
- Configure archive log location parameters (Primary)
- Configure Data Guard Broker parameters (Primary)
- Enable database archive option (Primary)
- Perform an online backup using RMAN (Primary)

- Configure the Oracle SQLNET Network (Standby)
- Transfer the Oracle password file to the standby system
- Create the basic parameter file (Standby)
- Create the directories structure (Standby)
- Start the instance on the NOMOUNT state (Standby)
- Prepare and run the RMAN DUPLICATE command (Primary)
- Verify transactions' log shipping (Primary, Standby)
- Start the Data Guard Broker (Primary, Standby)
- Verify Data Guard Broker configuration (Primary, Standby)
- Verify database's Data Guard transport and application (Primary, Standby)

Enable and Verify Database Force Logging Option (Primary)

By default, the Oracle database force logging option is disabled at the database level while enabled at the tablespace and object level. However, to guarantee that the transaction logging of the primary database logging is overridden at the tablespace or object levels, it's required by the Data Guard to enable the database logging at the database level. This overrides no forced logging when specified at tablespace or object levels.

```
SQL> CONN / AS SYSDBA
Connected.
SQL> SELECT force_logging
  2  FROM   v$database;

FORCE_LOGGING
---------------------------------------
NO

SQL> ALTER DATABASE FORCE LOGGING;

Database altered.

SQL> SELECT force_logging
  2  FROM   v$database;

FORCE_LOGGING
---------------------------------------
YES

SQL>
```

Configure the Standby Redo Log Files (Primary)

When the database is operating on standby role, the received transaction logs are written to the standby redo log files and not to the online redo log files. Primarily, to avoids creating the standby redo log files manually on the standby database in the event of database role change. Because best practices suggest identical transactions log configuration, the online standby redo log files are created on the primary database and

copied over to the standby database using RMAN. To avoid contention on the standby redo log files, the number of standby redo log groups should be one more than the online redo log groups.

```
SQL> SELECT group#, bytes/1024/1024 "M"
  2  FROM   v$log;

    GROUP#          M
---------- ----------
         1         50
         2         50
         3         50

SQL> ALTER DATABASE ADD STANDBY LOGFILE
  2  '/u01/app/oracle/fast_recovery_area/S1C1/onlinelog/stdby01.log'
  3  SIZE 50M;

Database altered.

SQL> ALTER DATABASE ADD STANDBY LOGFILE
  2  '/u01/app/oracle/fast_recovery_area/S1C1/onlinelog/stdby02.log'
  3  SIZE 50M;

Database altered.

SQL> ALTER DATABASE ADD STANDBY LOGFILE
  2  '/u01/app/oracle/fast_recovery_area/S1C1/onlinelog/stdby03.log'
  3  SIZE 50M;

Database altered.

SQL> ALTER DATABASE ADD STANDBY LOGFILE
  2  '/u01/app/oracle/fast_recovery_area/S1C1/onlinelog/stdby04.log'
  3  SIZE 50M;

Database altered.

SQL>
```

Configure Archive Log Location Parameters (Primary)

Add the following parameters needed to operate a standby database. Best practices uses the Fast Recovery Area (FRA) as the local archived files location. The redo transport will be operating on Maximum Performance mode as it's configured using the parameter LOG_ARCHIVE_DEST_2.

Parameter	Description
LOG_ARCHIVE_DEST_1	The Archive Log File Destination
LOG_ARCHIVE_CONFIG	The List of Database Unique Names
LOG_ARCHIVE_DEST_2	The Mode, Valid Role, and Unique Names' Specification
DG_BROKER_START	The Enable Data Guard Command-Line

```
SQL> ALTER SYSTEM SET log_archive_dest_1 =
  2  'LOCATION=USE_DB_RECOVERY_FILE_DEST';

System altered.

SQL>
```

Configure Data Guard Broker Parameters (Primary)

The Data Guard uses the list of unique database names (S1C1 and T2C1) for managing the broker's configuration. The parameter LOG_ARCHIVE_CONFIG lists the unique names for the primary and standby databases. The parameter LOG_ARCHIVE_DEST_2 configures the redo transport, the protection mode, and the optional optimization parameters.

```
SQL> ALTER SYSTEM SET log_archive_config =
  2  'dg_config=(S1C1, T1C1)';

System altered.

SQL> ALTER SYSTEM SET
  2  log_archive_dest_2 = 'service=T1C1 ASYNC
  3  valid_for=(online_logfile,primary_role)
  4  db_unique_name=T1C1';

System altered.

SQL>
```

Enable Database Archive Option (Primary)

Enabling the database archive mode, the archived destination is the Fast Recovery Area (FRA) as it's configured using LOG_ARCHIVE_DEST_1.

```
SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> STARTUP MOUNT
ORACLE instance started.

Total System Global Area   413372416 bytes
Fixed Size                   2289016 bytes
Variable Size              331350664 bytes
Database Buffers            75497472 bytes
Redo Buffers                 4235264 bytes
Database mounted.
SQL> ALTER DATABASE ARCHIVELOG;

Database altered.

SQL> ALTER DATABASE OPEN;
```

```
Database altered.

SQL>
```

Perform an Online Backup Using RMAN (Primary)

Perform an online database backup which will be used by the RMAN DUPLICATE command to instantiate the standby database. Set the size of the FRA to support the backup of the database and archived log files while considering the retention policy of database & archived log files.

```
SQL> ALTER SYSTEM SET
  2  db_recovery_file_dest_size=10G;

System altered.

SQL> exit
Disconnected from Oracle Database 12c Enterprise Edition Release 12.1.0.1.0 -
64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing
options
[oracle@source1 ~]$ rman

Recovery Manager: Release 12.1.0.1.0 - Production on Tue Feb 24 15:18:17 2015

Copyright (c) 1982, 2013, Oracle and/or its affiliates.  All rights reserved.

RMAN> CONNECT TARGET

connected to target database: S1C1 (DBID=130281323)

RMAN> BACKUP DATABASE PLUS ARCHIVELOG;


Starting backup at 24-FEB-15
current log archived
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=40 device type=DISK
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=5 RECID=1 STAMP=872522132
input archived log thread=1 sequence=6 RECID=2 STAMP=872522350
channel ORA_DISK_1: starting piece 1 at 24-FEB-15
channel ORA_DISK_1: finished piece 1 at 24-FEB-15
piece
handle=/u01/app/oracle/fast_recovery_area/S1C1/backupset/2015_02_24/o1_mf_annnn_
TAG20150224T151912_bgrv20ro_.bkp tag=TAG20150224T151912 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 24-FEB-15

Starting backup at 24-FEB-15
using channel ORA_DISK_1
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
```

```
input datafile file number=00001
name=/u01/app/oracle/oradata/S1C1/datafile/o1_mf_system_bgqw43fp_.dbf
input datafile file number=00003
name=/u01/app/oracle/oradata/S1C1/datafile/o1_mf_sysaux_bgqw1d5w_.dbf
input datafile file number=00004
name=/u01/app/oracle/oradata/S1C1/datafile/o1_mf_undotbs1_bgqw74ps_.dbf
input datafile file number=00006
name=/u01/app/oracle/oradata/S1C1/datafile/o1_mf_users_bgqw73dz_.dbf
channel ORA_DISK_1: starting piece 1 at 24-FEB-15
channel ORA_DISK_1: finished piece 1 at 24-FEB-15
piece
handle=/u01/app/oracle/fast_recovery_area/S1C1/backupset/2015_02_24/o1_mf_nnndf_
TAG20150224T151914_bgrv22dc_.bkp tag=TAG20150224T151914 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:01:37
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
including current control file in backup set
including current SPFILE in backup set
channel ORA_DISK_1: starting piece 1 at 24-FEB-15
channel ORA_DISK_1: finished piece 1 at 24-FEB-15
piece
handle=/u01/app/oracle/fast_recovery_area/S1C1/backupset/2015_02_24/o1_mf_ncsnf_
TAG20150224T151914_bgrv54wb_.bkp tag=TAG20150224T151914 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 24-FEB-15

Starting backup at 24-FEB-15
current log archived
using channel ORA_DISK_1
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=7 RECID=3 STAMP=872522454
channel ORA_DISK_1: starting piece 1 at 24-FEB-15
channel ORA_DISK_1: finished piece 1 at 24-FEB-15
piece
handle=/u01/app/oracle/fast_recovery_area/S1C1/backupset/2015_02_24/o1_mf_annnn_
TAG20150224T152054_bgrv56s9_.bkp tag=TAG20150224T152054 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time: 00:00:01
Finished backup at 24-FEB-15
RMAN>
```

Configure Oracle SQLNET Network (Standby)

Configure the standby database listener and bi-directional connection for SQL NET. Test connections using the TNSPING utility before proceeding further. Perform configuration and test it, bi-directionally.

```
[oracle@target1 ~]$ cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1           localhost.localdomain localhost
::1         localhost6.localdomain6 localhost6
192.168.1.152       target1.precisetrace.com
192.168.1.140       source1.precisetrace.com
[oracle@target1 ~]$ tnsping S1C1
```

```
TNS Ping Utility for Linux: Version 12.1.0.1.0 - Production on 23-FEB-2015
16:47:09

Copyright (c) 1997, 2013, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = source1.precisetrace.com)(PORT = 1521))) (CONNECT_DATA =
(SERVICE_NAME = S1C1.precisetrace.com)))
OK (0 msec)
[oracle@target1 ~]$ tnsping T1C1

TNS Ping Utility for Linux: Version 12.1.0.1.0 - Production on 23-FEB-2015
16:47:19

Copyright (c) 1997, 2013, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST = (ADDRESS = (PROTOCOL =
TCP)(HOST = target1.precisetrace.com)(PORT = 1521))) (CONNECT_DATA =
(SERVICE_NAME = T1C1.precisetrace.com)))
OK (0 msec)
[oracle@target1 ~]$
```

Transfer the Oracle Password File to the Standby System

The transport services use the password to authenticate the SYS user remote using the password file. Transfer the password file of the primary database to the standby database. The change of permission setting may be necessary.

```
[root@source1 dbs]# scp orapwS1C1
root@target1.precisetrace.com:/u01/app/oracle/product/12.1.0/dbhome_1/dbs
root@target1.precisetrace.com's password:
orapwS1C1                                     100% 7680     7.5KB/s   00:00
[root@target1 dbs]# chown oracle:oinstall orapwS1C1
[root@target1 dbs]#
```

Create the Basic Parameter File (Standby)

Prepare the standby database parameter file. It will contain the database name parameter only - DB_NAME=T1C1. Other parameters are applied by the RMAN DUPLICATE command. The list of related standby parameters are:

- db_unique_name
- db_create_file_dest
- db_recovery_file_dest

- db_recovery_file_dest_size
- control_files
- log_archive_max_processes
- fal_client
- fal_server
- standby_file_management

Create the Directories Structure (Standby)

Create the directory structure with proper permissions to avoid read/write errors when running the RMAN DUPLICATE command. Login as oracle and verify that the umask environment is set to 022.

```
[oracle@target1 ~]$ cd /u01/app/oracle/admin/
[oracle@target1 admin]$ mkdir T1C1
[oracle@target1 admin]$ cd T1C1
[oracle@target1 T1C1]$ mkdir adump dpdump pfile
[oracle@target1 T1C1]$
```

Start the Instance on NOMOUNT State (Standby)

Use the parameter file to start the standby instance on NOMOUNT state. This allows the DUPLICATE command to create the standby control file as specified by the parameter CONTROL_FILES.

```
SQL> STARTUP NOMOUNT PFILE='$ORACLE_HOME/dbs/initT1C1.ora'
ORACLE instance started.

Total System Global Area   217157632 bytes
Fixed Size                   2286656 bytes
Variable Size              159386560 bytes
Database Buffers            50331648 bytes
Redo Buffers                 5152768 bytes
SQL>
```

Prepare and Run the RMAN DUPLICATE Command (Primary)

Prepare and run the RMAN DUPLCIATE command. Before attempting to run the command, check the syntax using the RMAN command 'checksyntax' as illustrated below. There are several methods to run the RMAN DUPLICATE command. The command below is executed using an active database, which absolutely eliminates database downtime.

```
[oracle@source1 dbs]$ rman

Recovery Manager: Release 12.1.0.1.0 - Production on Tue Mar 3 05:25:19 2015
```

```
Copyright (c) 1982, 2013, Oracle and/or its affiliates.  All rights reserved.

RMAN> connect target sys/oracle_4U@S1C1

connected to target database: S1C1 (DBID=130281323)

RMAN> connect auxiliary sys/oracle_4U@T1C1

connected to auxiliary database: T1C1 (not mounted)

RMAN> @rmandup.txt

run {
allocate channel s1c11 type disk;
allocate channel s1c12 type disk;
allocate channel s1c13 type disk;
allocate channel s1c14 type disk;
allocate auxiliary channel t1c11 type disk;
duplicate target database for standby from active database NOFILENAMECHECK
spfile
parameter_value_convert 'S1C1','T1C1'
set db_unique_name='T1C1'
set db_create_file_dest='/u01/app/oracle/oradata/T1C1/datafile'
set db_file_name_convert='/u01/app/oracle/oradata/S1C1/datafile',
'/u01/app/oracle/oradata/T1C1/datafile'
setlog_file_name_convert='/u01/app/oracle/fast_recovery_area/S1C1/onlinelog',
'/u01/app/oracle/fast_recovery_area/T1C1/onlinelog'
set db_recovery_file_dest='/u01/app/oracle/fast_recovery_area'
set db_recovery_file_dest_size='8G'
set control_files='/u01/app/oracle/oradata/T1C1/datafile/controlfile'
set log_archive_max_processes='5'
set fal_server='S1C1'
set fal_client='T1C1'
set standby_file_management='AUTO'
set log_archive_config='dg_config=(S1C1,T1C1)'
set log_archive_dest_1='LOCATION=/u01/app/oracle/archivelog
valid_for=(all_logfiles, all_roles) db_unique_name=S1C1'
set log_archive_dest_2='service=T1C1 LGWR ASYNC
valid_for=(ONLINE_LOGFILES,PRIMARY_ROLES) db_unique_name=T1C1'
set log_archive_dest_state_1='enable'
set log_archive_dest_state_2='enable';
}
using target database control file instead of recovery catalog
allocated channel: s1c11
channel s1c11: SID=79 device type=DISK

allocated channel: s1c12
channel s1c12: SID=77 device type=DISK

allocated channel: s1c13
channel s1c13: SID=67 device type=DISK

allocated channel: s1c14
channel s1c14: SID=73 device type=DISK

allocated channel: t1c11
channel t1c11: SID=21 device type=DISK

Starting Duplicate Db at 03-MAR-15
```

```
contents of Memory Script:
{
   backup as copy reuse
   targetfile  '/u01/app/oracle/product/12.1.0/dbhome_1/dbs/orapwS1C1' auxiliary
format
 '/u01/app/oracle/product/12.1.0/dbhome_1/dbs/orapwT1C1'   targetfile
 '/u01/app/oracle/product/12.1.0/dbhome_1/dbs/spfileS1C1.ora' auxiliary format
 '/u01/app/oracle/product/12.1.0/dbhome_1/dbs/spfileT1C1.ora'   ;
   sql clone "alter system set spfile=
''/u01/app/oracle/product/12.1.0/dbhome_1/dbs/spfileT1C1.ora''";
}
executing Memory Script

Starting backup at 03-MAR-15
Finished backup at 03-MAR-15
…
…
executing Memory Script

datafile 1 switched to datafile copy
input datafile copy RECID=1 STAMP=873264744 file
name=/u01/app/oracle/oradata/T1C1/datafile/o1_mf_system_bgqw43fp_.dbf
datafile 3 switched to datafile copy
input datafile copy RECID=2 STAMP=873264744 file
name=/u01/app/oracle/oradata/T1C1/datafile/o1_mf_sysaux_bgqw1d5w_.dbf
datafile 4 switched to datafile copy
input datafile copy RECID=3 STAMP=873264744 file
name=/u01/app/oracle/oradata/T1C1/datafile/o1_mf_undotbs1_bgqw74ps_.dbf
datafile 6 switched to datafile copy
input datafile copy RECID=4 STAMP=873264744 file
name=/u01/app/oracle/oradata/T1C1/datafile/o1_mf_users_bgqw73dz_.dbf
Finished Duplicate Db at 03-MAR-15
released channel: s1c11
released channel: s1c12
released channel: s1c13
released channel: s1c14
released channel: t1c11

RMAN> **end-of-file**

RMAN>
```

Login to the Standby database and verify the database state and role.

```
SQL> CONN sys/oracle_4U AS SYSDBA
Connected.
SQL> SELECT open_mode, database_role
  2  FROM   v$database;

OPEN_MODE            DATABASE_ROLE
-------------------- ----------------
MOUNTED              PHYSICAL STANDBY

SQL>
```

Verify Transactions Log Shipping (Primary, Standby)

Perform a manual online redo log file switch command. This verifies that the database is shipping the archived redo log files to the standby database system. Also, verify the archived file shipping and application are consistent.

```
SQL> CONN sys/oracle_4U@S1C1 AS SYSDBA
Connected.
SQL> SELECT MAX(sequence#)
  2  FROM   v$archived_log;

MAX(SEQUENCE#)
--------------
            66

SQL> CONN sys/oracle_4U@T1C1 AS SYSDBA
Connected.
SQL> SELECT MAX(sequence#)
  2  FROM   v$archived_log;

MAX(SEQUENCE#)
--------------
            66

SQL>
```

Start and Configure the Data Guard Broker (Primary or Standby)

Enable the Data Guard Broker for the primary & standby databases by setting the parameter DG_BROKER_START=TRUE.

```
SQL> ALTER SYSTEM SET dg_broker_start=TRUE SCOPE=BOTH;

System altered.

SQL>

DGMGRL> CREATE CONFIGURATION 'DGENV' AS
> PRIMARY DATABASE IS 'S1C1'
> CONNECT IDENTIFIER IS 'S1C1';
Configuration "DGENV" created with primary database "S1C1"
DGMGRL> ADD DATABASE 'T1C1' AS
> CONNECT IDENTIFIER IS 'T1C1';
Database "T1C1" added
DGMGRL> ENABLE CONFIGURATION;
Enabled.
```

Verify Data Guard Broker Configuration (Primary, Standby)

Perform this step from the primary & standby Data Guard instance. It will list the current primary & physical standby database. Ensure the status is 'SUCCESS' before proceeding

to the next step. In case of warning due to inconsistent parameter settings, use the 'EDIT Data Guard' command to synchronize the settings. Review the Data Guard parameters settings as shown below.

```
DGMGRL> SHOW CONFIGURATION;

Configuration - DGENV

  Protection Mode: MaxPerformance
  Databases:
  S1C1 - Primary database
    T1C1 - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

DGMGRL> SHOW DATABASE VERBOSE 'S1C1';

Database - S1C1

  Role:              PRIMARY
  Intended State:    TRANSPORT-ON
  Instance(s):
    S1C1

  Properties:
    DGConnectIdentifier             = 'S1C1'
    ObserverConnectIdentifier       = ''
    LogXptMode                      = 'ASYNC'
    RedoRoutes                      = ''
    DelayMins                       = '0'
    Binding                         = 'optional'
    MaxFailure                      = '0'
    MaxConnections                  = '1'
    ReopenSecs                      = '300'
    NetTimeout                      = '30'
    RedoCompression                 = 'DISABLE'
    LogShipping                     = 'ON'
    PreferredApplyInstance          = ''
    ApplyInstanceTimeout            = '0'
    ApplyLagThreshold               = '0'
    TransportLagThreshold           = '0'
    TransportDisconnectedThreshold  = '30'
    ApplyParallel                   = 'AUTO'
    StandbyFileManagement           = 'MANUAL'
    ArchiveLagTarget                = '0'
    LogArchiveMaxProcesses          = '4'
    LogArchiveMinSucceedDest        = '1'
    DbFileNameConvert               = ''
    LogFileNameConvert              = ''
    FastStartFailoverTarget         = ''
    InconsistentProperties          = '(monitor)'
    InconsistentLogXptProps         = '(monitor)'
    SendQEntries                    = '(monitor)'
    LogXptStatus                    = '(monitor)'
    RecvQEntries                    = '(monitor)'
```

```
    StaticConnectIdentifier         =
'(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=source1.precisetrace.com)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=S1C1_DGMGRL.oracle-
ggs.com)(INSTANCE_NAME=S1C1)(SERVER=DEDICATED)))'
    StandbyArchiveLocation          = '/u01/app/oracle/archivelog'
    AlternateLocation               = ''
    LogArchiveTrace                 = '0'
    LogArchiveFormat                = '%t_%s_%r.dbf'
    TopWaitEvents                   = '(monitor)'

Database Status:
SUCCESS

DGMGRL>
```

Test the Switchover Using the Data Guard (Primary, Standby)

Now, test the Switchover command. Perform the command from the primary database to perform a database role change, followed by a switch from the original standby (current primary) to reverse the database role.

```
DGMGRL> SWITCHOVER TO 'T1C1';
Performing switchover NOW, please wait...
Operation requires a connection to instance "T1C1" on database "T1C1"
Connecting to instance "T1C1"...
Connected as SYSDBA.
New primary database "T1C1" is opening...
Operation requires startup of instance "S1C1" on database "S1C1"
Starting instance "S1C1"...
ORACLE instance started.
Database mounted.
Switchover succeeded, new primary is "T1C1"
DGMGRL> CONNECT sys/oracle_4U
Connected as SYSDG.
DGMGRL> SHOW CONFIGURATION;

Configuration - DGConfig

  Protection Mode: MaxPerformance
  Databases:
  T1C1 - Primary database
    S1C1 - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

DGMGRL> SWITCHOVER TO 'S1C1';
Performing switchover NOW, please wait...
Operation requires a connection to instance "S1C1" on database "S1C1"
Connecting to instance "S1C1"...
Connected as SYSDBA.
New primary database "S1C1" is opening...
Operation requires startup of instance "T1C1" on database "T1C1"
Starting instance "T1C1"...
```

```
ORACLE instance started.
Database mounted.
Switchover succeeded, new primary is "S1C1"
DGMGRL> SHOW CONFIGURATION;

Configuration - DGConfig

  Protection Mode: MaxPerformance
  Databases:
  S1C1 - Primary database
    T1C1 - Physical standby database

Fast-Start Failover: DISABLED

Configuration Status:
SUCCESS

DGMGRL>
```

Configure Oracle GoldenGate for the Standby Database

The next step is to configure the Oracle GoldenGate capture using the "Archived Log Only" (ALO) mode. The capture process reads from the Standby database archived log files and the delivery process writes to the target database, opened on read/write mode. To minimize the lag, create a series of primary database online redo log groups with a log switch to sustain the lag maximum target.

Enabling Archived Log Only Mode

The purpose of operating Oracle GoldenGate on the Archived Log Only (ALO) mode is for the non-real-time data capture to be acceptable. It's configured using the parameter 'TRANLOGOPTIONS'. The options of this parameter are database-specific for controlling non-standard log-based extract options. The following summarizes the common options when used for capture from Oracle archived log files.

```
TRANLOGOPTIONS ARCHIVEDLOGONLY
TRANLOGOPTIONS ALTARCHIVELOGDEST PRIMARY /u01/app/oracle/archivelog/
TRANLOGOPTIONS ALTARCHIVEDLOGFORMAT %t_%s_%r.dbf
```

If the error 'OGG-00730' (no minimum supplemental logging is enabled), this may cause the extract process to handle key update incorrectly if the key column is not in the first row piece. If this happens, then take the action below.

1. Disable supplemental logging temporarily by adding the parameter 'TRANLOGOPTIONS DISABLESUPPLOGCHECK.'
2. Advance the extract position using the 'GGSCI ALTER' command:

```
GGSCI (target1.precisetrace.com) 6> ALTER EXTRACT E01S1C1, BEGIN now
EXTRACT altered.
```

3. Start the extract and wait until it processes beyond the RBA position
4. Stop the extract and disable the parameter 'TRANLOGOPTIONS DISABLESUPPLOGCHECK' and start it again.

Configure the Primary Extract Process

The primary extract capture transaction using the archived log file is a technique that's exclusive when using the option 'ARCHIVELOGONLY.' The Extract 'ES1C101' capture behavior is controlled by the parameter 'TRANLOGOPTIONS.'

```
GGSCI (target1.precisetrace.com) 4> EDIT PARAMS E01S1C1
```

```
EXTRACT E01S1C1
SETENV (ORACLE SID=T1C1)
SETENV (ORACLE_HOME="/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS LANG=AMERICAN AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle SYSDBA
RMTHOST 192.168.1.70, MGRPORT 7810
RMTTRAIL ./dirdat/sa
DISCARDFILE ./dirrpt/E01S1C1.dsc, APPEND
REPORTCOUNT EVERY 30 MINUTES, RATE
-- TRANLOGOPTIONS DISABLESUPPLOGCHECK
TRANLOGOPTIONS ARCHIVEDLOGONLY
TRANLOGOPTIONS ALTARCHIVELOGDEST PRIMARY /u01/app/oracle/archivelog/
TRANLOGOPTIONS ALTARCHIVEDLOGFORMAT %t_%s_%r.dbf
FETCHOPTIONS, NOUSESNAPSHOT, NOUSELATESTVERSION, MISSINGROW REPORT
TABLE ggs_admin.customers, keycol(col1);
```

```
GGSCI (target1.precisetrace.com) 11> ADD EXTRACT E01S1C1, TRANLOG, BEGIN NOW
EXTRACT added.

GGSCI (target1.precisetrace.com) 12> ADD RMTTRAIL ./dirdat/sa, EXTRACT E01S1C1
RMTTRAIL added.

GGSCI (target1.precisetrace.com) 13>
```

Configure the Delivery Process

The delivery process applies the transactions to the target database that is opened for read-write operations. Since the Extract captures the transactions from the standby database archived log files only and Extract & Replicat lags, the target database always is relatively out-of-sync with the primary database. Configuring a small size of the online redo groups and increasing the number of database archive processes reduces the gap between the primary & target databases.

```
GGSCI (t2c1.precisetrace.com) 3> EDIT PARAMS R01S1C1
```

```
REPLICAT R01S1C1
SETENV (ORACLE_SID=T1C1)
SETENV (NLS LANG=AMERICAN AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/r01S1C1.dsc, PURGE
MAP ggs_admin.customers, TARGET ggs_admin.customers;
```

```
GGSCI (t2c1.precisetrace.com) 8> ADD REPLICAT R01S1C1, EXTTRAIL ./dirdat/sa
REPLICAT added.

GGSCI (t2c1.precisetrace.com) 9>
```

Starting Replicat and Extract Processes

Begin by starting the Replicat 'R01S1C1.' The Replicat applies the remote trails to the target database – T1C1.

```
GGSCI (t2c1.precisetrace.com) 10> START REPLICAT R01S1C1

Sending START request to MANAGER ...
REPLICAT R01S1C1 starting


GGSCI (t2c1.precisetrace.com) 11> INFO ALL

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     R01S1C1     00:00:00      00:00:01


GGSCI (t2c1.precisetrace.com) 12>
```

Next, start the primary Extract 'E01S1C1.' Since there is no secondary extract (data pump) configured, the primary Extract performs the capture & route functions. It captures from the standby database archive log files and routes to the target database.

```
GGSCI (target1.precisetrace.com) 14> START EXTRACT E01S1C1

Sending START request to MANAGER ...
EXTRACT E01S1C1 starting


GGSCI (target1.precisetrace.com) 15> INFO ALL

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     E01S1C1     00:00:00      00:06:50


GGSCI (target1.precisetrace.com) 16>
```

Verify Extract, Replicat, and Lagging

During a normal database operation, there will be a gap. The database remains relatively unsynchronized and the target database is behind the primary database. Oracle GoldenGate lag is measured between the standby database and the target database only. For precise measurement, the use of Oracle GoldenGate Veridata between the primary (the target) is an excellent option.

```
GGSCI (t2c1.precisetrace.com) 12> LAG REPLICAT R01S1C1

Sending GETLAG request to REPLICAT R01S1C1 ...
No records yet processed.
At EOF, no more records to process.

GGSCI (t2c1.precisetrace.com) 13>
```

Migration from Oracle Streams to GoldenGate

Oracle Streams is an internal component of the Oracle database. Oracle Streams delivers a high-performance Oracle database replication only. Enterprises running multiple database types need a single software to support multiple database vendors with a flexible configuration environment.

According to Oracle strategic direction for data replication, Oracle GoldenGate has deprecated Oracle Streams. Oracle Streams will continue to be part of the database, but without taking advantage of the new release of Oracle Database.

Customers are encouraged to migrate from Oracle Streams to GoldenGate.
The following illustrates the migration phases from Oracle Streams to Oracle GoldenGate. The migration is online and delivers near-zero downtime.

Migration Architecture

Enterprises running critical mission system demands on an online migration from Oracle Streams to Oracle GoldenGate. Users continue to access the existing database until the new target database is completely synchronized with the source database. For highly secured environments, the option of deploying Oracle GoldenGate integrated capture eliminates the requirement of installing Oracle GoldenGate on the production system. Instead, it's installed on a downstream system receiving
a continuous streams of the database transaction logs for capture, routing, and delivery to the target database.

As shown in Figure 10-2, the migration consists of the following components:

- The Source Oracle Database (S1C1)
- Oracle Database Streams
- The Target Oracle Database (T1C1) for Oracle Streams
- Oracle GoldenGate Installation on the Source and Target Databases
- The Target Oracle Database (T2C1) for Oracle GoldenGate
- Oracle GoldenGate Veridata

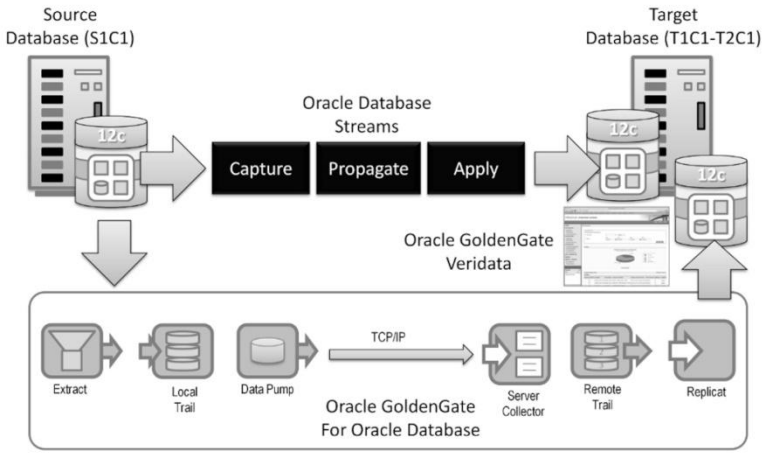


Figure 10-2: Migration from Streams to Oracle GoldenGate Components

GoldenGate Veridata helps make the decision to disable Oracle Streams after verifying the source and new target are synchronized. The downtime incurred is proportional to the database transactions volume at the time of momentarily disabling the source (production) database. Hence, the such decision should be carefully chosen during the lowest database transactions volume.

Oracle Streams Database Parameters

Oracle Streams set–up requires database parameters setting. These parameters have initial values. However, changing them enhances Oracle Streams performance and reliability.

```
SQL> ALTER DATABASE RENAME GLOBAL_NAME TO S2C1.ggs.com;

Database altered.

SQL> ALTER SYSTEM SET  aq_tm_processes = 1 SCOPE=BOTH;

System altered.

SQL> ALTER SYSTEM SET  job_queue_processes = 1000 SCOPE=BOTH;

System altered.
```

```
SQL>
```

Streams Database Administrator User

Oracle Streams needs a dedicated database user who is granted the privileges for performing the functions of Oracle Streams. The following statements create the tablespace and the user then grant the database privileges required by Oracle Streams.

Execute the below commands on the source and destination databases.

```
SQL> CREATE TABLESPACE STREAMSDB
  2  DATAFILE '/u01/app/oracle/oradata/S2C1/datafile/streamsdb01.dbf'
  3  SIZE 100M
  4  AUTOEXTEND ON;

Tablespace created.

SQL> CREATE USER streamadmin IDENTIFIED BY oracle
  2  DEFAULT TABLESPACE STREAMSDB
  3  QUOTA UNLIMITED ON STREAMSDB;

User created.

SQL> GRANT CONNECT, RESOURCE, AQ_ADMINISTRATOR_ROLE,DBA TO streamadmin;

Grant succeeded.

SQL> EXECUTE DBMS_STREAMS_AUTH.GRANT_ADMIN_PRIVILEGE('STREAMADMIN');

PL/SQL procedure successfully completed.

SQL> GRANT EXECUTE ON DBMS_AQADM TO streamadmin;

Grant succeeded.

SQL> GRANT EXECUTE ON DBMS_APPLY_ADM TO streamadmin;

Grant succeeded.

SQL> GRANT EXECUTE ON DBMS_CAPTURE_ADM TO streamadmin;

Grant succeeded.

SQL> GRANT EXECUTE ON DBMS_PROPAGATION_ADM TO streamadmin;

Grant succeeded.

SQL> GRANT EXECUTE ON DBMS_STREAMS TO streamadmin;

Grant succeeded.

SQL> GRANT EXECUTE ON DBMS_STREAMS_ADM TO streamadmin;

Grant succeeded.
```

```
SQL> BEGIN
  2  DBMS_RULE_ADM.GRANT_SYSTEM_PRIVILEGE(
  3  privilege =>DBMS_RULE_ADM.CREATE_RULE_SET_OBJ,
  4  grantee =>'streamadmin',
  5  grant_option =>FALSE);
  6  END;
  7  /

PL/SQL procedure successfully completed.

SQL>
```

Database Links

Oracle Streams uses Database Links to propagate database to the destination database. The database links connect to the Oracle Streams Schema created above. The following SQL commands create the database link and perform a basic test. Query the data dictionary view DBA_DB_LINKS to find database link details.

```
SQL> CREATE PUBLIC DATABASE LINK D1C1.ggs.com
  2  CONNECT TO streamadmin IDENTIFIED BY oracle
  3  USING 'D1C1.ggs.com';

Database link created.

SQL> SELECT * FROM dual@D1C1.ggs.com;

D
-
X

SQL> CREATE PUBLIC DATABASE LINK S2C1.ggs.com
  2  CONNECT TO streamadmin IDENTIFIED BY oracle
  3  USING 'S2C1.ggs.com';

Database link created.

SQL> SELECT * FROM dual@S2C1.ggs.com;

D
-
X

SQL>
```

Configure Streams Capture and Apply Queues

The capture and apply queues are created using multiple steps. The following PL/SQL steps create Oracle Streams ANYDATA queue, proration rules, and instantiation of the destination schema.

- Before creating any of Oracle Streams components, use the 'PL/SQL DBMS_STREAMS_ADM.SET_UP_QUEUE' to create any 'ANYDATA' queue. This queue is used by the capture, propagation, and application processes.

```
SQL> CONN streamadmin/oracle@S2C1.ggs.com
Connected.
SQL> BEGIN
  2  DBMS_STREAMS_ADM.SET_UP_QUEUE(
  3  queue_name =>'STREAMS_Q1',
  4  queue_table =>'STREAMS_Q1_TAB',
  5  queue_user =>'STREAMADMIN');
  6  END;
  7  /

PL/SQL procedure successfully completed.

SQL> CONN streamadmin/oracle@D1C1.ggs.com
Connected.
SQL> BEGIN
  2  DBMS_STREAMS_ADM.SET_UP_QUEUE(
  3  queue_name =>'STREAMS_Q1',
  4  queue_table =>'STREAMS_Q1_TAB',
  5  queue_user =>'STREAMADMIN');
  6  END;
  7  /

PL/SQL procedure successfully completed.
```

- Determine included and excluded messages. Use the package 'DBMS_STREAMS_ADM.ADD_TABLE_PROPAGATION_RULES.' Specify the same Streams name, source queue, and destination queue as defined for the existing propagation.

```
SQL> CONN streamadmin/oracle@S2C1.ggs.com
Connected.
SQL> BEGIN
  2  DBMS_STREAMS_ADM.ADD_GLOBAL_PROPAGATION_RULES(
  3  streams_name =>'STREAMS_PROP_R1',
  4  source_queue_name =>'STREAMADMIN.STREAMS_Q1',
  5  destination_queue_name =>'STREAMADMIN.STREAMS_Q1@D1C1.ggs.com',
  6  include_dml =>TRUE,
  7  include_ddl =>TRUE,
  8  source_database =>'S2C1.ggs.com');
  9  END;
 10  /

PL/SQL procedure successfully completed.

SQL> BEGIN
  2  DBMS_STREAMS_ADM.ADD_GLOBAL_RULES(
  3  streams_type =>'CAPTURE',
  4  streams_name =>'STREAMADMIN_CAPTURE',
  5  queue_name =>'STREAMADMIN.STREAMS_Q1',
  6  include_dml =>TRUE,
  7  include_ddl =>TRUE,
```

```
  8  source_database =>'S2C1.ggs.com');
  9  END;
 10  /

PL/SQL procedure successfully completed.

SQL> CONN streamadmin/oracle@D1C1.ggs.com
Connected.
SQL> BEGIN
  2  DBMS_STREAMS_ADM.ADD_GLOBAL_RULES(
  3  streams_type =>'APPLY',
  4  streams_name =>'STREAMADMIN_APPLY',
  5  queue_name =>'STREAMADMIN.STREAMS_Q1',
  6  include_dml =>TRUE,
  7  include_ddl =>TRUE,
  8  source_database =>'S2C1.ggs.com');
  9  END;
 10  /

PL/SQL procedure successfully completed.

SQL> BEGIN
  2  DBMS_APPLY_ADM.SET_PARAMETER(
  3  apply_name =>'STREAMADMIN_APPLY',
  4  parameter =>'disable_on_error',
  5  value =>'n');
  6  END;
  7  /

PL/SQL procedure successfully completed.
```

- From the source database, manually instantiate the destination schema as shown below. The instantiation uses the System Change Number (SCN) to ensure source and target objects are consistent from the determined SCN.

```
SQL> CONN streamadmin/oracle@S2C1.ggs.com
Connected.
SQL> DECLARE
  2    vscn NUMBER;
  3  BEGIN
  4     vscn:=DBMS_FLASHBACK.GET_SYSTEM_CHANGE_NUMBER();
  5     DBMS_APPLY_ADM.SET_SCHEMA_INSTANTIATION_SCN@S2C1.ggs.com(
  6     SOURCE_SCHEMA_NAME => 'SCOTT',
  7     SOURCE_DATABASE_NAME => 'D1C1.ggs.com',
  8     INSTANTIATION_SCN => vscn,
  9     RECURSIVE => TRUE);
 10  END;
 11  /

PL/SQL procedure successfully completed.

SQL> conn streamadmin/oracle@D1C1.ggs.com
Connected.
SQL> DECLARE
  2    vscn NUMBER;
  3  BEGIN
  4     vscn:=DBMS_FLASHBACK.GET_SYSTEM_CHANGE_NUMBER();
```

```
 5     DBMS_APPLY_ADM.SET_SCHEMA_INSTANTIATION_SCN@D1C1.ggs.com(
 6     SOURCE_SCHEMA_NAME => 'SCOTT',
 7     SOURCE_DATABASE_NAME => 'S2C1.ggs.com',
 8     INSTANTIATION_SCN => vscn,
 9     RECURSIVE => TRUE);
10  END;
11  /

PL/SQL procedure successfully completed.

SQL>
```

Starting the Application and Capture Processes

The final step is to start the application on the destination and capture on the source database.

```
SQL> CONN streamadmin/oracle@D1C1.ggs.com
Connected.
SQL> BEGIN
  2  DBMS_APPLY_ADM.START_APPLY(
  3  apply_name =>'STREAMADMIN_APPLY');
  4  END;
  5  /

PL/SQL procedure successfully completed.

SQL> CONN streamadmin/oracle@S2C1.ggs.com
Connected.
SQL> BEGIN
  2  DBMS_CAPTURE_ADM.START_CAPTURE(
  3  capture_name =>'STREAMADMIN_CAPTURE');
  4  END;
  5  /

PL/SQL procedure successfully completed.

SQL>
```

Initial Data Load

The new database (D2C1) must be instantiated and there are several initial data load techniques. For homogenous, Oracle-to-Oracle environments, the Recovery Manager (RMAN) is an option. For heterogeneous environments, Oracle GoldenGate initial-load methods and the Oracle database utilities are alternative options.

Next, the Oracle GoldenGate Direct Load method is illustrated to establish the new target database (D2C1). The Direct Load method implements an initial-load Extract, the Extract reads and sends data directly to a remote task (Replicat). The Replicat uses the native database SQL calls to perform sequential database insert statements.

Refer to Chapter 3 for more details regarding the initial load techniques.

Configure Initial-Load Extract

The Initial Load Extract (ILE1S2C1) fetches data from the source tables through the TCP/IP and sends the data to a target Replicat remote task.

```
GGSCI (source1.precisetrace.com) 3> EDIT PARAMS ILE1S2C1
```

```
EXTRACT ILE1S2C1
USERID ggs_admin@S2C1, PASSWORD oracle
RMTHOST t2c1.precisetrace.com, MGRPORT 7813
RMTTASK REPLICAT, GROUP ILR1S2C1
TABLE scott.salgrade;
TABLE scott.bonus;
TABLE scott.dept;
TABLE scott.emp;
```

```
GGSCI (source1.precisetrace.com) 6> ADD EXTRACT ILE1S2C1, SOURCEISTABLE
EXTRACT added.

GGSCI (source1.precisetrace.com) 7>
```

Configure Initial-Load Replicat

The Initial-Load Replicat (ILR1S2C1) is configured to be used as a special task. It's started by the Initial-Load Extract.

```
GGSCI (t2c1.precisetrace.com) 5> EDIT PARAMS ILR1S2C1
```

```
REPLICAT ILR1S2C1
USERID ggs_admin@D2c1, password oracle
ASSUMETARGETDEFS
HANDLECOLLISIONS
DISCARDFILE ./dirrpt/initILR1S2C101.dsc, PURGE
MAP scott.*, TARGET scott.*;
```

```
GGSCI (t2c1.precisetrace.com) 4> ADD REPLICAT ILR1S2C1, SPECIALRUN
REPLICAT added.

GGSCI (t2c1.precisetrace.com) 5>
```

Starting the Initial Data Load Processes

The Initial Data Load capture and delivery is performed by starting the Initial Data Load capture only. The Extract automatically starts the Initial Load Replicat as a special task

process. After the Initial Data Load capture is completed, the Initial Data Load Replicat applies the last transactions then gracefully terminates.

```
GGSCI (source1.precisetrace.com) 5> START EXTRACT ILE1S2C1

Sending START request to MANAGER ...
EXTRACT ILE1S2C1 starting

GGSCI (source1.precisetrace.com) 6
```

Verify the Initial Data Load

Run the 'GGSCI REPORT' command from the source and/or the target database. The "Run Time Statistics" section of the report displays the occurred inserts for each table configured for the Initial Data Load.

```
GGSCI (source1.precisetrace.com) 1> VIEW REPORT ILE1S2C1
```

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


Report at 2015-03-27 14:39:00 (activity since 2015-03-27 14:38:34)

Output to ILR1S2C1:

From Table SCOTT.SALGRADE:
       #                    inserts:         8
       #                    updates:         0
       #                    deletes:         0
       #                   discards:         0
From Table SCOTT.DEPT:
       #                    inserts:         4
       #                    updates:         0
       #                    deletes:         0
       #                   discards:         0
From Table SCOTT.EMP:
       #                    inserts:        14
       #                    updates:         0
       #                    deletes:         0
       #                   discards:         0


REDO Log Statistics
--More--(98%)
```

Deploying Changed Data Synchronization

After the new destination (target) is successfully instantiated, the following sequence of steps configure the Changed Data Synchronization from the source to the target database. The synchronization is performed in parallel with the Capture-Propagate-Apply of Oracle Streams. As per best practices, the configuration deploys a secondary capture (data pump) for sending data to the remote Data Collector Server. The data pump enables data capture continuity in the event of a network failure. The Primary Extract continues to capture and

write to the local trails. After the network is restored and the data pump process is restarted, the target database synchronization resumes.

For an environment with a multiple independent database schema design, create a separate Extract-Data pump and Replicat. For tables with high-volume transactions, consider the use of a parallel capture and delivery design. Refer to Chapter 1 for details regarding creating a parallel Extract and Replicat configuration.

Configure the Primary Extract Process

Prepare the parameter file for the extract 'E01S2C1.' This extract process writes to the local trail './dirdat/sa.' It captures data for the objects listed by the 'TABLE' parameters.

```
GGSCI (source1.precisetrace.com) 5> EDIT PARAMS E01S2C1
```

```
EXTRACT E01S2C1
SETENV (ORACLE SID=S2C1)
SETENV (ORACLE_HOME="/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS LANG=AMERICAN AMERICA.AL16UTF16)
USERID ggs_admin@S2C1, PASSWORD oracle
EXTTRAIL ./dirdat/sa
TABLE scott.dept;
TABLE scott.emp;
TABLE scott.bonus;
TABLE scott.salgrade;
```

The 'ADD EXTRACT' command creates the primary extract – 'E01S2C1.' The 'ADD EXTTRAIL' creates the local trail - ./dirdata/sa, and links it with the primary extract – 'E01S2C1.'

```
GGSCI (source1.precisetrace.com) 9> ADD EXTRACT E01S2C1, TRANLOG, BEGIN now,
THREADS 1
EXTRACT added.

GGSCI (source1.precisetrace.com) 10> ADD EXTTRAIL ./dirdat/sa, EXTRACT E01S2C1
EXTTRAIL added.

GGSCI (source1.precisetrace.com) 11>
```

Configure the Secondary Extract Process

As per Oracle GoldenGate best practices, a secondary extract process (data pump) configuration is deployed. Prepare the parameter file for the data pump – 'P01S2C1.' The data pump reads the local trail and sends it to the destination (target database).

```
GGSCI (source1.precisetrace.com) 12> EDIT PARAMS P01S2C1
```

```
EXTRACT P01S2C1
USERID ggs_admin@S2C1, PASSWORD oracle
PASSTHRU
RMTHOST t2c1.precisetrace.com, MGRPORT 7813
RMTTRAIL ./dirdat/ta
TABLE scott.dept;
TABLE scott.emp;
TABLE scott.bonus;
TABLE scott.salgrade;
```

The command 'ADD EXTRACT' creates the data pump extract, the option 'EXTTRAILSOURCE' instructs the extract to read from the local trail.

```
GGSCI (source1.precisetrace.com) 18> ADD EXTRACT P01S2C1, EXTTRAILSOURCE
./dirdat/sa
EXTRACT added.

GGSCI (source1.precisetrace.com) 19> ADD RMTTRAIL ./dirdat/ta, EXTRACT P01S2C1
RMTTRAIL added.

GGSCI (source1.precisetrace.com) 20>
```

Configure the Replicat Process

On the destination system, prepare the parameter file for the Replicat – 'R01S2C1.' The Replicat uses the target database native SQL interface to apply the transactions. The MAP command uses the wildcard character '*'. It designates all objects associated with the data pump 'TABLE' parameters.

```
GGSCI (t2c1.precisetrace.com) 3> EDIT PARAMS R01S2C1
```

```
REPLICAT R01S2C1
SETENV (ORACLE SID=D2C1)
SETENV (NLS LANG=AMERICAN AMERICA.AL16UTF16)
USERID ggs admin@D2C1, PASSWORD oracle
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/r01s2c1.dsc, PURGE
MAP scott.*, TARGET scott.*;
```

As per best practices and default settings, the Replicat requires a checkpoint table. Ensure the table created is already created or use the 'ADD CHECKPOINTTABLE' command to create a new database table.

Refer to Chapter 1 for more details regarding creating the checkpoint table.

```
GGSCI (t2c1.precisetrace.com) 5> INFO CHECKPOINTTABLE

No checkpoint table specified. Using GLOBALS specification
(ggs_admin.chkpttab)...
```

```
Checkpoint table ggs_admin.chkpttab created 2015-03-27 15:22:16.
```

The command 'ADD REPLICAT' creates the Replicat process. Specify the remote trail name.

```
GGSCI (t2c1.precisetrace.com) 6> ADD REPLICAT R01S2C1, EXTTRAIL ./dirdat/ta
REPLICAT added.

GGSCI (t2c1.precisetrace.com) 7>
```

Start and Verify Oracle GoldenGate Extracts and Replicat Processes

Now, it's time to start the Extracts and Replicat processes. From the source system, the command 'START *' starts the primary and secondary extracts. Use the 'INFO ALL' to verify the status.

```
GGSCI (source1.precisetrace.com) 22> START *

Sending START request to MANAGER ...
EXTRACT E01S2C1 starting

Sending START request to MANAGER ...
EXTRACT P01S2C1 starting

GGSCI (source1.precisetrace.com) 23> INFO ALL

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     E01S2C1     00:23:41      00:00:03
EXTRACT     RUNNING     P01S2C1     00:00:00      00:00:00

GGSCI (source1.precisetrace.com) 24>
```

From the destination system (target), the command 'START *' starts the Replicat process. Use the 'INFO ALL' to verify the status.

```
GGSCI (t2c1.precisetrace.com) 8> START *

Sending START request to MANAGER ...
REPLICAT R01S2C1 starting


GGSCI (t2c1.precisetrace.com) 9> INFO ALL

Program     Status      Group       Lag at Chkpt  Time Since Chkpt
```

```
MANAGER     RUNNING
REPLICAT    RUNNING     R01S2C1     00:00:00      00:00:02


GGSCI (t2c1.precisetrace.com) 10>
```

Data Verification and Failback Option

The purpose of using Oracle GoldenGate Veridata is to ensure the target database is fully synchronized with the source database prior to making the final decision for the switchover. Figure 10-3 shows Oracle GoldenGate Veridata.

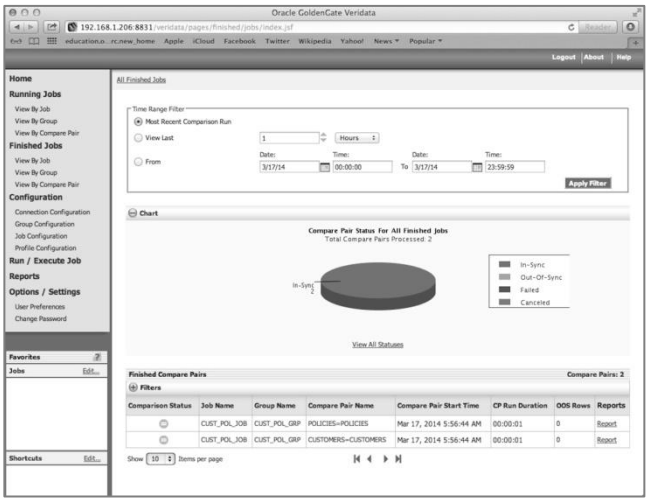


Figure 10-3: Oracle GoldenGate Veridata

The failback option is already in place by concurrently running Oracle Streams and Oracle GoldenGate. A worst case failback scenario is to rebuild the GoldenGate instance with impacting the source and destination system. Refer to Chapter 6 for more details regarding Oracle GoldenGate Veridata implementation.

Without using Oracle GoldenGate Veridata, the admin user can verify the concurrent data manipulation and movement by Oracle Streams and Oracle GoldenGate.

- Insert data into the source system (S2C1)

```
SQL> CONN system/oracle 4U@S2C1
Connected.
SQL> INSERT INTO scott.emp VALUES (
  2  8001, 'FAWZI', 'ANALYST', 7698, SYSDATE, 10000, NULL, 10);

1 row created.
```

```
SQL> COMMIT;

Commit complete.

SQL>
```

- Verify transaction movement maintained by Oracle Streams

```
SQL> CONN system/oracle 4U@D1C1.GGS.COM
Connected.
SQL> SELECT count(*) FROM scott.emp
  2  WHERE  EMPNO=8001;

  COUNT(*)
----------
      1

SQL>
```

- Verify the same transaction movement maintained by Oracle GoldenGate

```
SQL> CONN system/oracle_4U@D2C1
Connected.
SQL> SELECT count(*) FROM scott.emp
  2  WHERE  EMPNO=8001;

  COUNT(*)
----------
      1

SQL>
```

The above deployment is performed using Oracle GoldenGate classic capture. For highly restricted and secured environments, the Oracle GoldenGate Integrated Capture is deployed on a downstream system. The downstream system does not require the installation of Oracle GoldenGate on the production system. The next section addresses the details of configuration the Oracle GoldenGate integrated capture, a new Oracle GoldenGate 11.2 and above for Oracle Database only.

Integrated Capture for Oracle Database

For Oracle Database only, Oracle GoldenGate 11.2 supports two types of capture — classic and integrated. The primary purpose of the integrated capture is to overcome the major limitations of the classic capture. As the name suggests, it's the integration of the Oracle GoldenGate capture technique with the Oracle Database RDBMS. Hence the name derived – Integrated Capture. The integrated capture understands the internal format of the database's Logical Change Record (LCR) produced by the Logmining server. In regard to the database backup and recovery, the Recovery Manager

(RMAN) manages the database archives by retaining archived log files needed by the integrated capture, which is not the case with the classic capture.

Unlike the classic capture, the integrated capture has support for the following database features:

- Basic and OLTP Compression
- Exadata Hybrid Columnar Compression
- IOTs Support
- XML Files (stored as binary)
- Parallel DML on a Real Application Cluster (RAC)

The integrated capture and classic capture can work together, where a set of objects are captured by the classic capture and another set of objects are captured by the integrated capture. However, existing Oracle GoldenGate installations are highly recommended to migrate to the integrated capture.

Architecture

The Integrated Capture core component is the Logmining Database. The Logmining Database creates the Logical Change Record (LCR) from the database transactions log, online redo log files for real-time capture, or archived files for deferred capture.
The integrated capture has two deployment modes— local and downstream deployments. Figure 10-4 shows the components of the two deployment options. Regardless of the deployment type, the Logmining Database must be release 11.2.0.3 or higher. Because the data is not written to the database files, the Logmining Database has a small footprint database without overhead of the source database data files, it has only the metadata and infrastructure required for processing the online redo log or archive files. The Integrated Capture is highly integrated with the Recovery Manager (RMAN); hence, it's capable of switching among mirrors of the archive log files when operating on a deferred configuration. For Oracle Database release 10.2.0.0, the integrated capture must be operated as a downstream system, this has the advantage of offloading the source database from the capture processing and avoiding installing Oracle GoldenGate on a highly secured OLTP environment.

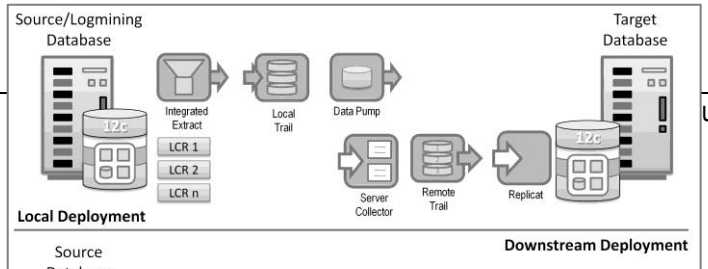


Figure 10-4 Integrated Capture Deployment Modes

Downstream Deployment Mode

The Logmining Database operates on a separate system (downstream system). The source system sends the transaction logs using the TCP/IP to the Logmining Database designated destination.

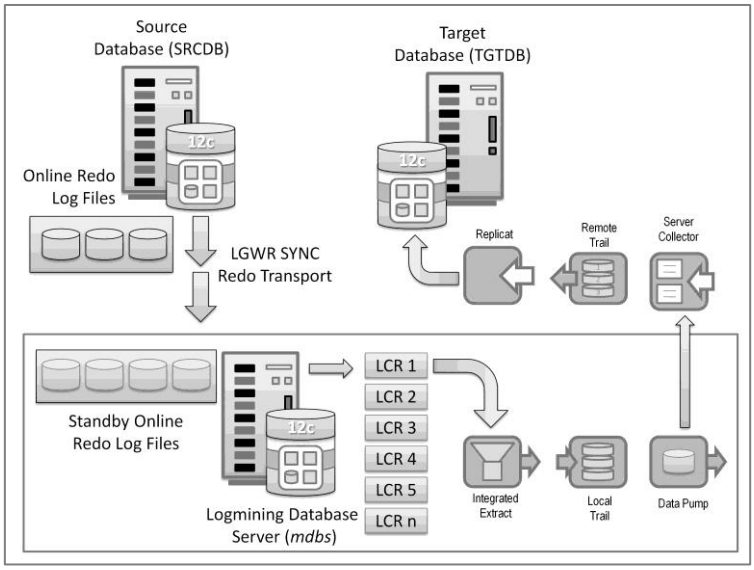


Figure 10-5: Downstream Mode Integrated Capture

The configuration of the source database parameters are similar to the parameters settings for the Data Guard configuration. Sending the data from the source database to the

Logmining Database performed using either Asynchronous or Synchronous transport mode. The major advantage required by enterprises, the database configuration parameters are dynamic, which enables an online deployment of Oracle GoldenGate on a downstream deployment mode. Presumably, the source database is operating on Archive Log mode. The bottom section of Figure 10-4 shows the downstream deployment mode. This mode requires additional configuration to send the database transaction logs to the downstream system for creating the LCR and performing the preferred capture.

Local Deployment Mode

The Logmining Database operates using the same source database instance and infrastructure. The Logmining Database creates the Logical Change Record (LCR) on the same instance designated destination, then the Integrated Capture acts on the Logical Change Record (LCR) for writing to the local or the remote trail. The top section of Figure 10-4 shows the local deployment option. Because redo transport is not required, this mode requires no additional database configuration parameters. Figure 10-6 depicts the components for a local integrated capture.

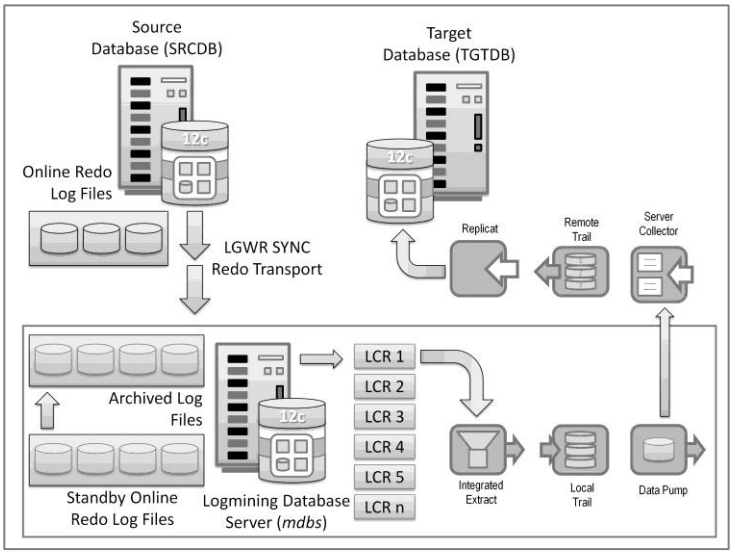


Figure 10-6: Local Mode Integrated Capture

Whether the integrated capture deployed locally or using a downstream system, the capture is configurable as a real-time or deferred capture.

Real-Time Integrated Capture Configuration

The real-time integrated capture is enabled using the standby online redo log file

by the Logmining Database. The redo transport should be delivered on Synchronous mode to deliver an absolute real-time capture.

Deferred Integrated Capture Configuration

The deferred integrated capture is enabled using the archive files by the Logmining Database. The redo transport delivered on Synchronous or Asynchronous mode. The Integrated Capture (Extract) lag is proportional to the log switch occurs on the standby online redo log files, which is inherited from the source database online redo log groups, in term of group size and number.

The Downstream Logmining Database Server

This is where the integrated capture is defined. The Downstream Logmining Database Server must operate on Archive Log Mode. It must be configured to archive the redo or archived log files received from the source database.

Integrated Capture Prerequisites and Options

The following database settings must be performed on the source database, which are specific to an integrated capture.

- Memory Requirement Settings
- Enable an Integrated Capture Specific Database Supplemental Logging
- Create and Grant Oracle GoldenGate Database User

Database Memory Requirements for Integrated Capture

The integrated capture uses the database System Global Area (SGA) for creating the Logical Change Records (LCR). It utilizes the Streams Pool memory structure; The recommended size is approximately 700MB. When the SGA is dynamically allocated, manually allocate enough Streams Pool for an Oracle GoldenGate integrated capture. Query the data dictionary V$SGAINFO to display the current allocated memory size for the different System Global Area (SGA) components, if the Stream Pool Size reported is less than 700MB, then change the setting using the ALTER SYSTEM command.

```
SQL> SELECT NAME, BYTES
  2  FROM   v$sgainfo;

NAME                                  BYTES
-------------------------------- ----------
Fixed SGA Size                      2289016
Redo Buffers                        4235264
```

```
Buffer Cache Size                159383552
Shared Pool Size                 138412032
Large Pool Size                    8388608
Java Pool Size                     4194304
Streams Pool Size                  4194304
Shared IO Pool Size                4194304
Data Transfer Cache Size                 0
Granule Size                       4194304
Maximum SGA Size                 413372416

NAME                                 BYTES
-------------------------------- ----------
Startup overhead in Shared Pool    113301824
Free SGA Memory Available           92274688

13 rows selected.

SQL> ALTER SYSTEM SET streams_pool_size=700M SCOPE=BOTH;

System altered.

SQL>
```

INTEGRATEDPARAMS Options

For fine-grain memory sizing, Logmining Database server degree of parallelism and real-time mode employed by the integrated capture, use the Oracle GoldenGate parameter 'TRANLOGOPTIONS INTEGRATEDPARAMS' as outlined below.

```
TRANLOGOPTIONS INTEGRATEDPARAMS (max_sga_size 128)
TRANLOGOPTIONS INTEGRATEDPARAMS (parallelism 2)
TRANLOGOPTIONS INTEGRATEDPARAMS (downstream_real_time_mine Y)
```

The parameter 'max_sga_size' specifies the amount (in Gegabytes) of the System Global Area (SGA) used by the Logmining Database server. If the database parameter streams_pool_size is greater than 1GB, it defaults to 1GB. Otherwise, it's calculated as 75% of the streams_pool_size value.

The parameter 'parallelism' specifies the maximum number of processes spawned by the Logmining Database server. Be careful when setting the degree of parallelism, setting the degree to a large number may consume unnecessary system resources without taking advantage of optimized parallel processing. The default value is 0 (parallelism is disabled).

The parameter 'downstream_real_time_mine' is for downstream deployment mode only. When setting the parameter to 'Y', the integrated capture performs a real-time capture from the standby online redo log files. When setting the parameter to 'N' (Default), the integrated capture performs a delayed capture from the archive log file.

Enable GoldenGate Replication

For an integrated capture, Oracle Database 12.1 requires setting the database parameter 'ENABLE_GOLDENGATE_REPLICATION' to 'TRUE.' It enables supplemental logging for new data types. The parameter is dynamic as shown below.

```
SQL> ALTER SYSTEM SET ENABLE_GOLDENGATE_REPLICATION=TRUE SCOPE=BOTH;

System altered.

SQL>
```

Set the parameter for the source and target database.

GoldenGate Database User

For the integrated capture, the Oracle GoldenGate database user needs to be granted 'SELECT' privilege on the 'V_$DATABASE' dynamic view. In addition, execute the PL/SQL database package 'DBMS_GOLDENGATE_AUTH' to grant the administrative role to the GoldenGate database user.

```
SQL> GRANT DBA TO ggs_admin IDENTIFIED BY oracle;

Grant succeeded.

SQL> BEGIN
  2     DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE
  3          (grantee=>'GGS_ADMIN',
  4           privilege_type=>'capture',
  5           grant_select_privileges=>true,
  6           do_grants=>TRUE);
  7  END;
  8  /

PL/SQL procedure successfully completed.

SQL> GRANT SELECT ON V_$DATABASE to ggs_admin;

Grant succeeded.

SQL>
```

Deploying an Integrated Capture on the Local System

The deployment of an integrated capture on the local system eliminates the major limitations of the classic capture. The Logmining Database server is the same source database instance. Hence, using the source online redo log files to generate the Logical Charge Record (LCR) passed to the integrated capture for processing. The only step

required for deploying an integrated capture on the local system is registering the Extract with the source database.

The steps to deploy an integrated capture on the local system are:

- Login to the source database using 'DBLOGIN' to register the Extract
- Specify the 'TRANLOGOPTIONS INTEGRATEDPARAMS' parameter options
- Add the Extract group specify 'INTEGRATED TRANLOG' mode

Register the Integrated Capture with the Database

Registering the capture process with the database enables the capture to understand that the format of the Logical Change Record (LCR) is passed by the Logmining Database server.

For locally deployed integrated captures, the Logmining database is the same as the source database. Login to the source database using 'DBLOGIN' and issue a 'REGISTER EXTRACT' command. It's not required to login using 'MININGDBLOGIN' to register the integrated capture extract.

```
GGSCI (oltp01.oracle-ggs.com) 1> DBLOGIN USERID ggs admin, PASSWORD oracle
Successfully logged into database.

GGSCI (oltp01.oracle-ggs.com) 2> REGISTER EXTRACT E01SRCDB WITH DATABASE
Extract E01SRCDB successfully registered with database at SCN 1828148.

GGSCI (oltp01.oracle-ggs.com) 3>
```

After successfully registering the integrated capture Extract, query the database dictionary view 'DBA_CAPTURE' to list the currently registered captures. Unregistering the Extract with the database removes the Oracle GoldenGate integrated capture from the capture list. The query below verifies the above Extract registration with the database.

```
SQL> desc dba_capture
 Name                                      Null?    Type
 ----------------------------------------- -------- --------------

 CAPTURE_NAME                              NOT NULL VARCHAR2(30)
 QUEUE_NAME                                NOT NULL VARCHAR2(30)
 QUEUE_OWNER                               NOT NULL VARCHAR2(30)
 RULE_SET_NAME                                      VARCHAR2(30)
 RULE_SET_OWNER                                     VARCHAR2(30)
 CAPTURE_USER                                       VARCHAR2(30)
 START_SCN                                          NUMBER
 STATUS                                             VARCHAR2(8)
 CAPTURED_SCN                                       NUMBER
 APPLIED_SCN                                        NUMBER
```

```
 USE_DATABASE_LINK                                  VARCHAR2(3)
 FIRST_SCN                                          NUMBER
 SOURCE_DATABASE                                    VARCHAR2(128)
 SOURCE_DBID                                        NUMBER
 SOURCE_RESETLOGS_SCN                               NUMBER
 SOURCE_RESETLOGS_TIME                              NUMBER
 LOGMINER_ID                                        NUMBER
 NEGATIVE_RULE_SET_NAME                             VARCHAR2(30)
 NEGATIVE_RULE_SET_OWNER                            VARCHAR2(30)
 MAX_CHECKPOINT_SCN                                 NUMBER
 REQUIRED_CHECKPOINT_SCN                            NUMBER
 LOGFILE_ASSIGNMENT                                 VARCHAR2(8)
 STATUS_CHANGE_TIME                                 DATE
 ERROR_NUMBER                                       NUMBER
 ERROR_MESSAGE                                      VARCHAR2(4000)
 VERSION                                            VARCHAR2(64)
 CAPTURE_TYPE                                       VARCHAR2(10)
 LAST_ENQUEUED_SCN                                  NUMBER
 CHECKPOINT_RETENTION_TIME                          NUMBER

SQL> SELECT capture_name, client_name, purpose
  2  FROM   dba_capture;

CAPTURE_NAME                   CLIENT_NAME PURPOSE
------------------------------ ----------  ------------------------------
OGG$CAP_E01SRCDB                E01SRCDB   GoldenGate Capture

SQL>
```

Enable Table Level Supplemental Logging

This can be performed for each table or using the Oracle GoldenGate wildcard character to add supplemental logging for the schema's table.

```
GGSCI (oltp01.oracle-ggs.com) 8> DBLOGIN USERID ggs_admin, PASSWORD oracle
Successfully logged into database.

GGSCI (oltp01.oracle-ggs.com) 9> ADD TRANDATA scott.*
```

The table without a primary key received a warning message— suggesting to use the 'KEYCOL' option of the TABLE statement.

Prepare and Define the Primary Integrated Extract

The integrated capture (E01SRCDB) is already registered with the database.
This database has supplemental log data and forced logging is enabled. Use the 'EDIT' command to prepare the parameter file. Notice the parameter 'TRANLOGOPTIONS' has been extended to include the 'INTEGRATEDPARAMS' option. Then execute the 'ADD EXTRACT' command to define the Extract in the 'INTEGRATED TRANLOG' mode. The last step, when defining the primary extract, is to execute the 'ADD EXTTRAIL' command to link the Extract (E01SRCDB) with the local trail './dirdat/sa.'

```
GGSCI (oltp01.oracle-ggs.com) 3> EDIT PARAMS E01SRCDB
```

```
EXTRACT E01SRCDB
USERID ggs admin@SRCDB, PASSWORD oracle
TRANLOGOPTIONS INTEGRATEDPARAMS (max_sga_size 128)
TRANLOGOPTIONS INTEGRATEDPARAMS (parallelism 2)
EXTTRAIL ./dirdat/sa
TABLE scott.*;
```

```
GGSCI (oltp01.oracle-ggs.com) 6> ADD EXTRACT E01SRCDB, INTEGRATED TRANLOG, BEGIN
now
EXTRACT added.
```

```
GGSCI (oltp01.oracle-ggs.com) 7> ADD EXTTRAIL ./dirdat/sa, EXTRACT E01SRCDB,
MEGABYTES 15
EXTTRAIL added.

GGSCI (oltp01.oracle-ggs.com) 8>
```

In order to allow the primary integrated extract to receive current redo logs from the source database, execute the SQL statement to perform a log switch.

```
SQL> ALTER SYSTEM SWITCH LOGFILE;

System altered.

SQL>
```

Prepare and Define the Secondary Extract

The Secondary Extract is optional, but a highly recommend Oracle GoldenGate practice. The purpose is to offload the primary extract from forwarding data to the target system and to provide system resilience in case of a network disconnect. The following 'GGSCI' command prepares the parameter file, defines the Secondary Extract, and links the remote trail with the Secondary Extract (data pump).

```
GGSCI (oltp01.oracle-ggs.com) 8> EDIT PARAMS P01SRCDB
```

```
EXTRACT P01SRCDB
USERID ggs_admin@SRCDB, PASSWORD oracle
RMTHOST 192.168.1.78, MGRPORT 7809
RMTTRAIL ./dirdat/sb
PASSTHRU
TABLE scott.*;
```

```
GGSCI (oltp01.oracle-ggs.com) 10> ADD EXTRACT P01SRCDB, EXTTRAILSOURCE
./dirdat/sa
EXTRACT added.
```

```
GGSCI (oltp01.oracle-ggs.com) 11> ADD RMTTRAIL ./dirdat/sb, EXTRACT P01SRCDB,
MEGABYTES 15
RMTTRAIL added.

GGSCI (oltp01.oracle-ggs.com) 12>
```

Start Primary and Secondary Extract

Start and verify the Extracts' status. The 'INFO ALL' command shows the status is RUNNING. If not, use the 'VIEW REPORT' command to display the report file which indicates the cause of failure.

```
GGSCI (oltp01.oracle-ggs.com) 39> START EXTRACT *

Sending START request to MANAGER ...
EXTRACT E01SRCDB starting

Sending START request to MANAGER ...
EXTRACT P01SRCDB starting

GGSCI (oltp01.oracle-ggs.com) 40> INFO ALL

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     E01SRCDB    00:00:41      00:00:05
EXTRACT     RUNNING     P01SRCDB    00:00:00      00:00:04

GGSCI (oltp01.oracle-ggs.com) 41>
```

Prepare and Define the Replicat

Regardless of the capture mode, the Replicat is configured to read the remote trail and apply the transactions using the database native SQL calls. Use the 'GGSCI EDIT' command to prepare the parameter file. Before defining the Replicat, login to the target database and define the checkpoint table. The final step defines the Replicat, which uses the checkpoint table and is defined globally.

```
GGSCI (oltp02.oracle-ggs.com) 5> EDIT PARAMS R01SRCDB
```

```
REPLICAT R01SRCDB
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/R01SRCDB.dsc, purge
USERID ggs_admin@TGTDB, PASSWORD oracle
MAP scott.*, TARGET scott.*;
```

```
GGSCI (oltp02.oracle-ggs.com) 1> DBLOGIN USERID ggs_admin, PASSWORD oracle
Successfully logged into database.

GGSCI (oltp02.oracle-ggs.com) 2> ADD CHECKPOINTTABLE
```

```
No checkpoint table specified. Using GLOBALS specification (ggs_admin.chkpt)...

Successfully created checkpoint table ggs_admin.chkpt.

GGSCI (oltp02.oracle-ggs.com) 3>
```

```
GGSCI (oltp02.oracle-ggs.com) 3> ADD REPLICAT R01SRCDB, EXTTRAIL ./dirdat/sb
REPLICAT added.

GGSCI (oltp02.oracle-ggs.com) 4>
```

Starting the Replicat

Start and verify the Replicat process. The 'INFO ALL' command indicates the status is RUNNING.

```
GGSCI (oltp02.oracle-ggs.com) 9> START REPLICAT *

Sending START request to MANAGER ...
REPLICAT R01SRCDB starting

GGSCI (oltp02.oracle-ggs.com) 10> INFO ALL

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     R01SRCDB    00:00:00      00:00:02

GGSCI (oltp02.oracle-ggs.com) 11>
```

Insert Data and Run Statistic Report

Test the data movement by inserting on the source database.

```
SQL> INSERT INTO scott.dept VALUES (50, 'GG', 'DUBAI');

1 row created.

SQL> COMMIT;

Commit complete.

SQL>
```

Verify the data is inserted into the target database.

```
SQL> SELECT * FROM scott.dept;

    DEPTNO  DNAME           LOC
----------  --------------   -------------
```

```
         10  ACCOUNTING       NEW YORK
         20  RESEARCH         DALLAS
         30  SALES            CHICAGO
         40  OPERATIONS       BOSTON
         50  GG               DUBAI

SQL>
```

Also, use the 'GGSCI STATS' command to display statistics for INSERT, UPDATE and DELETE DML operations.

```
GGSCI (oltp02.oracle-ggs.com) 7> STATS R01SRCDB

Sending STATS request to REPLICAT R01SRCDB ...

Start of Statistics at 2015-05-03 23:47:52.

Replicating from SCOTT.DEPT to SCOTT.DEPT:

*** Total statistics since 2015-05-03 23:42:59 ***
     Total inserts                                       1.00
     Total updates                                       0.00
     Total deletes                                       0.00
     Total discards                                      0.00
     Total operations                                    1.00

*** Daily statistics since 2015-05-03 23:42:59 ***
     Total inserts                                       1.00
     Total updates                                       0.00
     Total deletes                                       0.00
     Total discards                                      0.00
     Total operations                                    1.00

*** Hourly statistics since 2015-05-03 23:42:59 ***
     Total inserts                                       1.00
     Total updates                                       0.00
     Total deletes                                       0.00
     Total discards                                      0.00
     Total operations                                    1.00

*** Latest statistics since 2015-05-03 23:42:59 ***
     Total inserts                                       1.00
     Total updates                                       0.00
     Total deletes                                       0.00
     Total discards                                      0.00
     Total operations                                    1.00

End of Statistics.

GGSCI (oltp02.oracle-ggs.com) 8>
```

Deploying Integrated Capture on a Downstream System

The following sequence of steps prepares the source database (SRCDB) and the Logmining Database system (mdbs) for an integrated capture deployment. The source database performs a redo transport to the downstream database in preparation for performing a real-time integrated capture.

Prior to starting the deployment of the integrated capture, configure the Logmining database system listener and verify the database is reachable from the source database using the TNSPING utility.

```
[oracle@oltp01 admin]$ tnsping mdbs

TNS Ping Utility for Linux: Version 12.1.0.2.0 - Production on 08-MAY-2015
17:20:45

Copyright (c) 1997, 2014, Oracle.  All rights reserved.

Used parameter files:


Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST =
oltp02.oracle-ggs.com)(PORT = 1521)) (CONNECT_DATA = (SERVER = DEDICATED)
(SERVICE_NAME = mdbs.oracle-ggs.com)))
OK (10 msec)
[oracle@oltp01 admin]$
```

Prepare the Source Database

The integrated capture employs the Streams Pool Memory of the System Global Area (SGA). Allocate an adequate size; The recommended value setting is 700MB. Query the current Stream Pool Size and change it to meet the recommended minimum. When the SGA is dynamically allocated, the Stream Pool Size is an auto-tuned component and the size is based on the current workload.

```
SQL> SELECT NAME, BYTES
  2  FROM   v$sgainfo;

NAME                             BYTES
-------------------------------- ----------
Fixed SGA Size                    2289016
Redo Buffers                      4235264
Buffer Cache Size                 159383552
Shared Pool Size                  138412032
Large Pool Size                   8388608
Java Pool Size                    4194304
Streams Pool Size                 4194304
Shared IO Pool Size               4194304
Data Transfer Cache Size          0
Granule Size                      4194304
```

```
Maximum SGA Size                  413372416

NAME                             BYTES
-------------------------------- ----------
Startup overhead in Shared Pool   113301824
Free SGA Memory Available          92274688

13 rows selected.

SQL> ALTER SYSTEM SET streams_pool_size=700M SCOPE=BOTH;

System altered.

SQL>
```

Prepare the Downstream Database

The downstream Logmining Database system must operate on the archive log mode. The following SQL steps enable the archive log mode and supplemental logging required by the Oracle GoldenGate capture. A new parameter introduced in database 11.2.0.4 is 'ENABLE_GOLDENGATE_REPLICATION.' The parameter default value is 'FALASE', enable the use of XStream API by setting the parameter value to 'TRUE.'

```
SQL> SHUTDOWN IMMEDIATE
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> STARTUP MOUNT
ORACLE instance started.

Total System Global Area  1269366784 bytes
Fixed Size                   2287912 bytes
Variable Size              855639768 bytes
Database Buffers           402653184 bytes
Redo Buffers                 8785920 bytes
Database mounted.
SQL> ALTER DATABASE ARCHIVELOG;

Database altered.

SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;

Database altered.

SQL> ALTER DATABASE FORCE LOGGING;

Database altered.

SQL> ALTER DATABASE OPEN;

Database altered.

SQL> ALTER SYSTEM SWITCH LOGFILE;

System altered.
```

```
SQL> SELECT log_mode, force_logging, supplemental_log_data_min
  2  FROM   v$database;

LOG_MODE     FORCE_LOGGING                       SUPPLEME
------------ ----------------------------------- --------
ARCHIVELOG   YES                                 YES

SQL> ALTER SYSTEM SET enable_goldengate_replication=TRUE SCOPE=BOTH;

System altered.

SQL>
```

Create the Password Files

Use the Oracle utility 'orapwd' to create the password files. The source and target password files must match. For a specific Oracle GoldenGate capture, it's mandatory to use the 'IGNORECASE=Y' option to ignore the password case sensitivity.

```
[oracle@oltp01 dbs]$ cd $ORACLE_HOME/dbs
[oracle@oltp01 dbs]$ orapwd file=orapwSRCDB password=oracle_4U ignorecase=Y
entries=10 force=y
[oracle@oltp01 dbs]$

[oracle@oltp02 ~]$ cd $ORACLE_HOME/dbs
[oracle@oltp02 dbs]$ orapwd file=orapwmdbs password=oracle_4U ignorecase=Y
entries=10 force=y
[oracle@oltp02 dbs]$
```

Configure Redo Transport of the Source Database

Use the database parameter 'LOG_ARCHIVE_DEST_2' to enable the source database to ship the transaction logs to the downstream Logmining Database. For the real-time redo transport, specify the 'SYNC' option. Specify the connect string of the downstream database for the 'SERVICE' attribute. Then, enable the destination using the 'SQL ALTER' command.

```
SQL> ALTER SYSTEM SET
  2  LOG_ARCHIVE_DEST_2='SERVICE=mdbs LGWR SYNC NOREGISTER
  3  VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) REOPEN=15
  4  DB_UNIQUE_NAME=mdbs';

System altered.

SQL> ALTER SYSTEM SET LOG_ARCHIVE_DEST_STATE_2=ENABLE;

System altered.

SQL>
```

Specify the Data Guard Configuration

Add the source and downstream Logmining Database using the database parameters 'LOG_ARCHIVE_CONFIG.'

```
SQL> ALTER SYSTEM SET
  2  LOG_ARCHIVE_CONFIG='DG_CONFIG=(SRCDB, mdbs)';

System altered.

SQL>
```

Determine the Source Database Redo Log Files

This step is needed when configuring the downstream Logmining Database standby redo log groups. The best practice is to create one additional standby redo log group of the same size.

```
SQL> SELECT group#, bytes/1024/1024 bytes
  2  FROM   v$log;

    GROUP#       BYTES
----------  ----------
     1              50
     2              50
     3              50

SQL>
```

Based on the above query result, the downstream Logmining Database needs (4) standby redo log groups — each the size of 50M.

Configure the Directory Structure

Create the following directory structures for the archive, redo, and standby files.

```
[oracle@oltp02 dbs]$ mkdir -p /u01/app/oracle/oradata/mdbs/arch
[oracle@oltp02 dbs]$ mkdir -p /u01/app/oracle/oradata/mdbs/redo
[oracle@oltp02 dbs]$ mkdir -p /u01/app/oracle/oradata/mdbs/standby
```

Configure Downstream Archive locations

The downstream Logmining Database is configured to receive and archive locally and to create the standby redo data. Enable the destinations using the SQL ALTER command.

```
SQL> ALTER SYSTEM SET
  2  log_archive_dest_1='LOCATION=/u01/app/oracle/oradata/mdbs/arch
```

```
  3  VALID_FOR=(ONLINE_LOGFILE,PRIMARY_ROLE)';

System altered.

SQL> ALTER SYSTEM SET
  2  log_archive_dest_2='LOCATION=/u01/app/oracle/oradata/mdbs/standby
  3  VALID_FOR=(STANDBY_LOGFILE,ALL_ROLES)';

System altered.

SQL> ALTER SYSTEM SET log_archive_dest_state_1='ENABLE';

System altered.

SQL> ALTER SYSTEM SET log_archive_dest_state_2='ENABLE';

System altered.

SQL>
```

Specify the Data Guard Configuration

Add the source and downstream Logmining Database using the database parameters 'LOG_ARCHIVE_CONFIG.'

```
SQL> ALTER SYSTEM SET
  2  LOG_ARCHIVE_CONFIG='DG_CONFIG=(SRCDB, mdbs)';

System altered.

SQL>
```

Create the Standby Redo Log Files

The last configuration step of the downstream Logmining Database is creating the standby redo log files. Refer to "Determine the Source Database Redo Log Files" to determine the number and size of standby redo log files, which is calculated as:

- The number of standby redo log file groups is the number of redo log groups in the source database, plus one: 1 + 3 = 4.
- The size of the standby redo log file is at least the same size as the source redo log files: 50MB.

```
SQL> ALTER DATABASE ADD STANDBY LOGFILE GROUP 4
  2  '/u01/app/oracle/oradata/mdbs/redo/stdby1.log' SIZE 50M;

Database altered.

SQL> ALTER DATABASE ADD STANDBY LOGFILE GROUP 5
  2  '/u01/app/oracle/oradata/mdbs/redo/stdby2.log' SIZE 50M;

Database altered.
```

```
SQL> ALTER DATABASE ADD STANDBY LOGFILE GROUP 6
  2  '/u01/app/oracle/oradata/mdbs/redo/stdby3.log' SIZE 50M;

Database altered.

SQL> ALTER DATABASE ADD STANDBY LOGFILE GROUP 7
  2  '/u01/app/oracle/oradata/mdbs/redo/stdby4.log' SIZE 50M;

Database altered.

SQL>
```

Create Database Users for GoldenGate

Database users are needed by Oracle GoldenGate to perform a capture on the source and delivery on the target database. The following illustrates creating and granting a user on the source and downstream Logmining Database.

- Source Database User

```
SQL> CREATE TABLESPACE ggs_dat;

Tablespace created.

SQL> ALTER DATABASE FORCE LOGGING;

Database altered.

SQL> GRANT DBA TO ggs_admin IDENTIFIED BY oracle;

Grant succeeded.

SQL> ALTER USER ggs_admin DEFAULT TABLESPACE ggs_dat
  2                       QUOTA UNLIMITED ON ggs_dat;

User altered.

SQL>
```

- Downstream Logmining Database User

```
SQL> CREATE TABLESPACE ggs_dat;

Tablespace created.

SQL> GRANT DBA TO ggs_admin IDENTIFIED BY oracle;

Grant succeeded.

SQL> ALTER USER ggs_admin DEFAULT TABLESPACE ggs_dat
  2                       QUOTA UNLIMITED ON ggs_dat;

User altered.
```

```
SQL> BEGIN
  2     DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE
  3          (grantee=>'GGS_ADMIN',
  4           privilege_type=>'capture',
  5           grant_select_privileges=>true,
  6           do_grants=>TRUE);
  7  END;
  8  /

PL/SQL procedure successfully completed.

SQL>
```

Database Connections Using Oracle GoldenGate

Perform database connections using Oracle GoldenGate. For integrated capture on a downstream deployment, it requires concurrently connecting to the source and Logmining Databases.

- Login to the source database. Use the 'DBLOGIN' command to login to the database.

```
[oracle@oltp01 oggcore_1]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.


GGSCI (oltp01.oracle-ggs.com) 1> DBLOGIN USERID ggs_admin@srcdb, PASSWORD
oracle
Successfully logged into database.

GGSCI (oltp01.oracle-ggs.com) 2>
```

- Login to the Downstream Logmining Database. Use the 'MINING DBLOGIN' command to login to the database.

```
[oracle@oltp02 oggcore_1]$ ./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 12c on Sep 25 2013 02:33:54
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2013, Oracle and/or its affiliates. All rights reserved.

GGSCI (oltp02.oracle-ggs.com) 2> MININGDBLOGIN USERID ggs_admin@mdbs,
PASSWORD oracle
Successfully logged into mining database.
```

```
GGSCI (oltp02.oracle-ggs.com) 3>
```

Register the Integrated Capture

The Integrated Capture must be registered with the database. Login to the source and the Logmining Databases to perform the Extract registration.

```
[oracle@oltp02 oggcore_2]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 12.1.2.1.2 20133048 20078949_FBO
Linux, x64, 64bit (optimized), Oracle 12c on Apr 17 2015 03:22:09
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2015, Oracle and/or its affiliates. All rights reserved.


GGSCI (oltp02.oracle-ggs.com) 1> DBLOGIN USERID ggs_admin@srcdb, PASSWORD oracle
Successfully logged into database.

GGSCI (oltp02.oracle-ggs.com as ggs_admin@SRCDB) 2> MININGDBLOGIN USERID
ggs_admin@mdbs, PASSWORD oracle
Successfully logged into mining database.

GGSCI (oltp02.oracle-ggs.com as ggs_admin@SRCDB) 3> REGISTER EXTRACT EASRCDB
DATABASE
Extract EASRCDB successfully registered with database at SCN 2678787.


GGSCI (oltp02.oracle-ggs.com as ggs_admin@SRCDB) 4>
```

Verify that the integrated capture has been successfully registered with the Logmining Database. Query the database dictionary view 'DBA_CAPTURE.'

```
SQL> SELECT capture_name, client_name, purpose
  2  FROM   dba_capture;

CAPTURE_NAMN        CLIENT_NAME                    PURPOSE
------------------  -----------------------------  -----------------
OGG$CAP_EASRCDB     EASRCDB                        GoldenGate Capture
OGG$CAP_EXY         EXY                            GoldenGate Capture

SQL>
```

Prepare and Define the Primary Integrated Capture

To enable the integrated capture, use the GoldenGate parameter 'TRANSLOGOPTIONS' with the following options: 'MININGUSER' specifies the database user of the Logmining Database. The 'INTEGERATEDPARAMS' specifies the SGA memory size and the real-time logs mining is enabled by the option 'downstream_real_time_mine' to Y. When defining an integrated capture, specify

the option 'INTEGRATED TRANLOG' of the 'ADD EXTRACT' command. Associate the local trail './dirdat/ia' with the integrated capture using the 'ADD EXTTRAIL' command.

```
GGSCI (oltp02.oracle-ggs.com) 1> EDIT PARAMS EASRCDB
```

```
EXTRACT EASRCDB
USERID ggs_admin@srcdb PASSWORD oracle
TRANSLOGOPTIONS MININGUSER ggs admin@mdbs MININGPASSWORD oracle
TRANSLOGOPTIONS INTEGERATEDPARAMS (max_sga_size 256, downstream_real_time_mine
Y)
EXTTRAIL ./dirdat/ia
TABLE scott.*;
```

```
GGSCI (oltp02.oracle-ggs.com) 2> ADD EXTRACT EASRCDB, INTEGRATED TRANLOG, BEGIN
NOW
EXTRACT added.

GGSCI (oltp02.oracle-ggs.com) 3> ADD EXTTRAIL ./dirdat/ia, EXTRACT EASRCDB,
MEGABYTES 15
EXTTRAIL added.

GGSCI (oltp02.oracle-ggs.com) 4>
```

Prepare and Define the Secondary Capture

As per Oracle GoldenGate best practices suggests, define a Secondary Extract (data pump). The Extract sends data to the Oracle GoldenGate instance specified by the combination of parameters 'RMTHOST' and 'MGRPORT.'

```
GGSCI (oltp02.oracle-ggs.com) 4> EDIT PARAMS PASRCDB
```

```
EXTRACT PASRCDB
USERID ggs_admin@srcdb, PASSWORD oracle
RMTHOST 192.168.1.78, MGRPORT 7809
RMTTRAIL ./dirdat/ib
PASSTHRU
TABLE scott.*;
```

```
GGSCI (oltp02.oracle-ggs.com) 13> ADD EXTRACT PASRCDB, EXTTRAILSOURCE
./dirdat/ib
EXTRACT added.

GGSCI (oltp02.oracle-ggs.com) 14> ADD RMTTRAIL ./dirdat/ib, EXTRACT PASRCDB,
MEGABYTES 15
RMTTRAIL added.

GGSCI (oltp02.oracle-ggs.com) 15>
```

Starting the Extracts

The next step is to start the Extracts 'EASRCDB' (Integrated Capture Extract) and the data pump (PASRCDB).

```
GGSCI (oltp02.oracle-ggs.com) 6> START EXTRACT *

Sending START request to MANAGER ...
EXTRACT EASRCDB starting

Sending START request to MANAGER ...
EXTRACT PASRCDB starting

GGSCI (oltp02.oracle-ggs.com) 7> INFO ALL

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     EASRCDB     unknown       00:00:09
EXTRACT     RUNNING     PASRCDB     00:00:00      00:00:00

GGSCI (oltp02.oracle-ggs.com) 8>
```

Prepare and Define the Replicat

The Replicat is defined in the same way of classic capture. Since there's no transformation, the Replicat RASRCDB uses the parameter 'ASSUMETARGETDEFS.' The MAP statement uses the schema-level rather
than specifying the table-level for identifying the list of target objects.

```
GGSCI (oltp02.oracle-ggs.com) 3> EDIT PARAMS RASRCDB
```

```
REPLICAT RASRCDB
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/RASRCDB.dsc, purge
USERID ggs_admin@TGTDB, PASSWORD oracle
MAP scott.*, TARGET scott.*;
```

```
GGSCI (oltp02.oracle-ggs.com) 5> ADD REPLICAT RASRCDB, EXTTRAIL ./dirdat/ib
REPLICAT added.

GGSCI (oltp02.oracle-ggs.com) 6>
```

Starting the Replicat

Starting the Replicat, the 'INFO ALL' command displays the status of the Replicat RASRCDB.

```
GGSCI (oltp02.oracle-ggs.com) 16> START RASRCDB
```

```
Sending START request to MANAGER ...
REPLICAT RASRCDB starting

GGSCI (oltp02.oracle-ggs.com) 17> INFO ALL

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     RASRCDB     00:00:00      00:00:00

GGSCI (oltp02.oracle-ggs.com) 18>
```

Insert Data and View Statistics Report

Login to the source database. Insert data and view the statistics. The database insert is transformed into the format Logical Change Record (LCR), processed, passed, and processed by the integrated capture.

```
GGSCI (oltp02.oracle-ggs.com) 6> stats EASRCDB

Sending STATS request to EXTRACT EASRCDB ...

Start of Statistics at 2015-05-08 18:24:47.

Output to ./dirdat/ia:

Extracting from SCOTT.DEPT to SCOTT.DEPT:

*** Total statistics since 2015-05-08 18:24:29 ***
     Total inserts                                       1.00
     Total updates                                       0.00
     Total deletes                                       0.00
     Total discards                                      0.00
     Total operations                                    1.00

*** Daily statistics since 2015-05-08 18:24:29 ***
     Total inserts                                       1.00
     Total updates                                       0.00
     Total deletes                                       0.00
     Total discards                                      0.00
     Total operations                                    1.00

*** Hourly statistics since 2015-05-08 18:24:29 ***
     Total inserts                                       1.00
     Total updates                                       0.00
     Total deletes                                       0.00
     Total discards                                      0.00
     Total operations                                    1.00

*** Latest statistics since 2015-05-08 18:24:29 ***
     Total inserts                                       1.00
     Total updates                                       0.00
     Total deletes                                       0.00
     Total discards                                      0.00
```

```
    Total operations                                1.00

End of Statistics.

GGSCI (oltp02.oracle-ggs.com) 7>
```

Migration from a Classic to Integrated Capture

A classic capture can be upgraded to an integrated capture. It's also possible to downgrade an integrated capture to a classic capture. Since classic and integrated captures may co-exist, this feature minimizes the impact (if any) of upgrading to an integrated capture by gradually migrating one schema or set of tables at a time.

Use the steps below to migrate the Extract:

1. Stop the Classic Capture Extract
2. Register the Extract with the Logmining Database
3. Alter the Extract for migration (to classic or integrated)

The following 'GGSCI' commands downgrade (migrate from integrated to classic) the extract then upgrades it again (from classic to integrated).

```
GGSCI (oltp01.oracle-ggs.com) 9> STOP EXTRACT E01SRCDB

Sending STOP request to EXTRACT E01SRCDB ...
Request processed.

GGSCI (oltp01.oracle-ggs.com) 10> DBLOGIN USERID ggs_admin, PASSWORD oracle
Successfully logged into database.

GGSCI (oltp01.oracle-ggs.com) 13> ALTER EXTRACT E01SRCDB, DOWNGRADE INTEGRATED
TRANLOG
Extract E01SRCDB successfully downgraded from integrated capture.

GGSCI (oltp01.oracle-ggs.com) 14> ALTER EXTRACT E01SRCDB, UPGRADE INTEGRATED
TRANLOG
Extract E01SRCDB successfully upgraded to integrated capture.

GGSCI (oltp01.oracle-ggs.com) 15>
```

Restrictions, Limitations, and Patches

Though an integrated capture removes the major limitations found with a classic capture, the following restrictions and limitations should be taken into consideration when using an integrated capture.

- When the Logmining Database name is alphanumeric, the 'GGSCI REGISTER' command fails with error message 'OCI 40,044.'

- When using an integrated capture with Oracle Database 11.2.0.3, apply Oracle database patch 14551959 to enable data types support.

- 'SECUREFILE LOBs' are captured from the redo log only when the LOBs are not compressed, encrypted, or de-duplicated and stored out-of-row. Otherwise, they are captured from the source table.

- Integrated Capture does not support the following data types:

  ANYDATASET
  ANYTYPE
  BFILE
  MLSLABEL
  ORDDICOM
  TIMEZONE_ABBR
  URITYPE/UROWID

Deploying Intermediate Systems Instances

The use of Oracle GoldenGate intermediate system extends Oracle GoldenGate scalability and enhances the source and the target systems performance by moving and consolidating the transformation and the application functions into the intermediate system. In a highly secured distributed database environment,
the intermediate system operates externally and does not connect to the source and target systems as illustrated in Figure 10-7, where data is tunneled through the firewalls.

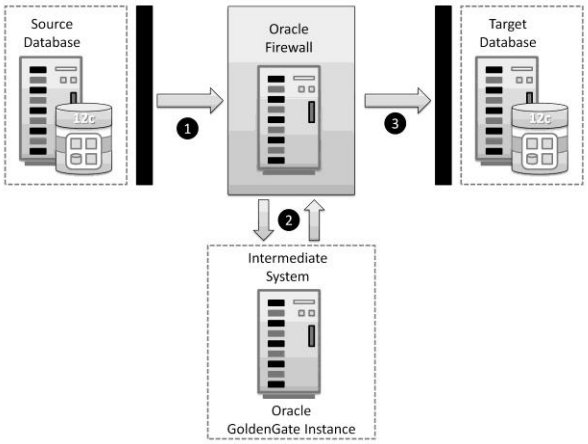


Figure 10-7: Oracle GoldenGate Intermediate System

Figure 10-7 depicts the architecture of a secured distributed environment where firewalls and ports restrictions apply. The Oracle GoldenGate Intermediate System manager processes parameter files must be configured to use an open range of ports specified by the parameter 'PORT' and 'DYNAMICPORTLIST.'

There are (3) options for deploying Oracle GoldenGate instance as an intermediate system.

- No-Database Intermediate System
- Database Intermediate System
- Cascade Topology Intermediate System

No-Database Intermediate System

The *No-Database* option is the most commonly deployed Oracle GoldenGate instance for an intermediate system. It only requires the Oracle Database software installation. The intermediate system continuously receives the stream of trail files shipped by the Primary or Secondary Extract. Through Oracle SQLNET, the Oracle GoldenGate intermediate system instance connects to the source and target databases. This type of intermediate system is suitable when database access is not required for data transformation, where data transformation is strictly performed using Oracle GoldenGate functions without the need to reference the database. Figure 10-8 shows the components for no-database intermediate system performing data transformation using Oracle GoldenGate functions.

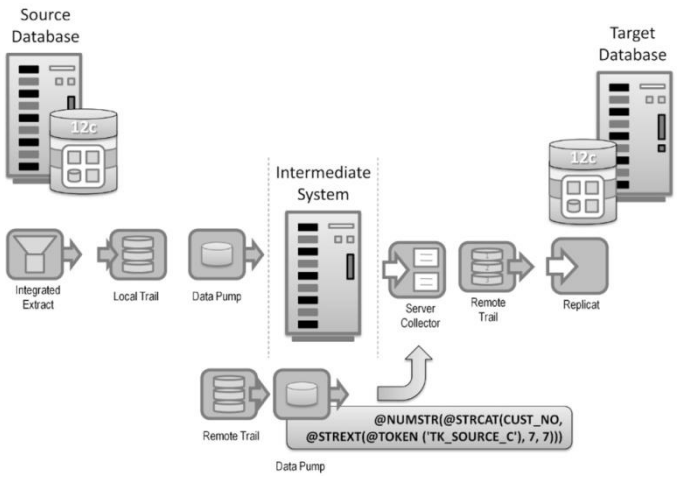


Figure 10-8: No-Database Intermediate System

Database Intermediate System

The *Database* option is applicable to a secured environment operated by independent but related groups. The source database Oracle GoldenGate instance forwards trails to the intermediate system. The intermediate system logs in to its dedicated database for table lookup queries and data transformation. The intermediate database schemas do not store transactional data. Rather, they store the application's lookup tables to translate coded values and form the rows expected by the target database tables. Figure 10-9 shows database an intermediate system performing data transformation using SQL queries.

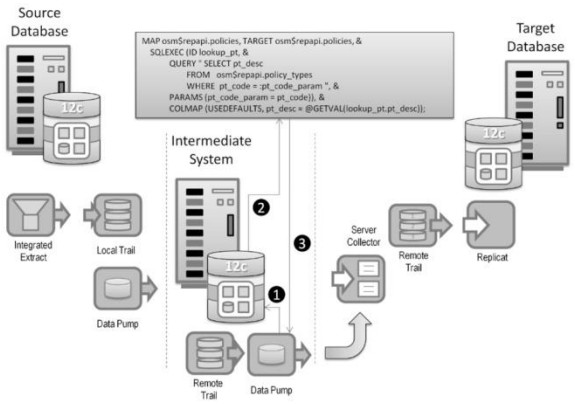


Figure 10-9: Database Intermediate System

Cascade Topology Intermediate System

The *Cascade* option is designed for a large-complex data warehouse environment, where tightly-coupled Oracle GoldenGate instances of intermediate systems are configured between the source and target databases.

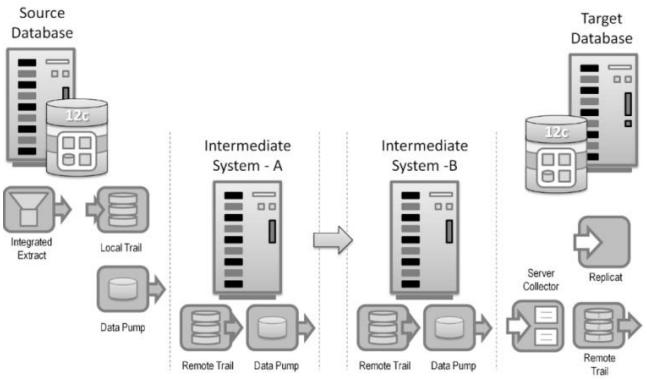


Figure 10-10: Cascade Topology Intermediate System

This type of deployment further promotes scalability by separating the processing across the Oracle GoldenGate instances. For example, the data transformation is performed by intermediate system 'A' and data delivery is performed by intermediate system 'B'. The set of cascaded Oracle GoldenGate instances may be a combination of No-Database and Database Oracle GoldenGate intermediate system instances. Figure 10-10 shows the cascade topology intermediate system.

Deploying No-Database Intermediate System

This section deploys a *No-Database* Intermediate System. The trails received from the source GoldenGate instance are subjected to various data transformations using Oracle GoldenGate functions. The data transformation dynamically changes the ENAME column value to upper case enforced by the target database using the CHECK constraint.

```
SQL> ALTER TABLE scott.emp ADD CONSTRAINT ENAME_UPC
  2  CHECK(ename=upper(ename));

Table altered.

SQL>
```

The deployment is based on the processes workflow depicted by Figure 10-8.

Prepare and Define the Primary Extract

Prepare the Primary Extract parameter file, which is the Extract defined to perform a capture from the online redo log file and write it to the local trail.

```
GGSCI (source1.precisetrace.com) 2> EDIT PARAMS E01S1C1
```

```
EXTRACT E01S1C1
SETENV (ORACLE SID=S1C1)
SETENV (ORACLE_HOME="/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS LANG=AMERICAN AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle SYSDBA
RMTHOST 192.168.1.70, MGRPORT 7810
RMTTRAIL ./dirdat/sb
DISCARDFILE ./dirrpt/E01S1C1.dsc, PURGE
TABLE scott.*;
```

```
GGSCI (source1.precisetrace.com) 5> ADD EXTRACT E01S1C1, TRANLOG, BEGIN NOW
EXTRACT added.

GGSCI (source1.precisetrace.com) 6> ADD EXTTRAIL ./dirdat/sb, EXTRACT E01S1C1,
MEGABYTES 15
EXTTRAIL added.

GGSCI (source1.precisetrace.com) 7>
```

Prepare and Define Level-1 Secondary Extract

Level-1 Secondary Extract forwards the local trail from the source system to the intermediate system. Prepare the Level-1 Secondary Extract parameter file, which is the extract defined to perform a capture from the local trail and write it to the intermediate system remote trail.

```
GGSCI (source1.precisetrace.com) 16> EDIT PARAMS P01S1C1
```

```
EXTRACT P01S1C1
USERID ggs_admin@S1C1, PASSWORD oracle
RMTHOST 192.168.1.70, MGRPORT 7810
RMTTRAIL ./dirdat/si
PASSTHRU
TABLE scott.*;
```

```
GGSCI (source1.precisetrace.com) 8> ADD EXTRACT P01S1C1, EXTTRAILSOURCE
./dirdat/sb
EXTRACT added.

GGSCI (source1.precisetrace.com) 10> ADD RMTTRAIL /dirdat/si, EXTRACT P01S1C1,
MEGABYTES 15
RMTTRAIL added.

GGSCI (source1.precisetrace.com) 11>
```

Prepare and Define Level-2 Intermediate System Secondary Extract

Level-2 Secondary Extract forwards the level-1 remote trail stored at the intermediate system to the target system. Prepare the Level-2 Secondary Extract parameter file, which is the extract defined to perform read from the level-1 remote trail and write it to the target remote trail. The Oracle GoldenGate intermediate system instance is composed of the manager process, remote trail, and the data pump only.

```
GGSCI (t2c1.precisetrace.com) 5> EDIT PARAMS P02S1C1
```

```
EXTRACT P02S1C1
USERID ggs_admin@T1C1, PASSWORD oracle
RMTHOST 192.168.1.152, MGRPORT 7809
RMTTRAIL ./dirdat/tb
PASSTHRU
TABLE scott.*;
```

```
GGSCI (t2c1.precisetrace.com) 3> ADD EXTRACT P02S1C1, EXTTRAILSOURCE ./dirdat/si
EXTRACT added.

GGSCI (t2c1.precisetrace.com) 4> ADD RMTTRAIL ./dirdat/tb, EXTRACT P02S1C1,
MEGABYTES 15
RMTTRAIL added.
```

```
GGSCI (t2c1.precisetrace.com) 5>
```

Prepare and Define the Replicat

Prepare the Replicat parameter file. The Replicat reads the remote trail then applies the transaction using the target database native SQL calls. Another option of delivery is to configure the Replicat local to the intermediate system instance. Connect remotely via SQLNET to the target database for applying the transactions.

```
GGSCI (target1.precisetrace.com) 2> EDIT PARAMS R01S1C1
```

```
REPLICAT R01S1C1
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/R01S1C1.dsc, purge
USERID ggs_admin@T1C1, PASSWORD oracle
MAP scott.*, TARGET scott.*;
```

```
GGSCI (target1.precisetrace.com) 1> DBLOGIN USERID ggs admin, PASSWORD oracle
Successfully logged into database.

GGSCI (target1.precisetrace.com) 2> ADD CHECKPOINTTABLE

No checkpoint table specified. Using GLOBALS specification
(ggs admin.chkpttab)...

Successfully created checkpoint table ggs admin.chkpttab.

GGSCI (target1.precisetrace.com) 3>
```

```
GGSCI (target1.precisetrace.com) 3> ADD REPLICAT R01S1C1, EXTTRAIL ./dirdat/tb
REPLICAT added.


GGSCI (target1.precisetrace.com) 4>
```

Starting Extracts and Replicat

The last step is to start all of the Extracts and Replicat. Begin by starting the Primary & Secondary Extract (data pump)…

```
GGSCI (source1.precisetrace.com) 3> START *

Sending START request to MANAGER ...
EXTRACT E01S1C1 starting

Sending START request to MANAGER ...
EXTRACT P01S1C1 starting

GGSCI (source1.precisetrace.com) 4> INFO ALL
```

```
Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     E01S1C1     00:00:25      00:00:03
EXTRACT     RUNNING     P01S1C1     00:00:00      00:00:23

GGSCI (source1.precisetrace.com) 5>
```

Then, start the intermediate system Secondary Extract (data pump) …

```
GGSCI (t2c1.precisetrace.com) 21> START *

Sending START request to MANAGER ...
EXTRACT P02S1C1 starting


GGSCI (t2c1.precisetrace.com) 22> INFO ALL

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
EXTRACT     RUNNING     P02S1C1     00:00:00      00:00:39


GGSCI (t2c1.precisetrace.com) 23>
```

Lastly, start the Replicat on the target system…

```
GGSCI (target1.precisetrace.com) 9> START *

Sending START request to MANAGER ...
REPLICAT R01S1C1 starting


GGSCI (target1.precisetrace.com) 10> INFO ALL

Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     RUNNING
REPLICAT    RUNNING     R01S1C1     00:00:00      00:00:02


GGSCI (target1.precisetrace.com) 11>
```

Insert Data and Verify Statistics

On the source database, perform the following INSERT statement.

```
SQL> INSERT INTO scott.dept
  2  VALUES (60, 'ODI', 'RESTON');

1 row created.
```

```
SQL> COMMIT;
```

On the target database, perform the following SELECT statement to verify the row has been added to the target database.

```
SQL> SELECT * FROM scott.dept;

    DEPTNO  DNAME                LOC
----------  --------------       --------------
        10  ACCOUNTING           NEW YORK
        20  RESEARCH             DALLAS
        30  SALES                CHICAGO
        40  OPERATIONS           BOSTON
        50  GG                   DUBAI
        60  ODI                  RESTON

6 rows selected.

SQL>
```

Another option is to display Extract and Replicat statistics. The following statement displays the Replicat Statistics and comprehensive DML statistics.

```
GGSCI (target1.precisetrace.com) 3> STATS R01S1C1

Sending STATS request to REPLICAT R01S1C1 ...

Start of Statistics at 2015-05-11 01:14:45.

Replicating from SCOTT.DEPT to SCOTT.DEPT:

*** Total statistics since 2015-05-11 01:09:11 ***
    Total inserts                                       2.00
    Total updates                                       0.00
    Total deletes                                       0.00
    Total discards                                      0.00
    Total operations                                    2.00

*** Daily statistics since 2015-05-11 01:09:11 ***
    Total inserts                                       2.00
    Total updates                                       0.00
    Total deletes                                       0.00
    Total discards                                      0.00
    Total operations                                    2.00

*** Hourly statistics since 2015-05-11 01:09:11 ***
    Total inserts                                       2.00
    Total updates                                       0.00
    Total deletes                                       0.00
    Total discards                                      0.00
    Total operations                                    2.00

*** Latest statistics since 2015-05-11 01:09:11 ***
    Total inserts                                       2.00
```

```
    Total updates                                    0.00
    Total deletes                                    0.00
    Total discards                                   0.00
    Total operations                                 2.00

End of Statistics.

GGSCI (target1.precisetrace.com) 4>
```

Data Definition Language Replications

In addition to replication of DML transactions, Oracle GoldenGate supports the replication of Data Definition Language (DDL) statements. This feature enables a dynamic in-house development team to rollout changes to a database objects definition by means of parameters and options for fine grain control and filtering of DDL statements from the source to one or more target databases.

By default, the Primary Extract has the DDL synchronization disabled, where as the Replicat has the DDL replication enabled. The Extract DDL synchronization is explicitly activated by the inclusion of the DDL parameter in the same parameter file for DML synchronization. The Secondary Extract (data pump) applies 'PASSTHRU,' and the Replicat applies 'ASSUMETARGETDETFS' parameters. DDL statements are limited to a length of 2MB. Larger statements are skipped by the Extract.

Filtering in which DDL statements are replicated is done by extending the Extract or Replicat DDL parameter to include conditions that must be stratified to capture the corresponding DDL statements. The conditions are evaluated using the 'AND' logical operator for the options as operands.

Supported Database Object Types

Oracle GoldenGate supports DDL statements replication of the major database object types. These are the objects types commonly utilized by the database applications.

- Tables
- Indexes
- Clusters
- Packages
- Procedures
- Synonyms
- Triggers
- Types
- Views
- Materialized Views
- Database Users

- Tablespaces
- Roles

Oracle Database 12c has 44 object types, only subsets of them are supported.

```
SQL> SELECT unique object_type
   2 FROM   dba_objects
   3 ORDER BY 1;
```

Activating DDL Statements Replication

For Oracle Database 12c, none of the steps below are required. However, Oracle Database 10g and Oracle Database 11g activation of DDL statements replication requires executing the following sequence of steps.

Disable the Database's Recyclebin Support

```
SQL> ALTER SYSTEM SET recyclebin=off SCOPE=SPFILE;

System altered.

SQL> STARTUP FORCE
ORACLE instance started.

Total System Global Area          1068937216 bytes
Fixed Size                           2296576 bytes
Variable Size                      973079808 bytes
Database Buffers                    88080384 bytes
Redo Buffers                         5480448 bytes
Database mounted.
Database opened.
SQL>
```

Create DDL Statements Replication Tables Support

Navigate to the Oracle GoldenGate instance software location. Execute the scripts 'maker_setup.sql and ddl_setup.sql.' When prompted, enter the Oracle GoldenGate database schema. Upon successful completion, it creates the following tables:

- GGS_MARKER
- GGS_DDL_RULES
- GGS_DDL_RULES_LOG
- GGS_DDL_HIST_ALT
- GGS_DDL_HIST

```
SQL> @marker_setup.sql

Marker setup script
```

```
You will be prompted for the name of a schema for the Oracle GoldenGate database
objects.
NOTE: The schema must be created prior to running this script.
NOTE: Stop all DDL replication before starting this installation.

Enter Oracle GoldenGate schema name:ggs_admin


Marker setup table script complete, running verification script...
Please enter the name of a schema for the GoldenGate database objects:
Setting schema name to GGS_ADMIN

MARKER TABLE
-------------------------------
OK

MARKER SEQUENCE
-------------------------------
OK

Script complete.
SQL>
```

```
SQL> @ddl_setup.sql

Oracle GoldenGate DDL Replication setup script

Verifying that current user has privileges to install DDL Replication...

You will be prompted for the name of a schema for the Oracle GoldenGate database
objects.
NOTE: For an Oracle 10g source, the system recycle bin must be disabled. For
Oracle 11g and later, it can be enabled.
NOTE: The schema must be created prior to running this script.
NOTE: Stop all DDL replication before starting this installation.

Enter Oracle GoldenGate schema name:ggs_admin

Working, please wait ...
Spooling to file ddl_setup_spool.txt

Checking for sessions that are holding locks on Oracle Golden Gate metadata
tables ...

Check complete.

Using GGS_ADMIN as a Oracle GoldenGate schema name.

Working, please wait ...

DDL replication setup script complete, running verification script...
Please enter the name of a schema for the GoldenGate database objects:
Setting schema name to GGS_ADMIN

CLEAR_TRACE STATUS:

Line/pos   Error
---------- -------------------------------------------------------------------
```

```
No errors  No errors

LOCATION OF DDL TRACE FILE
--------------------------------------------------------------------------------
/u01/app/oracle/diag/rdbms/s1c1/S1C1/trace/ggs_ddl_trace.log

Analyzing installation status...

VERSION OF DDL REPLICATION
--------------------------------------------------------------------------------
OGGCORE_12.1.2.0.0_PLATFORMS_130924.1316

STATUS OF DDL REPLICATION
--------------------------------------------------------------------------------
SUCCESSFUL installation of DDL Replication software components

Script complete.
SQL>
```

Create Database Role and Grant Privileges

Create the database role 'GGS_GGSUSER_ROLE' required for DDL statements replication. Then, grant the role to the Oracle GoldenGate database schema.

```
SQL> @role_setup.sql

GGS Role setup script

This script will drop and recreate the role GGS_GGSUSER_ROLE
To use a different role name, quit this script and then edit the params.sql
script to change the gg_role parameter to the preferred name. (Do not run the
script.)

You will be prompted for the name of a schema for the GoldenGate database
objects.
NOTE: The schema must be created prior to running this script.
NOTE: Stop all DDL replication before starting this installation.

Enter GoldenGate schema name:ggs_admin
Wrote file role_setup_set.txt

PL/SQL procedure successfully completed.


Role setup script complete

Grant this role to each user assigned to the Extract, GGSCI, and Manager
processes, by using the following SQL command:

GRANT GGS_GGSUSER_ROLE TO <loggedUser>

where <loggedUser> is the user assigned to the GoldenGate processes.

SQL> grant ggs_ggsuser_role to ggs_admin;

Grant succeeded.
```

Enable DDL Replication Support

The last step is to enable DDL statement replication support and pin the package into the database shared pool area.

```
SQL> @ddl_enable.sql

Trigger altered.

SQL> @ddl_pin ggs_admin

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

PL/SQL procedure successfully completed.

SQL>
```

DDL Parameter Syntax and Options

By default, DDL statements replication are unconstrained. Optional conditions are applied to control and filter the replication. If the DDL parameter's options are included, the DDL statements must satisfy all conditions to be replicated. The DDL parameter has the following syntax:

```
DDL [
 { INCLUDE | EXCLUDE }

  [, MAPPED | UNMAPPED | OTHER | ALL]
  [, OPTYPE ]
  [, OBJTYPE '']
  [, OBJNAME ]
  [, INSTR '']
  [, INSTRCOMMENTS '']
  [, STAYMETADATA]
  [, EVENTACTIONS {}
]
```

Table 10-1 lists the DDL parameter's options.

DDL Option	Description
Include and/or Exclude	Identify what to include and what to exclude when performing DDL statement replication
Mapped, Unmapped, Other R and All	Applying include and exclude according to the DDL operation scope.
OpType	Applying include and exclude to specific operation types.
ObjType	Applying include and exclude to specific object types.

ObjName	Applying include and exclude to matching object name.
InStr	Applying include or exclude to DDL statements based on matching characters of the DDL string.
InstrComments	Applying include or exclude to DDL statements that contain a matching characters of string within a comment.

Table 10-1: DDL Parameter Options

Strings Replacement

The string replacement allows the replacement of a source string with another string on the target. The following example illustrates the substitution of the schema 'SCOTT' with 'GGS_ADMIN' when performing the 'CRATE TABLE DDL' statement:

```
REPLICAT R01S1C1
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/R01S1C1.dsc, purge
DDL include ALL
DDLSUBST 'SCOTT' WITH 'GGS_ADMIN'
DDLERROR DEFAULT IGNORE RETRYOP
USERID ggs_admin@T1C1, PASSWORD oracle
MAP scott.*, TARGET scott.*;

SQL> CREATE TABLE scott.cust(cust_name VARCHAR2(40) PRIMARY KEY);

Table created.

SQL>
```

Replicated on the target under the schema GGS_ADMIN.

```
SQL> DESC ggs_admin.cust
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
 CUST_NAME                                 NOT NULL VARCHAR2(40)

SQL>
```

The Extract and the Replicat Parameters Files

The Extract parameters file may performs filtering using the object type option. Only the object of table type and index are replicated. The Replicat application of the DDL statement on the target database is unchanged. The filtering may be performed on either side.

```
EXTRACT E01S1C1
```

```
SETENV (ORACLE_SID=S1C1)
SETENV (ORACLE_HOME="/u01/app/oracle/product/12.1.0/dbhome_1")
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
USERID ggs_admin, PASSWORD oracle
EXTTRAIL ./dirdat/sb
DDL include ALL
DISCARDFILE ./dirrpt/E01S1C1.dsc, PURGE
TABLE scott.*;
REPLICAT R01S1C1
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/R01S1C1.dsc, purge
DDL include ALL
DDLERROR DEFAULT IGNORE RETRYOP
USERID ggs_admin@T1C1, PASSWORD oracle
MAP scott.*, TARGET scott.*;
```

Replication Examples of DDL Statements

The following are examples for demonstrating DDL replication from a source to target system. The DDL statement is executed on the source database then verified on the target database.

Tables Replication

The table 'SCOTT.PROFILES' is created on the source database. Since the Extract is configured to include all types of objects, it replicates the table DDL statement on the target database.

```
SQL> CREATE TABLE scott.profiles(
  2  pname  VARCHAR2(32) PRIMARY KEY,
  3  pemail VARCHAR2(64),
  4  paddr  VARCHAR2(128));

Table created.

SQL>
```

Connect to target to verify the table has been replicated successfully.

```
SQL> DESC scott.profiles
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 PNAME                                     NOT NULL VARCHAR2(32)
 PEMAIL                                             VARCHAR2(64)
 PADDR                                              VARCHAR2(128)

SQL>
```

Oracle GoldenGate DDL statements replication supports Index Organized Table (IOT) and the ALTER TABLE statement.

Indexes Replication

While connected to the source database, create an index on table 'SCOTT.PROFILES' for the column 'PEMAIL'.

```
SQL> CREATE INDEX scott.pemail_idx
  2  ON scott.profiles(pemail);

Index created.

SQL>
```

Connect to the target database and verify the index has been replicated successfully.

```
SQL> SELECT index_name
  2  FROM   dba_indexes
  3  WHERE  owner='SCOTT' and table_name='PROFILES';

INDEX_NAME
----------------------------
PEMAIL_IDX
SYS_C009867

SQL>
```

Clusters Replication

The purpose of clusters is to provide an efficient mechanism to store data blocks and to enhance the data retrieval (queries). Create a cluster on the target database then verify the replication on the target.

```
SQL> CREATE CLUSTER scott.dept no(deptno NUMBER);

Cluster created.

SQL>
```

Connect to the target database and query the data dictionary view 'DBA_CLUSTERS.'

```
SQL> SELECT cluster name
  2  FROM   dba_clusters
  3  WHERE  owner='SCOTT';

CLUSTER_NAME
--------------------
DEPT NO

SQL>
```

Procedures Replication

A procedure is an exactable PL/SQL code stored in the database. The following programs create a procedure called 'GETEMAIL.'

```
SQL> CREATE OR REPLACE PROCEDURE scott.getemail(Ppname   IN VARCHAR2,
  2                                             Ppemail OUT VARCHAR2) IS
  3      vpemail VARCHAR2(64);
  4  BEGIN
  5     SELECT pemail into vpemail
  6     FROM   scott.profiles
  7     WHERE  pname =ppname;
  8  END;
  9  /

Procedure created.

SQL>
```

Connect to the target database to verify it has been created successfully.

```
SQL> DESC scott.getemail
PROCEDURE scott.getemail
 Argument Name                  Type                    In/Out Default?
 ------------------------------ ----------------------- ------ --------
 PPNAME                         VARCHAR2                IN
 PPEMAIL                        VARCHAR2                OUT

SQL>
```

Synonyms Replication

From the source database create a synonym for 'SCOTT.PROFILES.'

```
SQL> CREATE PUBLIC SYNONYM profiles FOR scott.profiles;

Synonym created.

SQL>
```

Query the data dictionary view 'ALL_SYNONYMS' to verify it has been created successfully.

```
SQL> SELECT SYNONYM_NAME
  2  FROM   all synonyms
  3  WHERE  table_owner='SCOTT' AND table_name='PROFILES';

SYNONYM_NAME
-----------------------
PROFILES

SQL>
```

Views Replication

Views Replication is a result set of a query. The views enable database users to access one or more base tables through the views. Create the view 'EMP_VW_10' which stores the result set of department 10 only.

```
SQL> CREATE VIEW scott.emp_vw_90
  2  AS
  3  SELECT *
  4  FROM   scott.emp
  5  WHERE  deptno=10;

View created.

SQL>
```

Verify the view has been successfully created on the target database.

```
SQL> DESC scott.emp_vw_90
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 EMPNO                                     NOT NULL NUMBER(4)
 ENAME                                              VARCHAR2(10)
 JOB                                                VARCHAR2(9)
 MGR                                                NUMBER(4)
 HIREDATE                                           DATE
 SAL                                                NUMBER(7,2)
 COMM                                               NUMBER(7,2)
 DEPTNO                                             NUMBER(2)

SQL>
```

Database Users Replication

Create a new user on the source database then grant 'CREATE SESSION' to the user.

```
SQL> CREATE USER ggs2_admin IDENTIFIED BY oracle;

User created.

SQL> GRANT CREATE SESSION TO ggs2_admin;

Grant succeeded.

SQL>
```

Navigate to the target system. Connect using the database username created on the source database.

```
SQL> CONN ggs2_admin/oracle
```

```
Connected.
SQL>
```

Tablespaces Replication

Create an Oracle Managed File (OMF) tablespace 'GGS2_DAT' on the source system.

```
SQL> CREATE TABLESPACE ggs2 dat;

Tablespace created.

SQL>
```

On the target system, query the data dictionary 'DBA_TABLESPACES' to verify the tablespace has been successfully created.

```
SQL> SELECT tablespace_name
  2  FROM   dba_tablespaces;

TABLESPACE NAME
------------------------------
SYSTEM
SYSAUX
UNDOTBS1
TEMP
USERS
GGS2_DAT

6 rows selected.

SQL>
```

Roles Replication

Create a database role 'SEL_ON_EMP' on the source database. Grant the role basic object privilege.

```
SQL> CREATE ROLE sel on emp;

Role created.

SQL> GRANT SELECT ON scott.emp TO sel on emp;

Grant succeeded.

SQL>
```

Connect to the target database. Grant the role another object privileges, which verifies the role of the DDL statement has been replicated successfully.

```
SQL> GRANT UPDATE ON scott.emp TO sel_on_emp;

Grant succeeded.

SQL>
```

Summary

This last chapter of the book has covered integration-specific techniques, normally found in large scale environments. The integration of Oracle GoldenGate with Oracle database features extends the functionalities of applications and maximizes the Oracle GoldenGate Return of Investment (ROI). Though the chapter has covered specific integration areas, Oracle GoldenGate is engineered to support highly complex infrastructures, such as ATM networks, legacy systems, and various adapters.

Since Oracle customers are using Oracle Data Guard for building their disaster recovery (DR) site, Oracle GoldenGate allows these customers to synchronize a second read-write copy from the read-only standby database. This database can be used for different projects including reports and business intelligence.

Oracle's decision of not enhancing Oracle Streams 11.2 announced following the acquisition of Oracle GoldenGate. Hence, customers should plan the migration from Oracle Streams to Oracle GoldenGate. The feature sets delivered by Oracle GoldenGate enable customers to extend their distributed applications to support maximum Heterogeneities, a key feature not available from Oracle Streams.

Oracle GoldenGate Intermediate System is designed to maximize the scalability of a complex environment consisting of a large number of targets. The intermediate system becomes the data hub for distributing data to meet the target system's requirements by performing data transformation prior to sending the data.

Oracle GoldenGate 11.2 introduced the Integrated Capture option for Oracle Database only. It eliminates installing Oracle GoldenGate on the production systems. Instead, Oracle GoldenGate is installed on a downstream system receiving the database transaction logs, performing the capture, and creating the trails for the next systems. The integrated capture option is suitable for a highly secured environment.

Data Definition Language (DDL) is useful for application with frequent changes and ad-hoc deployments. Database objects such as tablespaces, tables, indexes, users, etc. are replicated. DDL replication is flexible for enhanced security and database objects filtering.

And lastly, Oracle GoldenGate 12c is engineered to be highly integrated with advanced database features, in particular Oracle Real Application Clusters (RAC) and Oracle Data Guard (DG). Oracle GoldenGate is positioned to complement these database options and not to replace them. Adapting state-of-the-art transformation capabilities on heterogeneous environment, and it continues support for the major database vendors: Oracle, Microsoft, IBM, Teradata, HP and open source (MySQL).

Oracle Corporation, World Headquarters
Oracle Parkway 500
Redwood Shores, CA 94065, USA

www.oracle.com

Worldwide Inquiries

Phone: +1.650.506.7000
Fax: +1.650.506.7200



Copyright © 2014, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

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. 0315

Connect with us

facebook.com/oracledataintegration
twitter.com/orclgoldengate
blogs.oracle.com/dataintegration
oracle.com/goto/oraclegoldengate