Oracle GoldenGate Hands-on Tutorial

Version Date: 1st Oct, 2012

Editor: Ahmed Baraka

Document Purpose

This document is edited to be a quick how-to reference to set up and administer Oracle GoldenGate. It presents the topic in hands-on tutorial approach. Usually, no details explanation is presented.

The document is simply oriented based on the required task, the code to perform the task and any precautions or warnings when using the code.

The examples in this document were executed on virtual machine-based systems.

Prerequisites

The document assumes that the reader has already the basic knowledge of RDBMS and replication concepts.

GoldenGate Version

Examples in this document were developed and tested using Oracle GoldenGate version 11.2.

How to Use the Document

1. Go to Contents section
2. Search the required task
3. Click on the required task link

Obtaining Latest Version of the Document

Latest version can be obtained from my site or by emailing me at info@ahmedbaraka.com

Acronyms Used in the Document

GG Oracle GoldenGate.

Oracle GoldenGate is the official name of the software. However, "GoldenGate" term is used through this document for simplicity.

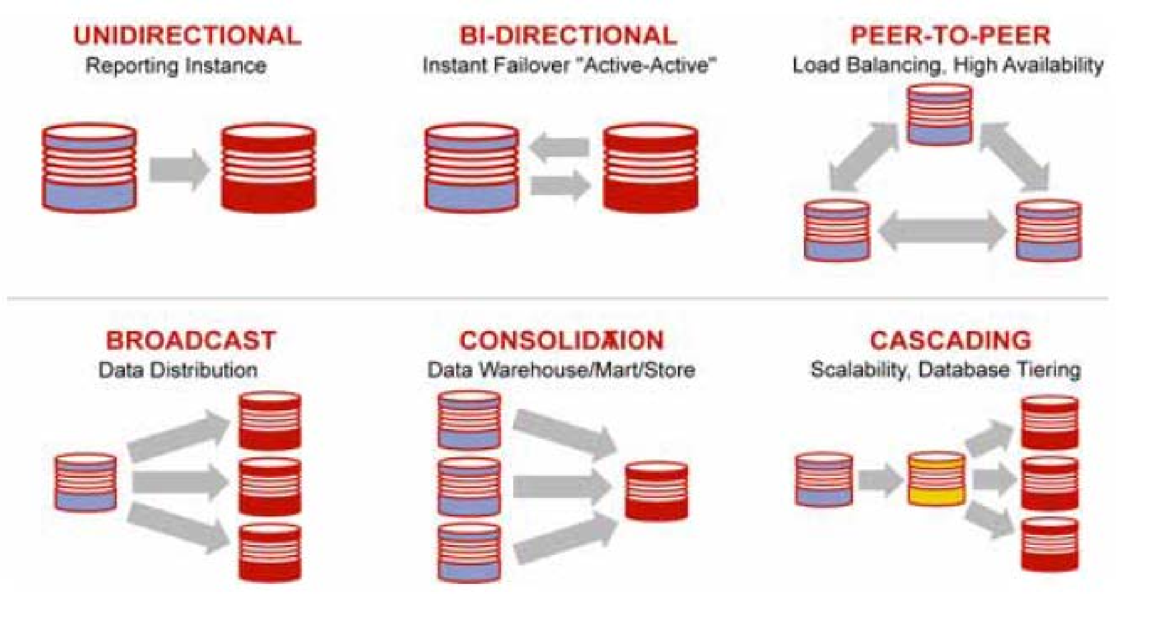
Usage Terms

- Anyone is authorized to copy this document to any means of storage and present it in any format to any individual or organization for *non-commercial* purpose free.
- No individual or organization may use this document for *commercial* purpose without a written permission from the editor.
- This document is for informational purposes only, and may contain typographical errors and technical inaccuracies.
- **There is no warranty of any type for the code or information presented in this document. The editor is not responsible for any loses or damage resulted from using the information or executing the code in this document.**
- If any one wishes to correct a statement or a typing error or add a new piece of information, please send the request to **info@ahmedbaraka.com**

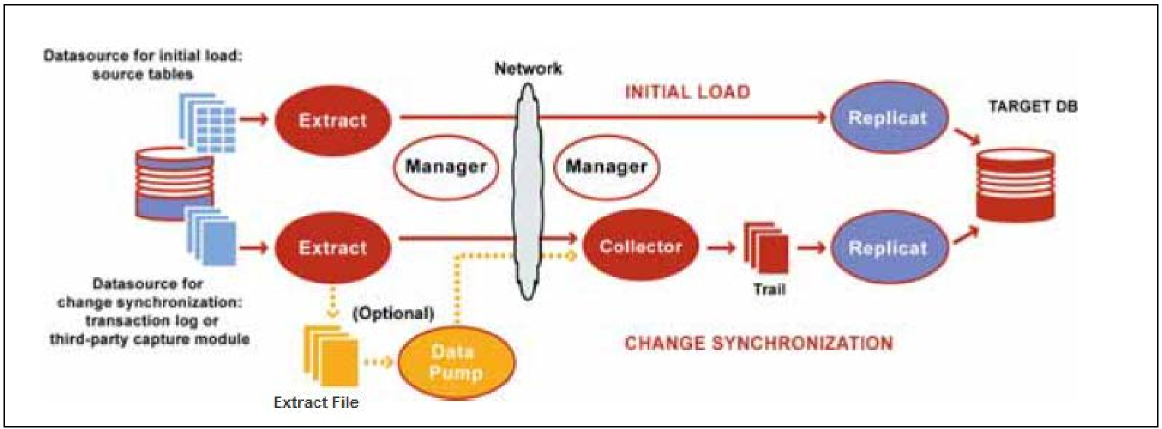
Contents

Golden Gate Components

Supported Topologies


UNIDIRECTIONAL
Reporting Instance
BI-DIRECTIONAL
Instant Failover "Active-Active"
PEER-TO-PEER
Load Balancing, High Availability
BROADCAST
Data Distribution
CONSOLIDATION
Data Warehouse/Mart/Store
CASCADING
Scalability, Database Tiering

GoldenGate Components



- If a data pump is not used, Extract must send the captured data operations to a remote trail on the target.
- Data pump and replicate can perform data transformation.
- You can use multiple Replicate processes with multiple Extract processes in parallel to increase throughput.
- Collectors is by default started by Manager (dynamic Collector), or it can be manually created (static). Static Collector can accept connections from multiple Extracts.

Supported Source Databases

- c-tree
- DB2 for Linux, UNIX, Windows
- DB2 for z/OS
- MySQL
- Oracle
- SQL/MX
- SQL Server
- Sybase
- Teradata

Supported Target Databases

- c-tree
- DB2 for iSeries
- DB2 for Linux, UNIX, Windows
- DB2 for z/OS
- Generic ODBC
- MySQL
- Oracle
- SQL/MX
- SQL Server
- Sybase
- TimesTen

Process Types

- Online Extract or Replicate
- Source-is-table Extract
- Special-run Replicate
- A remote task is a special type of initial-load process in which Extract communicates directly with Replicate over TCP/IP.

Commit Sequence Number

A Commit Sequence Number (CSN) is an identifier that GoldenGate constructs to identify a transaction for the purpose of maintaining transactional consistency and data integrity.

Replication Design Considerations

When planning for replication project, take the following into considerations:

- Source and target configurations
    - Are the databases supported?
    - How is the storage configured?
    - Are the servers stand-alone or clustered?

    - What are the charactersets used?

- Data currency/latency

- Data volume including size of changing data

- Data requirements
    - Are there any triggers or Sequences?
    - Is there any unsupported datatype?

- Security requirements
    - Should the data be encrypted in trail files?
    - Is there any TDE data?

- Network requirements
    - Are there any firewalls that need to be opened for the replication?
    - What is the distance between the source and target?
    - How much bandwidth is required to handle the data volume, and how much is available?

Loading with GoldenGate Options

To perform the initial load with GoldenGate:

- File Load: The Extract writes out the initial load records in a format that can be processed by DBMS utilities like Oracle SQL*Loader and the SQL Server Bulk Copy Program (BCP).
- Direct Load: GoldenGate extracts the data from the source and sends it directly to a Replicate task on the target server.

Note: Still, it is not necessary to use GoldenGate for initial loading anyway. You can SQL commands with database link, exp and imp utilities, transportable tablespaces… etc.

General Installation Requirements

Supported Databases

- The official reference is the version matrix provided by Oracle.

- Supported Oracle databases:
  - Oracle 9i R2
  - Oracle 10g R1 and R2
  - Oracle 11g R1 and R2
  - Microsoft SQL Server 2008

Note: It is not stated in the version matrix document issued by Oracle for GoldenGate 11.2 that versions 2000 and 2005 are supported. But they are there in the same document for GoldenGate 11.1. However, in the installation guide of GoldenGate 11.2, it is stated that SQL Server versions 2000 and 2005 are supported by GoldenGate 11.2.

  - IBM DB2 9.1, 9.5 and 9.7
  - Sybase 15.0 and 15.5
  - SQL/MX 2.3 and 3.1

- Supported Operating Systems:
  - Windows Server 2008 R2
  - Windows Server 2008 with SP1+
  - Windows 2003 with SP2+/R2+
  - Windows 2003
  - Oracle Linux 6 (UL1+)
  - Oracle Linux 5 (UL3+)
  - Oracle Linux 4 (UL7+)
  - Red Hat EL 6 (UL1+)
  - Red Hat EL 5 (UL3+)
  - Red Hat EL 4 (UL7+)
  - Solaris 11
  - Solaris 10 Update 6+
  - Solaris 10 Update 4+
  - Solaris 2.9 Update 9+
  - HP-UX 11i
  - AIX 7.1 (TL2+)
  - AIX 6.1 (TL2+)
  - AIX 5.3 (TL8+)
  - zOS 1.08 , 1.09 , 1.10 , 1.11 and 1.12

Memory Requirements

- At least between 25 and 55 Mb of RAM memory is required for each GoldenGate Replicate and Extract process.
- Oracle GoldenGate supports up to 300 concurrent processes for Extract and Replicate per GoldenGate instance.

Disk Space Requirements

Item	Disk Space Size required	Notes
Software Binaries	50-150 MB	
temporary files	10 GB	dirtmp directory under the installation directory

trail file	[log volume in one hour] x [number of hours downtime] x 0.4	In Oracle database, you can use the following query:
		`select trunc(COMPLETION_TIME),count(*)*100 size_in_MB`
		`from v$archived_log`
		`group by trunc(COMPLETION_TIME);`

OS Requirements

- In Windows platforms, the software "Microsoft Visual C ++ 2005 SP1 Redistributable Package" is required.
- The system account must have read/write access to Oracle database software, as well as to the Oracle online redo log files.
- For Itanium you must install the vcredist_IA64.exe runtime library package.
- In a clustered environment, GG software directory should be accessible to all nodes.
- In Oracle RAC: all GoldenGate software binary and executable files, trail files, and log files should be on shared storage accessible by all cluster nodes.

Installing GoldenGate Software

Installing GoldenGate on Windows

Reference: Oracle GoldenGate Oracle Installation and Setup Guide 11g Release 2 Patch Set (11.2.1.0.1) E29797-01

1. Get and read Release Notes.

2. Download and install the **Microsoft Visual C++ 2005 SP1** Redistributable on both the source and destination servers

3. Create new directory for the GG software:

   Note: Do not install GoldenGate into a folder that contains spaces in its name.

   For example:
   `C:\mkdir ggs_src`

4. Extract the GG software into the new folders

5. In gg home directory:

```
c:\ggs_src\ggsci
create subdirs
exit
```

6. Install the Manager process as a Windows Service (recommended)

By default, the Manager process (its default name is GGSMGR) is not installed as a service and can be run by a local or domain account. Therefore, it will stop when the user logs out.

To install the Manager process as a service:

6.1. (Recommended) Log on as the system administrator.
6.2. In command prompt, from the directory that contains the Manager program:

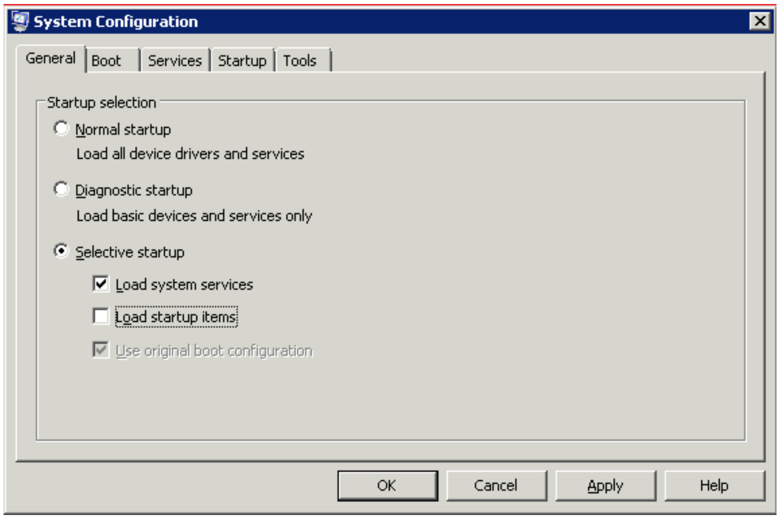
```
-- create the file mgr.prm if it doesn't exist
notepad E:\ggs_src\dirprm\mgr.prm
OR
GGSCI (WIN11SRC) 2>edit params mgr

-- add the following to it: ( or 7840 )
port 7809

-- recommended also:
BOOTDELAYMINUTES 3
autostart ER *
PURGEOLDEXTRACTS dirdat/*, USECHECKPOINTS, MINKEEPDAYS 3


-- install the Manager as a service:
E:\ggs_src>install ADDSERVICE AUTOSTART
Service 'GGSMGR' created.
Install program terminated normally.

-- start it in ggsci:
```

```
GGSCI (WIN11SRC) 1>start manager
Starting Manager as service ('GGSMGR')...
Service started.

GGSCI (WIN11SRC) 2> info manager
Manager is running (IP port WIN11SRC.7809).
```

Note: In, Windows Vista and Windows 2008 security you may receive the following error message when you try to start the Manager:

Process creation error: WIN32 API CALL Create Process failed 740 (The requested operation requires elevation.)

To resolve it: run the msconfig utility and set it as follows:



Startup selection

○ Normal startup
Load all device drivers and services

○ Diagnostic startup
Load basic devices and services only

● Selective startup

☑ Load system services

□ Load startup items

☑ Use original boot configuration

Installing GoldenGate on Linux

Unzip the files for Linux and UNIX platforms. Then run the ggsci command CREATE SUBDIRS as we performed previously for Windows.

Uninstalling GoldenGate

- It is typically the reverse procedure of installing the software.

```
-- stop all gg processes:
cd /home/oracle/ggs
ggsci

stop EXTRACT *
stop REPLICAT *
stop MGR

-- delete the process groups (if there is any)
dblogin userid ggs_owner password g
delete EXTRACT *
delete REPLICAT *

exit


-- remove GGMGR service and stop logging into the Windows Event Log
E:\ggs\install deleteevents deleteservice


-- in Windows, delete two gg related DLL files (I didn't see them in gg11gr2):
del c:\windows\system32\CATEGORY.DLL
del c:\windows\system32\GGSMSG.DLL


-- remove the gg home directory:
cd ..
rm -rf ggs


-- log in as sys and drop the gg owner user:
sqlplus / as sysdba
SQL> drop user ggs_owner cascade;
```

Preparing a Database for GoldenGate

After installing GoldenGate software, you need to configure the databases used by GoldenGate.

Preparing Oracle Database for GoldenGate

- **Reference**: Oracle GoldenGate Oracle Installation and Setup Guide 11g Release 2 Patch Set (11.2.1.0.1) E29797-01

1. Create GoldenGate schema:

```
create tablespace ggs_data
 datafile 'E:\ORADATA\ORADB\ggs_data.dbf' size 200m
 autoextend on next 10m maxsize 4g;

create user ggs_owner identified by g
 default tablespace ggs_data
 temporary tablespace temp;

grant connect, resource to ggs_owner;
grant select any dictionary, select any table to ggs_owner;
grant create table to ggs_owner;
grant flashback any table to ggs_owner;
grant execute on dbms_flashback to ggs_owner;
grant execute on utl_file to ggs_owner;
grant create any table to ggs_owner;
grant insert any table to ggs_owner;
grant update any table to ggs_owner;
grant delete any table to ggs_owner;
grant drop any table to ggs_owner;
```

2. Database must be running in archivelog mode:

```
-- to check it the database is running in archivelog mode:
select log_mode from v$database ;


-- to make it running in archivelog mode:
shutdown immediate startup mount
alter database archivelog;
```

3. In the database operating as a source: verify database-Level supplemental logging

```
-- it must be YES or IMPLICIT
SELECT SUPPLEMENTAL_LOG_DATA_MIN FROM V$DATABASE;

-- to enable it:
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
```

4. The following should be carried out to support DDL replication (if needed):

Note: To replicate just sequence values, you do not need to install the Oracle GoldenGate DDL support environment. You can just use the SEQUENCE parameter.

```
1. In both Source and Target databases:
```

```
-- create a file named as GLOBALS in the GoldenGate directory, if one does not
--  exist
E:\ggs_src>notepad GLOBALS

-- add the following into the GLOBALS file:
GGSCHEMA ggs_owner



2. Exit all users sessions to Oracle database. Prevent the start of any new
session



3. In the GoldenGate directory, run SQL*Plus and log in as SYSDBA:
-- Run the scripts: @marker_setup.sql, @ddl_setup.sql then @role_setup.sql
-- Supply the name of the Oracle GoldenGate schema for each script
E:\ggs_src>sqlplus /nolog
SQL*Plus: Release 11.2.0.3.0 Production on Fri Jul 27 10:21:38 2012
Copyright (c) 1982, 2011, Oracle.  All rights reserved.

SQL> conn sys/ora11g as sysdba
Connected.
SQL> @marker_setup.sql
...
SQL> @ddl_setup.sql
...
SQL> @role_setup.sql



4. Grant the created role (default name is GGS_GGSUSER_ROLE) to all Oracle
GoldenGate Extract users.
SQL> GRANT GGS_GGSUSER_ROLE TO hr;



5. Run the ddl_enable.sql script to enable the DDL trigger:
@ddl_enable.sql
```

5. Installing Support for Oracle Sequences

Preparing SQL Server 2008 Enterprise Edition for GoldenGate

Reference: Oracle GoldenGate SQL Server Installation and Setup Guide 11g Release 2 Patch Set 1 (11.2.1.0.1) E27293-01

Setting up Basic Replication

About Handling Collisions

- Controlled by the HANDLECOLLISIONS parameter of the Replicate
- It's possible values are: HANDLECOLLISIONS, NOHANDLECOLLISIONS
- HANDLECOLLISIONS does the following:
  - Updates to rows that are missing are ignored.
  - Deletes to rows that are missing are ignored.
  - Inserts that are duplicated are changed to updates.
- NOHANDLECOLLISIONS stops the Replicate and reports an error.
- Typically, during the initial-load, you use HANDLECOLLISIONS. After everything is stable, you use NOHANDLECOLLISIONS.
- You can see if the Replicate encountered any collisions by using the STATS REPLICAT.

Notes on Tables to Replicate in Oracle database

- The table to replicate should not have the NOLOGGING option set.
- If there is no primary key or unique index exists on any table, GG will use all the columns as supplemental logging key pair for both extracts and replicates. Avoid this!
- If we define key columns in the GG extract parameter file and if we don't have the supplemental logging enabled on that key columns combination, then GG will assume missing key columns record data as "NULL".

Replicate Database Connection Options in SQL Server

The following are the ways in which Replicate can connect to a SQL Server **target** database to perform DML operations:

- ODBC or Connect through OLE DB (slightly better performance):
  1. To use ODBC exclusively, include the DBOPTIONS parameter with the USEODBC option in the Replicate parameter file. (To use default OLE DB, no parameter is required.)
  2. Disable triggers and constraints on the target tables.
  3. Configure ODBC connection to SQL Server. See the steps in Configuring an ODBC connection to SQL Server
- OLE DB as the SQL Server replication user. NOT FOR REPLICATION must be set on IDENTITY columns, foreign key constraints, and triggers. Check the documentation "Oracle GoldenGate SQL Server Installation and Setup Guide 11g Release 2" for pros and cons of this option and how to configure it.

Configuring an ODBC connection to SQL Server

Carry out the following steps to create an ODBC data source name (DSN) connection to SQL Server. This ODBC connection is used by the GG Replicate.

1. Run one of the following ODBC clients:

- If using a 32-bit version of gg on a 64-bit system, create the DSN by running the %SystemRoot%\SysWOW64\ODBCAD32.EXE

- If using a 64-bit version of gg on a 64-bit system, create a DSN by running the default ODBCAD32.EXE client in Control Panel>Administrative Tools>Data Sources (ODBC)

- If using a version of gg other than the preceding, use the default ODBC client in Control Panel>Administrative Tools>Data Sources (ODBC)

2. In the ODBC Data Source Administrator dialog box, select the **System DSN**> click **Add**.

3. Select the correct SQL Server driver as follows:

- SQL Server 2005: SQL Native Client driver

- SQL Server 2008: SQL Server Native Client 10.0 driver

4. Click **Finish**. The Create a New Data Source to SQL Server wizard is displayed

5. Supply the following:

- **Name**: choose any. In a Windows cluster, use one name across all nodes in the cluster

- **Server**: Select the SQL Server instance name

6. Click **Next**

7. For login authentication, select the authentication method and supply the login information

8. Click **Next**.

9. If the default database is not set to the one that gg will connect to, click **Change the default database to**, and then select the correct name. Set the other settings to use ANSI.

10. Click **Next**.

11. Leave the next page set to the defaults. Click **Finish**.

13. Click **Test Data Source** to test the connection.

14. Close the box

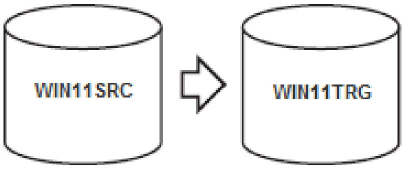
Processes Naming Convention

Here is a proposed naming convection standard for process names. A proposed on is a name that has the following:

- Process Type: local Extract (L or E), initial-load Extract (I), direct-load Replicate (D), Data Pump (P), or Replicate (R)
- The application: such as human resources (HR), payroll (PRL), sales (SL)..etc
- Environment: development (DEV), production (PROD), testing (TST)
- A numeric indicator to show if there are multiple Extract or Replicate groups

Tutorial: Basic Replication between two Oracle Databases

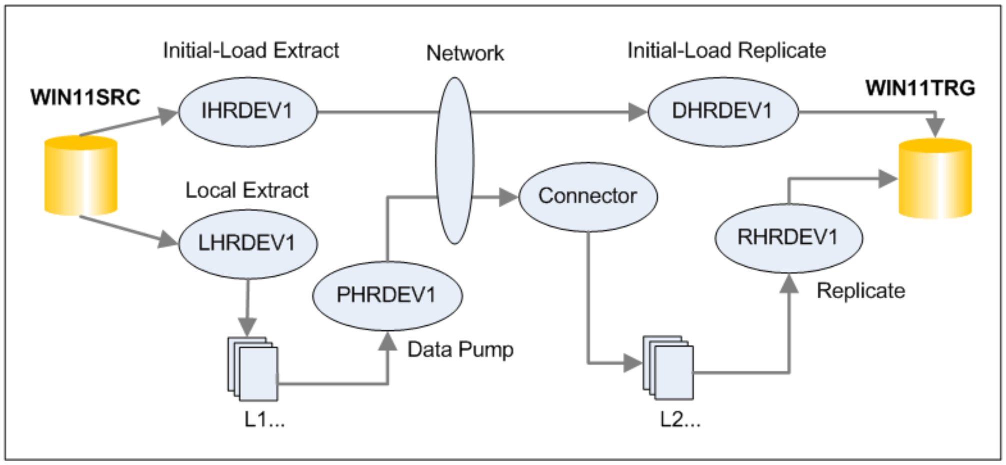
This tutorial shows how to configure GoldenGate to set up a basic one-way replication between two Oracle databases.



Testing environment used for this tutorial was configured as follows:

	Source Environment	Target Environment
Hostname	WIN11SRC	WIN11TRG
Database Software	Oracle Database 11g R2	Oracle Database 11g R2
OS	Windows XP 32-bit	Windows XP 32-bit
Extract Capture Mode	classic	
Replicated Tables	HR.COUNTRIES HR.DEPARTMENTS HR.EMPLOYEES HR.JOB_HISTORY HR.JOBS HR.LOCATIONS HR.REGIONS	HR.COUNTRIES HR.DEPARTMENTS HR.EMPLOYEES HR.JOB_HISTORY HR.JOBS HR.LOCATIONS HR.REGIONS

The processes naming will be as follows:



1. Install Oracle GoldenGate for Oracle Database. Refer to its section in this document.

2. Prepare both the source and target databases for GoldenGate replication. Refer to the section "Preparing Oracle Database for GoldenGate".

3. Review the section "Notes on Tables to Replicate" and check their influence on your case.

4. In the source, carry out the following verifications:

```
-- In the Source, make sure all the tables to replicate have key constraint:
select t.table_name from dba_tables t where t.owner='HR' and
t.table_name in
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','LOCATIONS','REGIO
NS')
minus
select c.table_name from dba_constraints c where c.owner='HR' and
c.constraint_type in ('P','U')



-- the LOGGING option in the table level should be ENABLED
select table_name from dba_tables where owner='HR' and logging='NO' and
table_name in
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','LOCATIONS','REGIO
NS')

-- to enable it:
begin
for r in ( select table_name from dba_tables where owner='HR' and logging='NO'
and table_name in
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','LOCATIONS','REGIO
NS')) loop
 execute immediate 'alter table hr.'|| r.table_name ||' LOGGING';
end loop;
end;

-- log the primary key values:
-- the gg command in the example actually executes in the background:
--  ALTER TABLE <..> ADD SUPPLEMENTAL LOG GROUP <...> (<keyCol>) ALWAYS;
GGSCI (WIN11SRC) 1> dblogin userid ggs_owner password g
GGSCI (WIN11SRC) 2> add trandata hr.employees


-- to generate the script for a group of tables:
set echo off
set verify off
set pagesize 2000
set linesize 250
set trim on
set heading off
set feedback off
spool &&SCHEMA..add_trandata.obey
 select 'add trandata &SCHEMA..'||table_name
 from dba_tables where owner = '&SCHEMA' ;
spool off

-- to run the script in gg:
GGSCI (WIN11SRC) 1> dblogin userid ggs_owner password g
GGSCI (WIN11SRC) 2> obey diroby/HR.add_trandata.obey

-- in gg: to confirm the table logging is enabled:
dblogin userid ggs_owner password g
info trandata hr.EMPLOYEES
```

```
-- in db: to confirm the table logging is enabled:
select owner, log_group_name, table_name
from dba_log_groups where owner = 'HR';

-- to know which column values are being logged:
select * from DBA_LOG_GROUP_COLUMNS where owner = 'HR';

Note: If you use ADD SCHEMATRANDATA command to add the supplement logging, the
information should be obtained from logmnr$always_suplog_columns.
eg: select * from table(logmnr$always_suplog_columns('SCHEMAUSER','T'));


Note: If you find out GG is logging ALL the columns because of the absence of
a key column, adding a key column is not enough to resolve the issue. You need
to do the following as well:
GGSCI> delete trandata hr.EMPLOYEES
GGSCI> add trandata hr.EMPLOYEES
```

5. In Target: disable triggers and cascade-delete constraints. Use either SUPPRESSTRIGGERS option of the Replicate DBOPTIONS parameter or the SQL commands as follows:

```
-- in Target: disable cascade-delete constraints
select 'alter table '||owner||'.'||table_name||
 ' disable constraint '||constraint_name||';'
from all_constraints
where delete_rule = 'CASCADE'
  and owner = 'HR';



-- disable triggers:
select 'alter trigger '||owner||'.'||trigger_name||
' disable ;'
from all_triggers
where owner = 'HR';
```

6. In the source: configure the local Extract.
   It is a good practice to decide on the process name based on a naming convention standard like the one suggested in the section "Processes Naming Convention".

```
1. verify the manager is running
GGSCI (WIN11SRC) 2> info manager
Manager is running (IP port WIN11SRC.7840)


2. create and edit the Extract parameter file (8 characters max):
GGSCI (WIN11SRC) 2>edit params LHRDEV1

-- add the following to it:

Extract LHRDEV1
-------------------------------------------------------------------
-- Local extract for HR schema
-------------------------------------------------------------------
SETENV (NLS_LANG = AMERICAN_AMERICA.WE8MSWIN1252)
-- this is needed, if you have more than one instance:
SETENV (ORACLE_HOME = "E:\oracle\product\11.2.0\dbhome_1")
SETENV (ORACLE_SID = "ORADB")


USERID GGS_OWNER@ORA11SRC, PASSWORD g
-- specify the two-character local Extract trail-file name
ExtTrail dirdat/L1
-- Table parameter must end with semicolon
Table HR.REGIONS;
Table HR.LOCATIONS;
Table HR.JOB_HISTORY;
Table HR.JOBS;
Table HR.EMPLOYEES;
Table HR.DEPARTMENTS;
Table HR.COUNTRIES;



3. add the Extract process
-- create the process and start checkpoint:
ADD EXTRACT LHRDEV1, TRANLOG, BEGIN NOW
-- add local extract trail file:
ADD EXTTRAIL dirdat/l1, EXTRACT LHRDEV1, MEGABYTES 100
-- start the extract (stop command stops it)
START EXTRACT LHRDEV1

/* verify the extract:  */
-- basic info displayed:
info extract LHRDEV1

-- detailed info displayed:
info extract LHRDEV1, DETAIL

-- get stats about the extract operation:
stats ext LHRDEV1


4. create the Data Pump parameter file:
-- no env variables should be defined
edit params PHRDEV1

Extract PHRDEV1
-------------------------------------------------------------------
-- Data Pump extract for HR schema
-------------------------------------------------------------------
-- source and desc identical
PassThru

RmtHost WIN11TRG, MgrPort 7840

-- remote trail file number prefix:
RmtTrail dirdat/l2

Table HR.REGIONS;
Table HR.LOCATIONS;
Table HR.JOB_HISTORY;
Table HR.JOBS;
Table HR.EMPLOYEES;
Table HR.DEPARTMENTS;
Table HR.COUNTRIES;



5. create the Data Pump process:
ADD EXTRACT PHRDEV1, EXTTRAILSOURCE dirdat/l1
ADD RMTTRAIL dirdat/l2, EXTRACT PHRDEV1, MEGABYTES 100
START EXTRACT PHRDEV1

-- verify the process:
info extract PHRDEV1
info extract PHRDEV1, detail
stats ext PHRDEV1
```

7. Configure data initialization (initial-load)

- Prerequisite: You must disable any foreign-key constraints on the target tables.
- To speed up: drop the check constraints and the indexes.

```
/* Option1: Using GoldenGate Initial-load */

-- implement the prerequisite and recommendation above

1. create direct-load Extract parameter file:
ON SOURCE
edit params IHRDEV1

Extract IHRDEV1
-------------------------------------------------------------------
-- Initial Load extract for HR schema
-------------------------------------------------------------------
SETENV (NLS_LANG = AMERICAN_AMERICA.WE8MSWIN1252)

USERID GGS_OWNER@ORA11SRC, PASSWORD g
RmtHost WIN11TRG, mgrport 7840
-- name of the Replicate: DHRDEV1 (D DIRECT LOAD)
RmtTask Replicat, Group DHRDEV1
Table HR.EMPLOYEES;


2. add the Initial-load Extract:
ADD EXTRACT IHRDEV1, SOURCEISTABLE


3. configure and add Initial-load replicate:
-- Replicat is started and stopped automatically by GoldenGate when
--  you start the initial-load Extract
ON THE TARGET host:

edit params DHRDEV1

Replicat DHRDEV1
-------------------------------------------------------------------
-- Initial load replicat for HR schema
-------------------------------------------------------------------
SETENV (NLS_LANG = AMERICAN_AMERICA.WE8MSWIN1252)
```

```
USERID GGS_OWNER@ORA11TRG, PASSWORD g
-- src and trg are identical (no data-definitions file)
AssumeTargetDefs
-- use COLMAP option if the tables do not match
Map HR.EMPLOYEES, Target HR.EMPLOYEES ;


4. add the Replicat group:
ADD REPLICAT DHRDEV1, SPECIALRUN


5. start the Initial-load:
GGSCI (WIN11SRC) > START EXTRACT IHRDEV1

-- to verify the Initial-load (after the initialization is finished):
GGSCI (WIN11TRG) > VIEW REPORT DHRDEV1
```

```
/* Option2: Using third party solution*/
1. Make sure the prerequisites are met.
2. Start the ongoing synchronization Locally and data-pump Extracts to capture
changes.
3. Execute the initial load using the utility you have chosen.
4. Start the ongoing synchronization Replicat to apply changes made during the
initialization.
```

8. In the target system: configure the replicate.

- Prerequisite: You must disable any foreign-key constraints on the target tables.
- To speed up: drop the check constraints and unnecessary indexes.

```
1. create the parameter file:
GGSCI (WIN11TRG)> edit params RHRDEV1

Replicat RHRDEV1
-------------------------------------------------------------------
-- Replicat for HR Schema
-------------------------------------------------------------------
SETENV (NLS_LANG = AMERICAN_AMERICA.WE8MSWIN1252)

USERID GGS_OWNER@ORA11TRG, PASSWORD g

-- typically, this is used some time after the initial-load.
--  Later, NOHANDLECOLLISIONS is used
HANDLECOLLISIONS

-- if the source and target are identical
-- if different, generate a data-definitions file
AssumeTargetDefs

-- semicolon is mandatroy
Map HR.REGIONS, Target HR.REGIONS;
Map HR.LOCATIONS, Target HR.LOCATIONS;
Map HR.JOB_HISTORY, Target HR.JOB_HISTORY;
Map HR.JOBS, Target HR.JOBS;
Map HR.EMPLOYEES, Target HR.EMPLOYEES;
Map HR.DEPARTMENTS, Target HR.DEPARTMENTS;
Map HR.COUNTRIES, Target HR.COUNTRIES;


2. add checkpoint table:
dblogin userid ggs_owner@ora11trg password g
ADD CHECKPOINTTABLE HR.GG_CHKPT


3. add the Replicate
ADD REPLICAT RHRDEV1, EXTTRAIL dirdat/l2, CHECKPOINTTABLE HR.GG_CHKPT
START REPLICAT RHRDEV1


4. verify the Replicate is running
GGSCI (WIN11TRG) > info replicat RHRDEV1
GGSCI (WIN11TRG) > info replicat RHRDEV1, detail
```

9. Test the replication:

```
-- example:
procedure random_update
as
 n number ;
begin
 n := 0;
 for r in 1..100 loop
  n := n + 1;
  update employees set salary = salary * 1 where employee_id =
round(dbms_random.VALUE(100,206));
  if mod(n,10) = 0 then
   commit;
  end if;
 end loop;
 commit;
end;
```

Customizing GoldenGate Replication

Managing the Process Report

- By default, the process report is generated every time it starts up.
- By default, it is located in /dirrpt folder and has rpt extension
- To determine the location of a process report: info <grp> detail
- To view a process report, use any of the following:
  - standard shell command for viewing a text file
  - Oracle GoldenGate Director
  - VIEW REPORT command in GGSCI
- The following parameters influence the report contents:

Parameter	Description	Examples
REPORT	specify when Extract or Replicat generates interim **runtime statistics** in a process report	REPORT AT 1:00 REPORT ON SUNDAY REPORT ON SUNDAY AT 1:00 To send it manually: SEND EXTRACT|REPLICAT xxx REPORT
REPORTCOUNT	To add a **count of the records** GoldenGate has processed since startup. Rate = # of records processed since startup / total time since startup Delta = # of records since last report / time since last report	REPORTCOUNT EVERY 5000 RECORDS -- SECONDS | MINUTES | HOURS REPORTCOUNT EVERY 5 MINUTES, RATE
REPORTROLLOVER	To force report files to age on a regular schedule It doesn't affect the stats data though	REPORTROLLOVER AT 05:00 ON Friday

Reporting Discarded Records

- Discard file is a log of the records that cannot be processed by gg:

Parameter	Description	Examples
DISCARDFILE	Discard filename. Filename can be fully qualified or relative to gg home. If maximum size is defined and reached, the process abends. DISCARDFILE <file name> [, APPEND | PURGE] [, MAXBYTES <n> | MEGABYTES <n>]	DISCARDFILE discard.dsc, PURGE, MEGABYTES 2
DiscardRollover	roll the discard file over	DiscardRollover at 02:00 ON SUNDAY

Purging Old Trail Files

- To automatically purge your old trail files after they have been processed.
- It's best to add it to your Manager parameter file (mgr.prm) so purging can be managed centrally for all Extracts and Replicates.
- When using this parameter, do not permit trail files to be deleted by any user or program other than Oracle GoldenGate. It will cause PURGEOLDEXTRACTS to function improperly.

Parameter	Description	Examples
PURGEOLDEXTRACTS	Deletes unprocessed trail files	PURGEOLDEXTRACTS dirdat/*, USECHECKPOINTS, MINKEEPDAYS 2

Adding Automatic Process Startup and Restart

- This Manager parameter starts one or more Extract and Replicate processes automatically when Manager starts.
- By default with AUTORESTART, GoldenGate tries to restart the Extracts and Replicates two times after waiting two minutes between restarts. Every two hours, GoldenGate tries again. You can adjust these defaults if needed.

Parameter	Description	Examples
AUTOSTART	AUTOSTART <process type> <group name> <process type> = EXTRACT, REPLICAT, ER	AUTOSTART ER *

- You may need to delay AUTOSTART parameters (for example, until database services are started)

Parameter	Description	Examples
BOOTDELAYMINUTES	Supported ONLY in Windows BOOTDELAYMINUTES <minutes>	BOOTDELAYMINUTES 5

Adding a Checkpoint Table

- Add an entry to the GLOBALS file to define the checkpoint file name.
- You can also add a specific checkpoint table for an individual Replicate as part of the ADD REPLICAT command

```
edit params .\GLOBALS
CHECKPOINTTABLE GGS_OWNER.CHKPTAB

dblogin userid GGS_OWNER password userpw
add checkpointtable

info checkpointtable GGS_OWNER.CHKPTAB
```

Securing the Replication

Encrypting Passwords

- You should encrypt the GoldenGate database user passwords in your parameter file.

```
encrypt password abc
USERID 'GGER', PASSWORD "AACAAAAAAAAAAAADAVHTDKHHCSCPIKAFB", ENCRYPTKEY default
```

Encrypting the Trail Files

- The ENCRYPTTRAIL parameter must come before the trail file to be encrypted.

```
-- in source extract:
ENCRYPTTRAIL
ExtTrail dirdat/l1

-- in data pump:
PassThru
DECRYPTTRAIL
ENCRYPTTRAIL
RmtHost targetserver, MgrPort 7809
RmtTrail dirdat/l2

-- in replicate:
DECRYPTTRAIL
Map HR.*, Target HR.* ;
```

Adding Data Filtering and Mapping

Filtering Tables, Columns and Rows

- You may filter in any of the Extract, Replicate, Data Pump or all. It is a good practice though to let the Extract extract all required data and apply the filtering in the data pump process.

```
-- all the tables in the HR schema
HR.*


-- specific tables:
Table HR.EMPLOYEES;
Table HR.JOBS;


-- specific columns (include the Key-column):
-- Note: passthru isn't possible when this option is used
Table HR.EMPLOYEES COLS (employee_Id, first_name, last_name);


-- all except specifi colums:
Table HR.EMPLOYEES COLSEXCEPT (email);


-- filtering rows:
-- WHERE clause to your TABLE statement on the Extract or
-- to your MAP statement on the Replicat
Table HR.EMPLOYEES, WHERE (EMPLOYEE_ID < 100);

-- Double quotes are required for string literals
Map HR.EMPLOYEES, Target HR.EMPLOYEES, WHERE (JOB_ID = "IT_PROG");


-- filtering based on the DML operation
Table HR.EMPLOYEES, FILTER (ON DELETE, SALARY / 12 < 1000);
```

Mapping Different Objects

- This is to handle the situations where the source and target are **not** identical (when ASSUMETARGETDEFS is not applicable). In this case, a data definitions file is required
- If the mapping is on the target server, you generate the data definitions file on the source server and transfer it to the target server.
- If the mapping is on the source using the Extract, you create a data definitions file on the target server and transfer it to the source server.

```
/* this example assumes the mapping is on the target server */

1. on the source:
EDIT PARAMS defgen

defsfile ./dirdef/RHRDEV1.defs
USERID GGS_OWNER@ORA11SRC, PASSWORD g
```

```
TABLE HR.EMPLOYEES;


2. use defgen utility to generate the definition file:
defgen paramfile dirprm/defgen.prm


3. Copy ./dirdef/RHRDEV1.defs to target in dirdef directory.


4. Modify the Replicate param file:
edit params RHRDEV1

...
--AssumeTargetDefs
SourceDefs dirdef/RHRDEV1.defs
...
Map HR.EMPLOYEES, Target HR.STAFF,
COLMAP (USEDEFAULTS,
WAGES = @COMPUTE(SALARY * 12)
FULL_NAME = @STRCAT(LAST_NAME,",",FIRST_NAME));
```

Setting up Bi-directional Replication

Note: Bi-directional replication is sometimes named as "active-active replication".

Bi-directional Replication Restrictions

- You can’t replicate truncates in bidirectional replication. You can work around this restriction by configuring gg to allow truncates only from one database out of the databases configured for the bidirectional replication. See the example tutorial.
- From a practical standpoint, bidirectional replication is typically done with a maximum of two **identical** databases. Thus, ASSUMETARGETDEFS is always used in this scenario.
- In an active-active configuration, it isn’t possible to disable triggers, so you should add code to your triggers to ignore DML generated by the GoldenGate Replicate.
- If you share the same key values across both databases, then you must develop some conflict-resolution routines.
- You may need to plan for database-generated sequential keys so there is no overlap across databases.
- If you’re using CASCADE DELETE constraints, these need to be disabled and converted to triggers.

Excluding Replicate Transactions

- By default, DML transactions issued by Replicate are ignored by an Extract process that is processing the same tables on the same system. This is to prevent endless looping. This behavior is controlled by the GETREPLICATES or IGNOREREPLICATES (default) Extract parameters.
- For most databases, however, you still need to identify to GoldenGate which Replicate user transactions to exclude. You do so with different parameters depending on the DBMS, as shown in the following :
    - For Oracle 10g and later, include the following local Extract parameter file to exclude the gg user:

      TRANLOGOPTIONS EXCLUDEUSER ggs_owner
    - For Oracle 9i, include the following TRACETABLE parameter in both the Extract and Replicate parameter files before any TABLE or MAP statements. When an update occurs, the Replicate updates the trace table. This helps the Extract identify and exclude any GoldenGate transactions. You have to create the trace table first, using the ADD TRACETABLE command. This example uses the trace table ggs_trace:

      TRACETABLE ggs_trace
    - For **DB2 z/OS** and **LUW**, **Ingres**, and **Sybase** code, use the following **local Extract** parameter file to exclude the gg user:

      TRANLOGOPTIONS EXCLUDEUSER gg_owner
    - For **Sybase**, include the following in the **local Extract** parameter file to exclude the default transaction name of ggs_repl. For SQL Server, this parameter is required only if the Replicate transaction name is something other than the default ggs_repl:

      TRANLOGOPTIONS EXCLUDETRANS ggs_repl

- For **Teradata**, you don't need to identify the Replicate transaction; however, you must include the following SQLEXEC statements in the **Replicate** parameter file to execute a procedure that sets the Replicate session automatically at startup:

  SQLEXEC "SET SESSION OVERRIDE REPLICATION ON;"

  SQLEXEC "COMMIT;"

- For **SQL/MX**, include the following in the **local Extract** parameter file to exclude the operations on the checkpoint table gger.ckpt:

  TRANLOGOPTIONS FILTERTABLE gger.chkpt

Handling conflicts using CDR feature

- Conflict Detection and Resolution (CDR) feature in GoldenGate requires the database to reside on a Windows, Linux, or UNIX platform.
- The columns to compare should be of types: NUMERIC, DATE, TIMESTAMP, CHAR/NCHAR, VARCHAR/NVARCHAR

Note: GG started supporting bulk data loads with their 11.1 release but any NOLOGGING data changes will be silently ignored without any warning!

1. Enable the logging of before images of the columns to be used by GG for update and delete conflict detection. Include the columns that will be used by gg to determine if a conflict occurs.

```
ADD TRANDATA HR.EMPLOYEES, COLS (LAST_NAME, EMAIL, JOB_ID, FIRST_NAME, SALARY,
COMMISSION_PCT, MANAGER_ID, DEPARTMENT_ID, DML_TIMESTAMP);

-- if it is already partially logged:
DELETE TRANDATA HR.EMPLOYEES
ADD TRANDATA HR.EMPLOYEES, COLS (LAST_NAME, EMAIL, JOB_ID, FIRST_NAME, SALARY,
COMMISSION_PCT, MANAGER_ID, DEPARTMENT_ID, DML_TIMESTAMP);
INFO trandata hr.employees

-- to confirm in the db side:
SELECT LOG_GROUP_NAME, TABLE_NAME,
 DECODE(ALWAYS, 'ALWAYS', 'Unconditional', 'CONDITIONAL', 'Conditional')
ALWAYS,
LOG_GROUP_TYPE
FROM DBA_LOG_GROUPS;


-- If the type of log group is USER LOG GROUP, then
--  you can list the columns in the log group by:
select OWNER, LOG_GROUP_NAME, TABLE_NAME, COLUMN_NAME, POSITION,
LOGGING_PROPERTY
from DBA_LOG_GROUP_COLUMNS
where owner='HR'
```

2. Configure the Oracle GoldenGate parameter files for conflict resolution using the parameters: GETBEFORECOLS (Extract), COMPARECOLS (Replicate), and RESOLVECONFLICT (Replicate)

```
-- semicolon is mandatroy
Map HR.EMPLOYEES, Target HR.EMPLOYEES,
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))),
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))),
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)),
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)),
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));


/* Example 2: to define which columns are used when comparing the
    after and before images */
MAP fin.src, TARGET fin.tgt,
COMPARECOLS
(ON UPDATE KEYINCLUDING (address, phone, salary, last_mod_time),
ON DELETE KEYINCLUDING (address, phone, salary, last_mod_time)),
...

-- you can use ALL except:
MAP fin.src, TARGET fin.tgt,
COMPARECOLS
(ON UPDATE ALLEXCLUDING (comment)),...



/* Example 3: use you won routine to handle the conflicts */
MAP HR.EMPLOYEES, TARGET HR.EMPLOYEES, &
REPERROR (90000, DISCARD), &
SQLEXEC (ID checkemployee, ON UPDATE, &
QUERY "select count(*) empduplicate from HR.EMPLOYEES where employee_id = ?
and &
employee_timestamp > ?", &
PARAMS (p1 = employee_id, p2 = employee_timestamp), BEFOREFILTER, ERROR
REPORT, &
TRACE ALL),&
FILTER (checkemployee.empduplicate = 0, ON UPDATE, RAISEERROR 90000);
```

3. Configure the Oracle GoldenGate parameter files for handling errors. This is shown in the following section.

4. To obtain CDR statistics:

   - View the generated report file
   - The ggsci command: STATS REPLICAT <group>, REPORTCDR. If no conflict was detected, you won't see any CDR stats.
   - Use Column-conversion functions as follows. They can be inserted in an exceptions table or used in other Oracle GoldenGate parameters that accept input from column-conversion functions

```
-- conflict stats for a specific table or set of wildcarded tables:
-- toal conflicts
@GETENV("STATS","TABLE","SCHEMA.TABLNAME","CDR_CONFLICTS")
-- resolved
@GETENV("STATS","TABLE","SCHEMA.TABLNAME","CDR_RESOLUTIONS_SUCCEEDED")
-- not resolved
@GETENV("STATS","TABLE","SCHEMA.TABLNAME","CDR_RESOLUTIONS_FAILED")


-- conflict stats for all of the tables in all of the MAP statements
--  in the Replicat parameter file:
@GETENV("STATS","CDR_CONFLICTS")
@GETENV("STATS","CDR_RESOLUTIONS_SUCCEEDED")
@GETENV("STATS","CDR_RESOLUTIONS_FAILED")
```

```
-- replace "STATS" with "DELTASTATS" to return the counts since
--  the last execution of "DELTASTATS".
```

Handling Errors in CDR

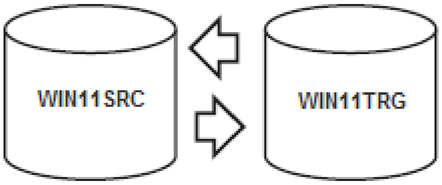
- Use the parameter REPERROR which controls how Replicate responds to errors.
- Optionally, you can create an exceptions table to hold all the information about exceptions.

```
/* This example maps source and target columns, but no extra columns. */
The source and target exceptions columns are identical in name and definition.
There are no other columns in the exceptions table.


-- REPERROR error handling: DEFAULT represents all error types. DISCARD
-- writes operations that could not be processed to a discard file.
REPERROR (DEFAULT, DISCARD)
-- Specifies the discard file.
DISCARDFILE ./dirrpt/rhrdev1_discards.dsc, PURGE

-- The regular MAP statement with the CDR parameters
MAP fin.src, TARGET fin.tgt, &
COMPARECOLS (ON UPDATE ALL, ON DELETE ALL), &
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX (last_mod_time)), &
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX (last_mod_time)), &
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
-- the update will be converted into INSERT but the before image
-- data should exist in the logging
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD)), &
);

-- Optionally, direct replicate to insert unhandled rows in
--  an exception table
MAP fin.src, TARGET fin.exception, &
-- map operations that caused the error specified in REPERROR
EXCEPTIONSONLY, &
-- the exceptions table should not have any pk or unq constraints
INSERTALLRECORDS &
;


/* Example 2: exceptions mapping with additional columns
              in the exceptions table */
-- rows from source and target
-- further errors details are also caught
-- everything is the same as above except the MAP which
-- should be replaced by:

MAP fin.src, TARGET fin.exception, &
-- only map operations that caused the error
EXCEPTIONSONLY, &
-- directs Replicat to convert all the exceptions to inserts into the
-- exceptions table
INSERTALLRECORDS &
```

```
-- SQLEXEC query to select the values from the target record before the
-- Replicat statement is applied. These are mapped to the “*_target”
-- columns later.
SQLEXEC (id qry, query "select name, phone, address, salary, balance, &
comment, last_mod_time from fin.tgt where name = :p1", PARAMS(p1 = name )), &
-- Start of the column mapping, specifies use default column definitions
COLMAP ( &
-- USEDEFAULTS maps the source columns to the target exceptions columns
-- that receive the "after" image that Replicat applied or tried to apply.
-- In this case, USEDEFAULTS can be used because the names and definitions
-- of the source and target exceptions columns are identical; otherwise
-- the columns must be mapped explicitly in the COLMAP clause.
USEDEFAULTS, &
-- captures the timestamp when the resolution was performed.
res_date = @DATENOW(), &
-- captures and maps the DML operation type.
optype = @GETENV("LASTERR", "OPTYPE"), &
-- captures and maps the database error number that was returned.
dberrnum = @GETENV("LASTERR", "DBERRNUM"), &
-- captures and maps the database error that was returned.
dberrmsge = @GETENV("LASTERR", "DBERRMSG"), &
-- captures and maps the name of the target table
tabname = @GETENV("GGHEADER", "TABLENAME"), &
-- If the names and definitions of the source columns and the target
-- exceptions columns were not identical, the columns would need to
-- be mapped in the COLMAP clause instead of using USEDEFAULTS:
-- name_after = name, &
-- phone_after = phone, &
-- address_after = address, &
-- salary_after = salary, &
-- balance_after = balance, &
-- comment_after = comment, &
-- last_mod_time_after = last_mod_time &
-- maps the before image of each column from the trail to a column in the
-- exceptions table.
name_before = before.name, &
phone_before = before.phone, &
address_before = before.address, &
salary_before = before.salary, &
balance_before = before.balance, &
comment_before = before.comment, &
last_mod_time_before = before.last_mod_time, &
-- maps the results of the SQLEXEC query to rows in the exceptions table
-- to show the current image of the row in the target.
name_current = qry.name, &
phone_current = qry.phone, &
address_current = qry.address, &
salary_current = qry.salary, &
balance_current = qry.balance, &
comment_current = qry.comment, &
last_mod_time_current = qry.last_mod_time) &
;
```

Tutorial: Setting Up Bi-directional Replication between Oracle Databases

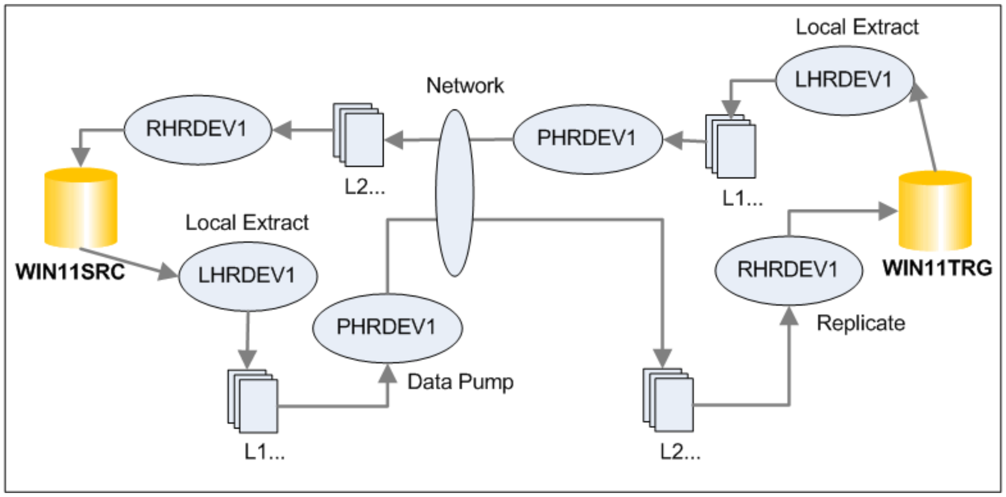
This tutorial shows how to configure GoldenGate to set up a bi-directional replication between two Oracle databases. This configuration will replicate the tables in HR schema and handle conflicts that might occur.



Testing environment used for this tutorial was configured as follows:

	Environment 1	Environment 2
Hostname	WIN11SRC	WIN11TRG
Database Software	Oracle Database 11g R2	Oracle Database 11g R2
OS	Windows XP 32-bit	Windows XP 32-bit
Extract Capture Mode	classic	
Replicated Tables	HR.COUNTRIES HR.DEPARTMENTS HR.EMPLOYEES HR.JOB_HISTORY HR.JOBS HR.LOCATIONS HR.REGIONS	HR.COUNTRIES HR.DEPARTMENTS HR.EMPLOYEES HR.JOB_HISTORY HR.JOBS HR.LOCATIONS HR.REGIONS
Allow Truncate	Yes	No
Database Role	Primary (initialization source)	Secondary (target)
Conflict Resolution	based on DML_TIMESTAMP column	based on DML_TIMESTAMP column

The processes are configured as follows (connectors omitted for simplicity):



1. Install Oracle GoldenGate. Refer to its section in this document.
2. Prepare both the source and target databases for GoldenGate replication. Refer to the section "Preparing Oracle Database for GoldenGate".
3. Review the section "Notes on Tables to Replicate" and check its influence on your case.
4. Carry out the following:

```
/* in the BOTH databases: */

-- Make sure all the tables to replicate have key constraint
select t.table_name from dba_tables t where t.owner='HR' and
t.table_name in
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','LOCATIONS','REGIO
NS')
minus
select c.table_name from dba_constraints c where c.owner='HR' and
c.constraint_type in ('P','U')


-- the LOGGING option in the table level should be ENABLED
select table_name from dba_tables where owner='HR' and logging='NO'
and table_name in
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','LOCATIONS','REGIO
NS')

-- to fix those with NOLOGGING:
begin
for r in ( select table_name from dba_tables where owner='HR' and logging='NO'
   and table_name in
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','LOCATIONS','REGIO
NS')) loop
 execute immediate 'alter table hr.'|| r.table_name ||' LOGGING';
end loop;
end;



-- to resolve the conflicts, DML_TIME column is added to the tables:
-- to generate the code:
select 'ALTER TABLE HR.' || t.table_name || ' ADD (DML_TIMESTAMP TIMESTAMP
DEFAULT SYSTIMESTAMP);' SCRIPT_COMMAND
from dba_tables t where t.owner='HR' and
t.table_name in
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','LOCATIONS','REGIO
NS')


select
'CREATE OR REPLACE TRIGGER HR.SET_'|| substr(table_name,1,6) || '_DMLTSTMP' ||
chr(10) ||
'BEFORE UPDATE
ON HR.'|| table_name ||'
REFERENCING NEW AS New OLD AS Old
FOR EACH ROW
BEGIN
 IF SYS_CONTEXT (''USERENV'', ''SESSION_USER'') != ''GGS_OWNER'' THEN
   :NEW.DML_TIMESTAMP := SYSTIMESTAMP;
 END IF;
```

```
END;
/' as SCRIPT_COMMAND
from dba_tables t where t.owner='HR' and
t.table_name in
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','LOCATIONS','REGIO
NS')


-- enable table level supplemental logging
Note: If you are using the Oracle GoldenGate DDL replication feature, use the
ADD SCHEMATRANDATA command to log the required supplemental data instead of
ADD TRANDATA command. You can use ADD TRANDATA even when using ADD
SCHEMATRANDATA if you need to use the COLS option to log any non-key columns,
such as those needed for FILTER statements and KEYCOLS clauses in the TABLE
and MAP parameters.

-- to generate its code:
select 'add trandata HR.'||t.table_name  || ' COLS(DML_TIMESTAMP)' AS
SCRIPT_COMMAND
from dba_tables t where t.owner='HR' and
t.table_name in
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','LOCATIONS','REGIO
NS')

-- save the output in a file, name it something like: hr_add_trandata.obey
-- to run the script in gg:
GGSCI > dblogin userid ggs_owner password g
GGSCI > obey diroby\hr_add_trandata.obey


-- in gg: to confirm the table logging is enabled:
dblogin userid ggs_owner password g
info trandata hr.*
```

5. Handle truncates: truncate statement against the replicated tables should be executed only from one of the databases in the replication. The other database should not allow truncating any of those tables.

   The following system trigger will disallow any user in WIN11TRG from truncating any of the replicated tables.

```
/* in WIN11TRG  */
CREATE OR REPLACE TRIGGER no_truncate_trg
before TRUNCATE on database
begin
 if ora_dict_obj_owner ||'.'|| ora_dict_obj_name in
('HR.COUNTRIES','HR.DEPARTMENTS','HR.EMPLOYEES','HR.JOB_HISTORY','HR.JOBS','HR
.LOCATIONS','HR.REGIONS') and ora_login_user <>'GGS_OWNER' then
  raise_application_error(-20001,'You should not truncate this table from this
database. Use WINSRC11 instead.');
 else
  null;
 end if;
end;
/
```

6. In both databases, depending on the code in the triggers, you may need to modify the triggers to ignore DML operations that are applied by Replicate to avoid looping.

Alternatively, for Oracle versions 10.2.0.5, 11.2.0.2 and later (11gR1 not included), you can use the DBOPTIONS parameter with the SUPPRESSTRIGGERS option to disable the triggers for the Replicate session. Refer to GoldenGate Reference documentation for details.

```
-- the following trigger prevents changing data on specific timings in the day
--  this doesn't apply to gg sessions. The code thus changed to the following:

CREATE OR REPLACE TRIGGER secure_employees
  BEFORE INSERT OR UPDATE OR DELETE ON employees
DISABLE
BEGIN
 IF SYS_CONTEXT ('USERENV', 'CURRENT_USER') != 'GGS_OWNER' THEN
  secure_dml;
 END IF;
END secure_employees;
/
```

7. Handle ON DELETE CASCADE constraints on both databases, if they exist.

   Do as what stated in Oracle documentation: "Disable ON DELETE CASCADE constraints and use a trigger on the parent table to perform the required delete(s) to the child tables. Create it as a BEFORE trigger so that the child tables are deleted before the delete operation is performed on the parent table. This reverses the logical order of a cascaded delete but is necessary so that the operations are replicated in the correct order to prevent "table not found" errors on the target."

```
select * from dba_constraints where owner='HR' and DELETE_RULE ='DELETE_RULE'
```

8. Database-generated values: apply a mechanism to avoid repeated Id in the replicated databases. You can set the increment value to 2 for each database with different starting value. You also can add a location identifier to the value to enforce uniqueness.

9. On both systems: create a Replicate checkpoint table.

```
dblogin userid ggs_owner@ora11src password g
ADD CHECKPOINTTABLE HR.GG_CHKPT

dblogin userid ggs_owner@ora11trg password g
ADD CHECKPOINTTABLE HR.GG_CHKPT
```

10. On both systems, configure the Manager process:

```
edit params mgr

port 7840
-- autostart Extract and Replicate when Manager starts up
autostart ER *

-- delete old trail files (optional)
PURGEOLDEXTRACTS dirdat/*, USECHECKPOINTS, MINKEEPDAYS 3
```

11. In Source: configure the local Extract and the Data Pump.

```
1. verify the manager is running
GGSCI (WIN11SRC) 2> info manager
Manager is running (IP port WIN11SRC.7840)
```

```
2. create and edit the Extract parameter file (8 characters max):
GGSCI (WIN11SRC) 2>edit params LHRDEV1

-- add the following to it:

Extract LHRDEV1
-------------------------------------------------------------------
-- Local extract for HR schema
-------------------------------------------------------------------
SETENV (NLS_LANG = AMERICAN_AMERICA.WE8MSWIN1252)
-- this is needed, if you have more than one instance:
SETENV (ORACLE_HOME = "E:\oracle\product\11.2.0\dbhome_1")
SETENV (ORACLE_SID = "ORADB")

-- handling process report parameters(optional)
REPORT AT 8:00
REPORTCOUNT EVERY 60 MINUTES, RATE
REPORTROLLOVER AT 08:00 ON MONDAY

-- discard file PURGE or APPEND (optional)
DISCARDFILE discard.dsc, PURGE, MEGABYTES 200
DiscardRollover at 02:00 ON SUNDAY

-- "truncate table" is only allowed from Source:
GETTRUNCATES

-- Exclude Replicate trans (Oracle)
TRANLOGOPTIONS EXCLUDEUSER ggs_owner COMPLETEARCHIVEDLOGTIMEOUT 600

-- ignore local replicate trans (default anyway):
IGNOREREPLICATES

-- get application trans (default anyway):
GETAPPLOPS

-- connect to the source database:
USERID GGS_OWNER@ORA11SRC, PASSWORD g

-- specify the two-character local Extract trail-file name
ExtTrail dirdat/L1

-- Table parameter must end with semicolon
-- Capture before images for conflict resolution:
TABLE HR.COUNTRIES, GETBEFORECOLS (ON UPDATE KEYINCLUDING (DML_TIMESTAMP), ON
DELETE KEYINCLUDING (DML_TIMESTAMP));
TABLE HR.DEPARTMENTS, GETBEFORECOLS (ON UPDATE KEYINCLUDING (DML_TIMESTAMP),
ON DELETE KEYINCLUDING (DML_TIMESTAMP));
TABLE HR.EMPLOYEES, GETBEFORECOLS (ON UPDATE KEYINCLUDING (DML_TIMESTAMP), ON
DELETE KEYINCLUDING (DML_TIMESTAMP));
TABLE HR.JOB_HISTORY, GETBEFORECOLS (ON UPDATE KEYINCLUDING (DML_TIMESTAMP),
ON DELETE KEYINCLUDING (DML_TIMESTAMP));
TABLE HR.JOBS, GETBEFORECOLS (ON UPDATE KEYINCLUDING (DML_TIMESTAMP), ON
DELETE KEYINCLUDING (DML_TIMESTAMP));
TABLE HR.LOCATIONS, GETBEFORECOLS (ON UPDATE KEYINCLUDING (DML_TIMESTAMP), ON
DELETE KEYINCLUDING (DML_TIMESTAMP));
TABLE HR.REGIONS, GETBEFORECOLS (ON UPDATE KEYINCLUDING (DML_TIMESTAMP), ON
DELETE KEYINCLUDING (DML_TIMESTAMP));
```

```
3. add a primary Extract group:
-- create the process and start checkpoint:
ADD EXTRACT LHRDEV1, TRANLOG, BEGIN NOW

-- add local extract trail file:
ADD EXTTRAIL dirdat/l1, EXTRACT LHRDEV1, MEGABYTES 100

-- start the extract (stop command stops it)
START EXTRACT LHRDEV1

/* verify the extract:  */
-- basic info displayed:
info extract LHRDEV1

-- detailed info displayed:
info extract LHRDEV1, DETAIL

-- get stats about the extract operation:
stats ext LHRDEV1



4. create the Data Pump parameter file:
-- no env variables should be defined
edit params PHRDEV1

Extract PHRDEV1
-------------------------------------------------------------------
-- Data Pump extract for HR schema
-------------------------------------------------------------------
-- source and desc identical
PassThru

RmtHost WIN11TRG, MgrPort 7840

-- remote trail file number prefix:
RmtTrail dirdat/l2

-- "truncate table" is only allowed from Source:
GETTRUNCATES

Table HR.* ;



5. create the Data Pump process:
ADD EXTRACT PHRDEV1, EXTTRAILSOURCE dirdat/l1
ADD RMTTRAIL dirdat/l2, EXTRACT PHRDEV1, MEGABYTES 100
START EXTRACT phrdev1

-- verify the process:
info extract PHRDEV1
info extract PHRDEV1, detail
stats ext PHRDEV1
```

12. On the Target: create the Replicate group.

```
1. create the parameter file:
GGSCI (WIN11TRG) 1> edit params RHRDEV1

Replicat RHRDEV1
-------------------------------------------------------------------
-- Replicat for HR Schema
-------------------------------------------------------------------
SETENV (NLS_LANG = AMERICAN_AMERICA.WE8MSWIN1252)

-- Specifies the discard file.
DISCARDFILE ./dirrpt/rhrdev1_discards.dsc, APPEND, MEGABYTES 100
DiscardRollover at 02:00 ON SUNDAY

-- connect to the target db
USERID GGS_OWNER@ORA11TRG, PASSWORD g

-- the source and target are identical
ASSUMETARGETDEFS

-- truncates is only allowed from source
-- IGNORETRUNCATES
GETTRUNCATES


-- Specify error handling rules:
-- writes operations that could not be processed to a discard file
REPERROR (DEFAULT, DISCARD)

-- semicolon is mandatory
-- KEYINCLUDING must match the one in Extract
MAP HR.COUNTRIES, Target HR.COUNTRIES, COMPARECOLS (ON UPDATE KEYINCLUDING
(DML_TIMESTAMP), ON DELETE KEYINCLUDING (DML_TIMESTAMP)), &
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
MAP HR.DEPARTMENTS, Target HR.DEPARTMENTS, COMPARECOLS (ON UPDATE KEYINCLUDING
(DML_TIMESTAMP), ON DELETE KEYINCLUDING (DML_TIMESTAMP)), &
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
MAP HR.EMPLOYEES, Target HR.EMPLOYEES, COMPARECOLS (ON UPDATE KEYINCLUDING
(DML_TIMESTAMP), ON DELETE KEYINCLUDING (DML_TIMESTAMP)), &
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
MAP HR.JOBS, Target HR.JOBS, COMPARECOLS (ON UPDATE KEYINCLUDING
(DML_TIMESTAMP), ON DELETE KEYINCLUDING (DML_TIMESTAMP)), &
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
MAP HR.JOB_HISTORY, Target HR.JOB_HISTORY, COMPARECOLS (ON UPDATE KEYINCLUDING
(DML_TIMESTAMP), ON DELETE KEYINCLUDING (DML_TIMESTAMP)), &
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
MAP HR.LOCATIONS, Target HR.LOCATIONS, COMPARECOLS (ON UPDATE KEYINCLUDING
(DML_TIMESTAMP), ON DELETE KEYINCLUDING (DML_TIMESTAMP)), &
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
MAP HR.REGIONS, Target HR.REGIONS, COMPARECOLS (ON UPDATE KEYINCLUDING
(DML_TIMESTAMP), ON DELETE KEYINCLUDING (DML_TIMESTAMP)), &
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));

-- Optionally, specify mapping of exceptions to exceptions table:
-- MAP <owner>.*, TARGET <owner>.<exceptions>, EXCEPTIONSONLY;
```

Note: the MAP settings can be generated by:

```
select 'MAP HR.'||t.table_name ||', Target HR.'|| t.table_name || ',
COMPARECOLS (ON UPDATE KEYINCLUDING (DML_TIMESTAMP), ON DELETE KEYINCLUDING
(DML_TIMESTAMP)), &
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));'
from dba_tables t where t.owner='HR' and
t.table_name in
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','LOCATIONS','REGIO
NS')
order by t.table_name
```

```
2. add the Replicate
ADD REPLICAT RHRDEV1, EXTTRAIL dirdat/l2, CHECKPOINTTABLE HR.GG_CHKPT
START REPLICAT rhrdev1
```

```
3. verify the Replicate is running
INFO REPLICAT rhrdev1
INFO REPLICAT rhrdev1, DETAIL
```

Note: The following is almost a reverse image of the configuration made in the Primary database.

13. In Secondary: configure the local Extract and the Data Pump.

```
1. create and edit the Extract parameter file (8 characters max):
GGSCI (WIN11TRG)> edit params LHRDEV1

-- add the following to it:

Extract LHRDEV1
-------------------------------------------------------------------
-- Local extract for HR schema
-------------------------------------------------------------------
SETENV (NLS_LANG = AMERICAN_AMERICA.WE8MSWIN1252)
-- this is needed, if you have more than one instance:
SETENV (ORACLE_HOME = "E:\oracle\product\11.2.0\dbhome_1")
SETENV (ORACLE_SID = "ORADB")

-- handling process report parameters(optional)
REPORT AT 8:00
REPORTCOUNT EVERY 60 MINUTES, RATE
REPORTROLLOVER AT 08:00 ON MONDAY

-- discard file PURGE or APPEND (optional)
DISCARDFILE discard.dsc, PURGE, MEGABYTES 200
DiscardRollover at 02:00 ON SUNDAY

-- "truncate table" is only allowed from Source:
-- GETTRUNCATES
IGNORETRUNCATES

-- Exclude Replicate trans (Oracle)
TRANLOGOPTIONS EXCLUDEUSER ggs_owner COMPLETEARCHIVEDLOGTIMEOUT 600

-- ignore local replicate trans (default anyway):
IGNOREREPLICATES

-- get application trans (default anyway):
GETAPPLOPS

-- connect to the target database:
USERID GGS_OWNER@ORA11TRG, PASSWORD g

-- specify the two-character local Extract trail-file name
ExtTrail dirdat/L1

-- Table parameter must end with semicolon
-- Capture before images for conflict resolution:
TABLE HR.COUNTRIES, GETBEFORECOLS (ON UPDATE KEYINCLUDING (DML_TIMESTAMP), ON
DELETE KEYINCLUDING (DML_TIMESTAMP));
TABLE HR.DEPARTMENTS, GETBEFORECOLS (ON UPDATE KEYINCLUDING (DML_TIMESTAMP),
ON DELETE KEYINCLUDING (DML_TIMESTAMP));
TABLE HR.EMPLOYEES, GETBEFORECOLS (ON UPDATE KEYINCLUDING (DML_TIMESTAMP), ON
DELETE KEYINCLUDING (DML_TIMESTAMP));
TABLE HR.JOB_HISTORY, GETBEFORECOLS (ON UPDATE KEYINCLUDING (DML_TIMESTAMP),
ON DELETE KEYINCLUDING (DML_TIMESTAMP));
TABLE HR.JOBS, GETBEFORECOLS (ON UPDATE KEYINCLUDING (DML_TIMESTAMP), ON
DELETE KEYINCLUDING (DML_TIMESTAMP));
TABLE HR.LOCATIONS, GETBEFORECOLS (ON UPDATE KEYINCLUDING (DML_TIMESTAMP), ON
DELETE KEYINCLUDING (DML_TIMESTAMP));
```

```
TABLE HR.REGIONS, GETBEFORECOLS (ON UPDATE KEYINCLUDING (DML_TIMESTAMP), ON
DELETE KEYINCLUDING (DML_TIMESTAMP));



3. add a primary Extract group:
ADD EXTRACT LHRDEV1, TRANLOG, BEGIN NOW
ADD EXTTRAIL dirdat/l1, EXTRACT LHRDEV1, MEGABYTES 100
START EXTRACT LHRDEV1

/* verify the extract:  */
info extract LHRDEV1
info extract LHRDEV1, DETAIL
stats ext LHRDEV1



4. create the Data Pump parameter file:
-- no env variables should be defined
edit params PHRDEV1

Extract PHRDEV1
-------------------------------------------------------------------
-- Data Pump extract for HR schema
-------------------------------------------------------------------
-- source and desc identical
PassThru

RmtHost WIN11SRC, MgrPort 7840

-- remote trail file number prefix:
RmtTrail dirdat/l2

-- "truncate table" is only allowed from Source:
IGNORETRUNCATES

Table HR.* ;



5. create the Data Pump process:
ADD EXTRACT PHRDEV1, EXTTRAILSOURCE dirdat/l1
ADD RMTTRAIL dirdat/l2, EXTRACT PHRDEV1, MEGABYTES 100
START EXTRACT phrdev1

-- verify the process:
info extract PHRDEV1
info extract PHRDEV1, detail
stats ext PHRDEV1
```

14. On the Primary: create the Replicate group.

```
1. create the parameter file :
Note: it can be copy of the one in the Secondary but the USERID must be
modified to connect to the source
GGSCI (WIN11SRC) > edit params RHRDEV1

Replicat RHRDEV1
```

```
-------------------------------------------------------------------
-- Replicat for HR Schema
-------------------------------------------------------------------
SETENV (NLS_LANG = AMERICAN_AMERICA.WE8MSWIN1252)

-- Specifies the discard file.
DISCARDFILE ./dirrpt/rhrdev1_discards.dsc, APPEND, MEGABYTES 100
DISCARDROLLOVER at 02:00 ON SUNDAY

-- connect to the source db
USERID GGS_OWNER@ORA11SRC, PASSWORD g

-- the source and target are identical
ASSUMETARGETDEFS

-- truncates is only allowed from source
-- IGNORETRUNCATES
IGNORETRUNCATES

-- Specify error handling rules:
-- writes operations that could not be processed to a discard file
REPERROR (DEFAULT, DISCARD)

-- semicolon is mandatory
-- KEYINCLUDING must match the one in Extract
MAP HR.COUNTRIES, Target HR.COUNTRIES, COMPARECOLS (ON UPDATE KEYINCLUDING
(DML_TIMESTAMP), ON DELETE KEYINCLUDING (DML_TIMESTAMP)), &
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
MAP HR.DEPARTMENTS, Target HR.DEPARTMENTS, COMPARECOLS (ON UPDATE KEYINCLUDING
(DML_TIMESTAMP), ON DELETE KEYINCLUDING (DML_TIMESTAMP)), &
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
MAP HR.EMPLOYEES, Target HR.EMPLOYEES, COMPARECOLS (ON UPDATE KEYINCLUDING
(DML_TIMESTAMP), ON DELETE KEYINCLUDING (DML_TIMESTAMP)), &
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
MAP HR.JOBS, Target HR.JOBS, COMPARECOLS (ON UPDATE KEYINCLUDING
(DML_TIMESTAMP), ON DELETE KEYINCLUDING (DML_TIMESTAMP)), &
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
MAP HR.JOB_HISTORY, Target HR.JOB_HISTORY, COMPARECOLS (ON UPDATE KEYINCLUDING
(DML_TIMESTAMP), ON DELETE KEYINCLUDING (DML_TIMESTAMP)), &
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
MAP HR.LOCATIONS, Target HR.LOCATIONS, COMPARECOLS (ON UPDATE KEYINCLUDING
(DML_TIMESTAMP), ON DELETE KEYINCLUDING (DML_TIMESTAMP)), &
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));
MAP HR.REGIONS, Target HR.REGIONS, COMPARECOLS (ON UPDATE KEYINCLUDING
(DML_TIMESTAMP), ON DELETE KEYINCLUDING (DML_TIMESTAMP)), &
RESOLVECONFLICT (UPDATEROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (INSERTROWEXISTS, (DEFAULT, USEMAX (DML_TIMESTAMP))), &
RESOLVECONFLICT (DELETEROWEXISTS, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (UPDATEROWMISSING, (DEFAULT, OVERWRITE)), &
RESOLVECONFLICT (DELETEROWMISSING, (DEFAULT, DISCARD));

-- Optionally, specify mapping of exceptions to exceptions table:
-- MAP <owner>.*, TARGET <owner>.<exceptions>, EXCEPTIONSONLY;



2. add the Replicate
ADD REPLICAT RHRDEV1, EXTTRAIL dirdat/l2, CHECKPOINTTABLE HR.GG_CHKPT
START REPLICAT rhrdev1



3. verify the Replicate is running
INFO REPLICAT rhrdev1
INFO REPLICAT rhrdev1, DETAIL

4. test the replication
```

Heterogeneous Replication

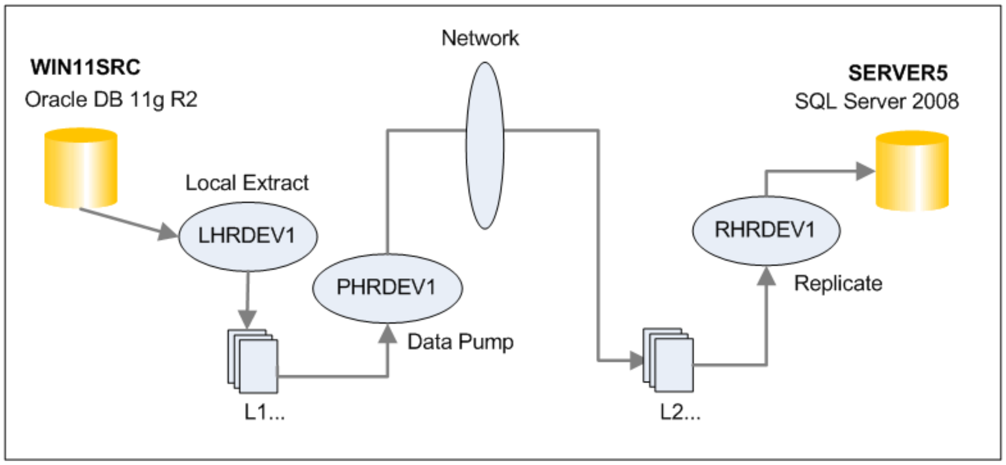
Tutorial: Oracle Database to SQL Server

This tutorial shows how to configure GoldenGate to set up a basic one-way replication from an Oracle database to SQL Server database.

Testing environment used for this tutorial was configured as follows:

	Source Environment	Target Environment
Hostname	WIN11SRC	SERVER5
Database Software	Oracle Database 11g R2	SQL Server 2008 R2
OS	Windows XP 32-bit	Windows 2003 SE 32-bit
Extract Capture Mode	classic	

The processes naming will be as follows:



1. In both systems: install Oracle GoldenGate. Refer to its section in this document. Remember, we need to use two different GG software binaries: one for Oracle and another one for SQL Server.

2. Prepare the source and target databases for GoldenGate replication. Refer to the section "Preparing Oracle Database for GoldenGate".

3. Review the section "Notes on Tables to Replicate" and check its influence on your case.

4. Prepare the tables in the source database by carrying out the following:
   Note: MOC 1315720.1 has a script which can be used to verify that a SQL Server is ready to be used by gg.

```
-- in the Source database:
-- Note: make sure all the tables to replicate have key constraint
select t.table_name from dba_tables t where t.owner='HR' and
t.table_name in
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','LOCATIONS','REGIO
NS')
```

```
minus
select c.table_name from dba_constraints c where c.owner='HR' and
c.constraint_type in ('P','U')

-- the LOGGING option in the table level should be ENABLED
select table_name from dba_tables where owner='HR' and logging='NO'

-- to enable it/them:
begin
for r in ( select table_name from dba_tables where owner='HR' and logging='NO'
and table_name in
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','LOCATIONS','REGIO
NS')) loop
 execute immediate 'alter table hr.'|| r.table_name ||' LOGGING';
end loop;
end;

-- log the primary key values:
-- this executes in the background:
--  ALTER TABLE <..> ADD SUPPLEMENTAL LOG GROUP <...> (<keyCol>) ALWAYS;
GGSCI (WIN11SRC) 1> dblogin userid ggs_owner password g
GGSCI (WIN11SRC) 2> add trandata hr.employees


-- to generate the script for a group of tables:
set echo off
set verify off
set pagesize 2000
set linesize 250
set trim on
set heading off
set feedback off
spool &&SCHEMA..add_trandata.obey
 select 'add trandata &SCHEMA..'||table_name
 from dba_tables where owner = '&SCHEMA'
 and table_name in
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','LOCATIONS','REGIO
NS') ;
spool off

-- to run the script in gg:
GGSCI (WIN11SRC) 1> dblogin userid ggs_owner password g
GGSCI (WIN11SRC) 2> obey diroby/HR.add_trandata.obey

-- in gg: to confirm the table logging is enabled:
dblogin userid ggs_owner password g
info trandata hr.*

-- in db: to confirm the table logging is enabled:
select owner, log_group_name, table_name
from dba_log_groups where owner = 'HR';

-- to know which column values are being logged:
select * from USER_LOG_GROUP_COLUMNS where owner = 'HR';
```

Note: If you use ADD SCHEMATRANDATA command to add the supplement logging, the information should be obtained from **logmnr$always_suplog_columns**.

```
eg: select * from table(logmnr$always_suplog_columns('SCHEMAUSER','T'));
```

Note: If you find out GG is logging ALL the columns because of the absence of a key column, adding a key column is not enough to resolve the issue. You need to do the following as well (wildcare naming is acceptable):

```
GGSCI> delete trandata hr.EMPLOYEES
GGSCI> add trandata hr.EMPLOYEES
```

10. In Source: make sure the Manager process is up and running. If the Manager process is not installed, carry out the steps to create it as shown in this document.

```
1. verify the manager is running. It must be there and running
GGSCI (WIN11SRC) 2> info manager
Manager is running (IP port WIN11SRC.7840)

-- check its parameter file:
GGSCI (WIN11SRC) 2> edit params mgr


-- suggested settings:
port 7840
BOOTDELAYMINUTES 3
autostart ER *

PURGEOLDEXTRACTS dirdat/*, USECHECKPOINTS, MINKEEPDAYS 3
```

11. In Source: configure the source definition file.

```
GGSCI (WIN11SRC)> edit param defgen
DEFSFILE dirdef/source.def, PURGE
USERID GGS_OWNER, PASSWORD g
TABLE HR.REGIONS;
TABLE HR.LOCATIONS;
TABLE HR.JOB_HISTORY;
TABLE HR.JOBS;
TABLE HR.EMPLOYEES;
TABLE HR.DEPARTMENTS;
TABLE HR.COUNTRIES;

Note: you can get the list of the tables from the following code:
select 'TABLE HR.'||table_name ||';'
from dba_tables where owner = 'HR'
 and table_name in
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','LOCATIONS','REGIO
NS')

GGSCI (WIN11SRC)> exit

-- Execute the Source Definition Generator
E:\ggs_src>defgen paramfile dirprm\defgen.prm


-- transfer the Source Definition File to the target:
E:\ggs>copy \\win11src\ggs_src\dirdef\source.def E:\ggs\dirdef
```

12. Copy the data across from the source to the target using whatever you like of methods. In this tutorial, a linked server was used in SQL Server. Starting from this step and up till end of the procedure, no DML should be applied on the source.

```
1. Install Oracle Client Software in SQL Server machine. Select
"Administrator" installation type


2. Install "ODAC 11.2 Release 4 (11.2.0.3.0) with Oracle Developer Tools for
Visual Studio" on the same home used in step 1.


3. Edit tnsnames.ora file to configure connection to the source database

ORA11SRC =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = WIN11SRC)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = oradb)
    )
  )


4. In SQL Server: log in to the SQL Server instance>
       > Server Objects
       > Linked Servers
       > OraOLEDB.Oracle
       > right-Click> Properties
       > mark Allow inprocess
       > OK


5. Create a Linked Server pointed to the TNS name created above. Name it as
ORA11SRC and use the Oracle HR schema to login to Oracle database.


6. Copy the data across from Oracle into SQL Server

USE [HR]
GO

select * into [REGIONS] from openquery(ORA11SRC, 'SELECT * FROM HR.REGIONS');
select * into [LOCATIONS] from openquery(ORA11SRC, 'SELECT * FROM
HR.LOCATIONS');
select * into [JOB_HISTORY] from openquery(ORA11SRC, 'SELECT * FROM
HR.JOB_HISTORY');
select * into [JOBS] from openquery(ORA11SRC, 'SELECT * FROM HR.JOBS');
select * into [EMPLOYEES] from openquery(ORA11SRC, 'SELECT * FROM
HR.EMPLOYEES');
select * into [DEPARTMENTS] from openquery(ORA11SRC, 'SELECT * FROM
HR.DEPARTMENTS');
select * into [COUNTRIES] from openquery(ORA11SRC, 'SELECT * FROM
HR.COUNTRIES');
GO

Note: you can create the script to generate the copy code from DBA_TABLES
```

```
7. create the primary key constraints in SQL Server (must):
-- here is the format in T-SQL:
ALTER TABLE dbo.'||table_name ||' ADD CONSTRAINT pk_'||table_name ||' PRIMARY
KEY  ( ...

-- you can get the primary keys created in the source with a query like:
select N.TABLE_NAME , COLUMN_NAME
from DBA_CONS_COLUMNS C, DBA_CONSTRAINTS N
where C.CONSTRAINT_NAME = N.CONSTRAINT_NAME
AND C.OWNER='HR' AND N.CONSTRAINT_TYPE='P'
AND N.TABLE_NAME in
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','LOCATIONS','REGIO
NS')
ORDER BY 1
```

13. In Source: configure the local Extract (Change Capture Process).

```
1. verify the manager is running. It must be there and running
GGSCI (WIN11SRC) 2> info manager
Manager is running (IP port WIN11SRC.7840)

-- check its parameter file:
GGSCI (WIN11SRC) 2> view params mgr


-- suggested settings:
port 7840
BOOTDELAYMINUTES 3
autostart ER *

PURGEOLDEXTRACTS dirdat/*, USECHECKPOINTS, MINKEEPDAYS 3



2. create and edit the Extract parameter file (8 characters max):
GGSCI (WIN11SRC) >edit params LHRDEV1

-- add the following to it:

Extract LHRDEV1
-------------------------------------------------------------------
-- Local extract for HR schema
-------------------------------------------------------------------
SETENV (NLS_LANG = AMERICAN_AMERICA.WE8MSWIN1252)
-- this is needed, if you have more than one instance:
SETENV (ORACLE_HOME = "E:\oracle\product\11.2.0\dbhome_1")
SETENV (ORACLE_SID = "ORADB")


USERID GGS_OWNER@ORA11SRC, PASSWORD g
-- specify the two-character local Extract trail-file name
ExtTrail dirdat/L1

-- Table parameter must end with semicolon
TABLE HR.COUNTRIES;
TABLE HR.DEPARTMENTS;
TABLE HR.EMPLOYEES;
TABLE HR.JOBS;
TABLE HR.JOB_HISTORY;
TABLE HR.LOCATIONS;
TABLE HR.REGIONS;



3. add the Extract process
-- create the process and start checkpoint:
ADD EXTRACT LHRDEV1, TRANLOG, BEGIN NOW
-- add local extract trail file:
ADD EXTTRAIL dirdat/l1, EXTRACT LHRDEV1, MEGABYTES 100
-- start the extract (stop command stops it)
START EXTRACT LHRDEV1

/* verify the extract:  */
-- basic info displayed:
info extract LHRDEV1

-- detailed info displayed:
info extract LHRDEV1, DETAIL

-- get stats about the extract operation:
stats ext LHRDEV1


4. create the Data Pump parameter file:
-- no env variables should be defined
edit params PHRDEV1

Extract PHRDEV1
-------------------------------------------------------------------
-- Data Pump extract for HR schema
-------------------------------------------------------------------
-- source and desc identical
PassThru

RmtHost SERVER5, MgrPort 7840

-- remote trail file number prefix:
RmtTrail dirdat/L2

TABLE HR.COUNTRIES;
TABLE HR.DEPARTMENTS;
TABLE HR.EMPLOYEES;
TABLE HR.JOBS;
TABLE HR.JOB_HISTORY;
TABLE HR.LOCATIONS;
TABLE HR.REGIONS;



5. create the Data Pump process:
ADD EXTRACT PHRDEV1, EXTTRAILSOURCE dirdat/L1
ADD RMTTRAIL dirdat/L2, EXTRACT PHRDEV1, MEGABYTES 100
START EXTRACT PHRDEV1
```

```
-- verify the process:
info extract PHRDEV1
info extract PHRDEV1, detail
stats ext PHRDEV1
```

14. On the target SQL Server system: choose an ODBC connection option to the SQL Server. OLE DB is the one used by this tutorial. For other options, refer to the section "Replicat Database Connection Options in SQL Server".

    As it is chosen to configure the Replicate to connect to the target database through OLE DB, you should disable triggers, cascade delete constraints, and cascade update constraints on the target tables, or alter them to ignore changes made by the Oracle GoldenGate database user:

```
/* Implement the prerequisites of OLE DB connection */
-- to list the triggers in a database with their code:
use [HR]
GO

-- Get Triggers w/ Code
SELECT      Tables.Name TableName,
      Triggers.name TriggerName,
      Triggers.crdate TriggerCreatedDate,
      Comments.Text TriggerText
FROM      sysobjects Triggers
      Inner Join sysobjects Tables On Triggers.parent_obj = Tables.id
      Inner Join syscomments Comments On Triggers.id = Comments.id
WHERE      Triggers.xtype = 'TR'
      And Tables.xtype = 'U'
ORDER BY Tables.Name, Triggers.name


-- to list CASCADE DELETE constraints:
use [HR]
GO
SELECT name, delete_referential_action_desc
FROM sys.foreign_keys
WHERE delete_referential_action_desc = 'CASCADE'


/* Configure ODBC connection to SQL Server */
Follow the steps in "Configuring an ODBC connection to SQL Server"
this tutorial assumes a DNS of name "HR" is created
```

15. In the target system: configure the replicate.

```
1. make sure the manager is up and running (and has same port number defined
in MgrPort in the source Data Pump process) :
info mgr



2. add a checkpoint table name
GGSCI (server5)> DBLOGIN SOURCEDB HR
GGSCI (server5)> ADD CHECKPOINTTABLE dbo.GG_CHKPT
```

```
3. create the Replicate parameter file:
GGSCI (server5)> edit params RHRDEV1

Replicat RHRDEV1
-------------------------------------------------------------------
-- Replicat for HR Schema
-------------------------------------------------------------------
-- TARGETDB <dsn>, USERID <login>, PASSWORD <password>
TARGETDB hr
SourceDefs dirdef/source.def
HANDLECOLLISIONS
DISCARDFILE ./dirrpt/rhrdev1_discards.dsc, PURGE
-- semicolon is mandatroy
Map HR.REGIONS, Target dbo.REGIONS;
Map HR.LOCATIONS, Target dbo.LOCATIONS;
Map HR.JOB_HISTORY, Target dbo.JOB_HISTORY;
Map HR.JOBS, Target dbo.JOBS;
Map HR.EMPLOYEES, Target dbo.EMPLOYEES;
Map HR.DEPARTMENTS, Target dbo.DEPARTMENTS;
Map HR.COUNTRIES, Target dbo.COUNTRIES;



4. add the Replicate
ADD REPLICAT RHRDEV1, EXTTRAIL dirdat/l2, CHECKPOINTTABLE dbo.GG_CHKPT
START REPLICAT RHRDEV1


5. verify the Replicate is running
info replicat RHRDEV1
info replicat RHRDEV1, detail
```

Performance Tuning in GoldenGate

Performance Tuning Methodology

1. Define the performance requirements (Design Stage)

- What is the rate of data updates?
- How much data needs to be replicated?
- Are there periods when there is special processing?

Examples:

The replication lag must be less than 1 minute at least 80 percent of the time. If the replication lag exceeds 1 minute, a warning should be issued and the replication should be closely monitored. If the replication lag exceeds 5 minutes, a critical alert should be issued and the problem needs to be addressed immediately.

The replication lag from the OLTP database to the reporting database must be less than 1 hour at least 80 percent of the time. If the replication lag exceeds 8 hours, a warning should be issued. If the replication lag exceeds 24 hours, a critical alert should be issued and the problem needs to be addressed within the following day.

2. Create a performance baseline (during Normal Operation)

Item	Description	Suggested Tool/Command
Server stats	CPU utilization, disk utilization, and memory usage	SAR or IOSTAT (Linux) perfmon (Windows)
Extract and Replicate stats	Name and status of each processing group	info *
	Processing stats	stats STATS RHREMD1, TOTALSONLY *, REPORTRATE MIN
	Processing rate	Report file, number of records processed, the rate, and the delta
	Processing lag	info command lag ext or lag rep commands
	Process statistics	CPU usage, memory, IO of Extract and Replicate processes perfmon (Windows) top –U gger (Linux)
Database stats	statistical reports on the source and target database usage	Oracle AWR report SQL Server dashboard report
Network stats	network throughput and utilization	Windows System Monitor, Solarwinds software netstat command on Linux HP OpenView or Tivoli

Steps 3 through 6 should be done to resolve a performance issue:

3. Evaluate current performance

- Gather the same statistics you gathered in normal operations
- Compare gathered stats with the baseline

4. Determine the problem

- Determine the components where there are significant differences in their statistics
- Contact the stakeholders to determine the expected tuning results
- Drill down on the GoldenGate processes that appear to be running slowly.
- View the GoldenGate reports and compare the processing rates to try to determine when the problem began:

```
98650000 records processed as of 2011-03-09 19:27:48 (rate 4728,delta 3969)
98660000 records processed as of 2011-03-09 19:28:50 (rate 4728,delta 4439)
98670000 records processed as of 2011-03-09 19:29:53 (rate 3685,delta 4183)
98680000 records processed as of 2011-03-09 19:30:55 (rate 3247,delta 3366)
98690000 records processed as of 2011-03-09 19:31:57 (rate 2327,delta 2630)
98700000 records processed as of 2011-03-09 19:32:00 (rate 1727,delta 2257)
98710000 records processed as of 2011-03-09 19:33:02 (rate 1135,delta 1736)
98720000 records processed as of 2011-03-09 19:34:04 (rate 768,delta 1125)
98730000 records processed as of 2011-03-09 19:35:06 (rate 436,delta 923)
```

- Review the GoldenGate ggserr.log file (can be done by `VIEW GGSEVT` ggsci command) , and look for any errors or warnings.
- Review the server system logs for any error messages that could be contributing to the slow performance, such as a bad disk drive.
- Review the built-in database performance reports (such as AWR for Oracle or the SQL Server dashboard), and compare them to the reports from the baseline period.

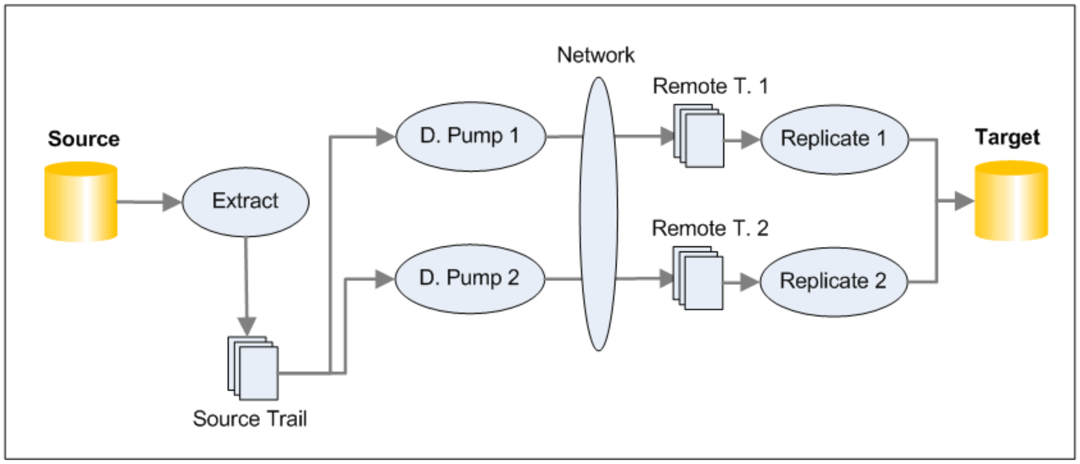
5. Design and implement a solution

- Design the action plan, its expected impact and the backout plan for every step in the plan
- Implement a step at a time and measure its impact

6. Repeat steps 3 to 5 as needed

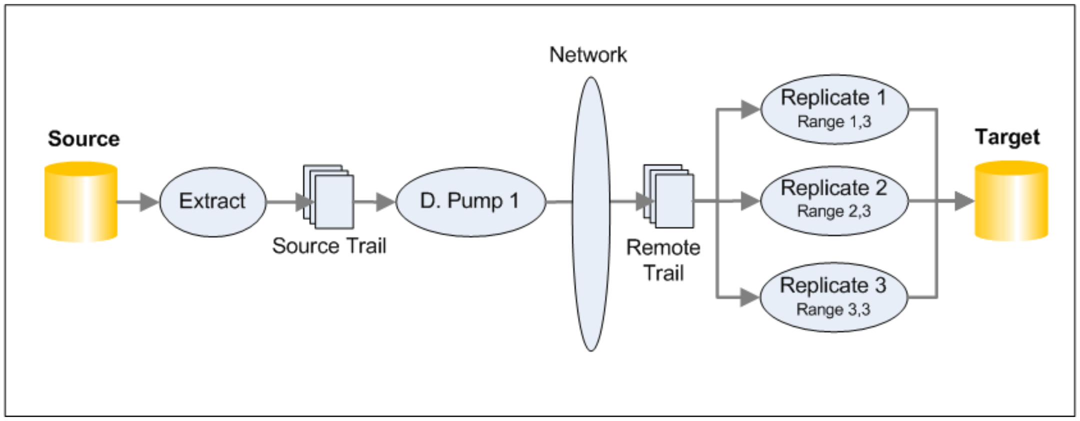
Using Parallel Extracts and Replicates

Oracle GoldenGate Extract and Replicate process groups can be configured to run in parallel. The following figure shows the configuration of a one-way replication with parallel data-pump Extract and Replicate groups:



You split the GoldenGate process groups by:

- **Table Filtering**: each group replicates different set of tables. You should keep tables related by SQL referential integrity constraints together in the same GoldenGate process groups.
- **Ranges**: use the @RANGE function to split the incoming rows into equal buckets using a GoldenGate hash algorithm on the key values. Those buckets are then distributed to gg parallel processes. The following figure shows how this configuration is set up:



Tutorial: Implementing Parallel Extracts and Replicates Using Table Filters

This tutorial is a modification on the replication environment created by the "Tutorial: Basic Replication between two Oracle Databases".

It implements parallel Extracts and Replicates as in the following figure:

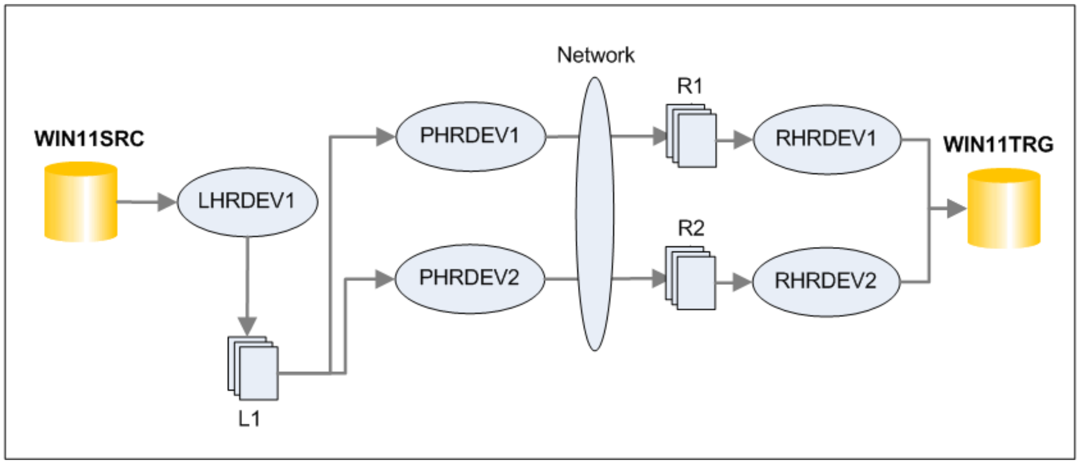


Table filtering will be implemented as follows:

Process	Tables
PHREMD1	EMPLOYEES JOB_HISTORY JOBS LOCATIONS
PHREMD2	COUNTRIES REGIONS

Note: Always make a backup of the GoldenGate parameter files before you make any change.

Note: The modification includes renaming the remote trail files from L2 to R1. This looks more convenient that linking L2 to RHRDEV1.

1. Backup the parameter files of all process groups in both source and target systems

2. Make sure all the process groups are up and running

```
info all
```

3. In source system, edit the existing data-pump parameter file PHREMD1 to add the table filtering:

```
edit params PHRDEV1

Extract PHRDEV1
-------------------------------------------------------------------
-- Data Pump extract for HR schema
-------------------------------------------------------------------
PassThru
RmtHost WIN11TRG, MgrPort 7840
RmtTrail dirdat/r1
```

```
ReportCount Every 10000 Records, Rate
Report at 01:00
ReportRollover at 01:15
DiscardFile dirrpt/PHRDEV1.dsc, Append
DiscardRollover at 01:00
Table HR.EMPLOYEES;
Table HR.JOBS;
Table HR.JOB_HISTORY;
Table HR.LOCATIONS;
Table HR.DEPARTMENTS;
```

4. In the source system, add the parameter file for the new data-pump Extract, PHRDEV2:

```
edit params PHRDEV2

Extract PHRDEV2
-------------------------------------------------------------------
-- Data Pump extract for HR schema
-------------------------------------------------------------------
PassThru
RmtHost WIN11TRG, MgrPort 7840
RmtTrail dirdat/r2
ReportCount Every 10000 Records, Rate
Report at 01:00
ReportRollover at 01:15
DiscardFile dirrpt/PHRDEV2.dsc, Append
DiscardRollover at 01:00
Table HR.COUNTRIES;
Table HR.REGIONS;
```

5. In the target system, edit the existing Replicate parameter file RHRDEV1 to add the table filtering:

```
edit params RHRDEV1

Replicat RHRDEV1
-------------------------------------------------------------------
-- Replicat for HR Schema
-------------------------------------------------------------------
USERID GGS_OWNER@ORA11TRG, PASSWORD g
HANDLECOLLISIONS
AssumeTargetDefs
ReportCount Every 30 Minutes, Rate
Report at 01:00
ReportRollover at 01:15
DiscardFile dirrpt/RHRDEV1.dsc, Append
DiscardRollover at 02:00 ON SUNDAY
Map HR.EMPLOYEES, Target HR.EMPLOYEES;
Map HR.JOBS, Target HR.JOBS;
Map HR.JOB_HISTORY, Target HR.JOB_HISTORY;
Map HR.LOCATIONS, Target HR.LOCATIONS;
Map HR.DEPARTMENTS, Target HR.DEPARTMENTS;
```

6. In the target system, add the parameter file for the new Replicate, RHRDEV2:

```
edit params RHRDEV2

Replicat RHRDEV2
```

```
-------------------------------------------------------------------
-- Replicat for HR Schema
-------------------------------------------------------------------
USERID GGS_OWNER@ORA11TRG, PASSWORD g
HANDLECOLLISIONS
AssumeTargetDefs
ReportCount Every 30 Minutes, Rate
Report at 01:00
ReportRollover at 01:15
DiscardFile dirrpt/RHRDEV2.dsc, Append
DiscardRollover at 02:00 ON SUNDAY
Map HR.COUNTRIES, Target HR.COUNTRIES;
Map HR.REGIONS, Target HR.REGIONS;
```

7. In the source system, stop the existing data-pump, PHRDEV1, and record EXTRBA and EXTSEQ from the info command.

```
stop ext PHRDEV1
GGSCI (WIN11SRC) 50> info ext PHRDEV1

EXTRACT    PHRDEV1   Last Started 2012-09-06 13:51   Status STOPPED
Checkpoint Lag       00:00:00 (updated 00:00:18 ago)
Log Read Checkpoint  File dirdat/l1000018
                     2012-09-06 13:51:46.995000  RBA 1052

-- EXTSEQ equals to the trail file number 18
-- EXTRBA: 1052
```

8. In the target system, when the Replicate RHRDEV1 has processed all the remaining changes from the current data-pump Extract trail and has zero lag showing, stop it:

```
REPLICAT   RHRDEV1   Last Started 2012-09-06 13:52   Status RUNNING
Checkpoint Lag       00:00:00 (updated 00:00:07 ago)
Log Read Checkpoint  File dirdat/l2000001
                     2012-09-06 13:51:46.910000  RBA 1186


stop rep RHRDEV1
```

9. In the source system, add the new data-pump Extract, PHRDEV2, and tell GoldenGate to start processing at the EXTRBA and EXTSEQ in the Local Extract trail L1 that you recorded.

```
ADD EXTRACT PHRDEV2, EXTTRAILSOURCE dirdat/l1, EXTSEQNO 18, EXTRBA 1052
ADD RMTTRAIL dirdat/r2, EXTRACT PHRDEV2, MEGABYTES 100

GGSCI (WIN11SRC) 56> info ext PHRDEV*

EXTRACT    PHRDEV1   Last Started 2012-09-06 13:51   Status STOPPED
Checkpoint Lag       00:00:00 (updated 00:28:09 ago)
Log Read Checkpoint  File dirdat/l1000018
                     2012-09-06 13:51:46.995000  RBA 1052

EXTRACT    PHRDEV2   Initialized   2012-09-06 15:14   Status STOPPED
Checkpoint Lag       00:00:00 (updated 00:00:29 ago)
Log Read Checkpoint  File dirdat/l1000018
                     First Record  RBA 1052
```

10. In the target system, add the new Replicate, RHRDEV2, and link it to the new R2 trail from data-pump Extract PHRDEV2 for processing. Because R2 is a new trail, you can take the default to start processing from the beginning of the trail.

```
dblogin userid ggs_owner@ora11trg password g
ADD CHECKPOINTTABLE HR.GG_CHKPT2
ADD REPLICAT RHRDEV2, EXTTRAIL dirdat/r2, CHECKPOINTTABLE HR.GG_CHKPT2

GGSCI (WIN11TRG) 30> info rep *

REPLICAT   RHRDEV1   Last Started 2012-09-06 13:52   Status STOPPED
Checkpoint Lag       00:00:00 (updated 00:19:41 ago)
Log Read Checkpoint  File dirdat/l2000001
                     2012-09-06 13:51:46.910000  RBA 1186

REPLICAT   RHRDEV2   Initialized   2012-09-06 15:23   Status STOPPED
Checkpoint Lag       00:00:00 (updated 00:00:10 ago)
Log Read Checkpoint  File dirdat/r2000000
                     First Record  RBA 0
```

11. In the target system, L2 remote trail files will be replaced by R1

```
ALTER REPLICAT RHRDEV1 , EXTTRAIL dirdat/r1
ADD RMTTRAIL dirdat/r1, EXTRACT PHRDEV1
```

12. In the source system, start the data pump process group:

```
-- to start PHRDEV1 and PHRDEV2
start PHRDEV*

info ext PHRDEV*
```

13. In the target system, start the replicate process group:

```
start rep RHRDEV*
```

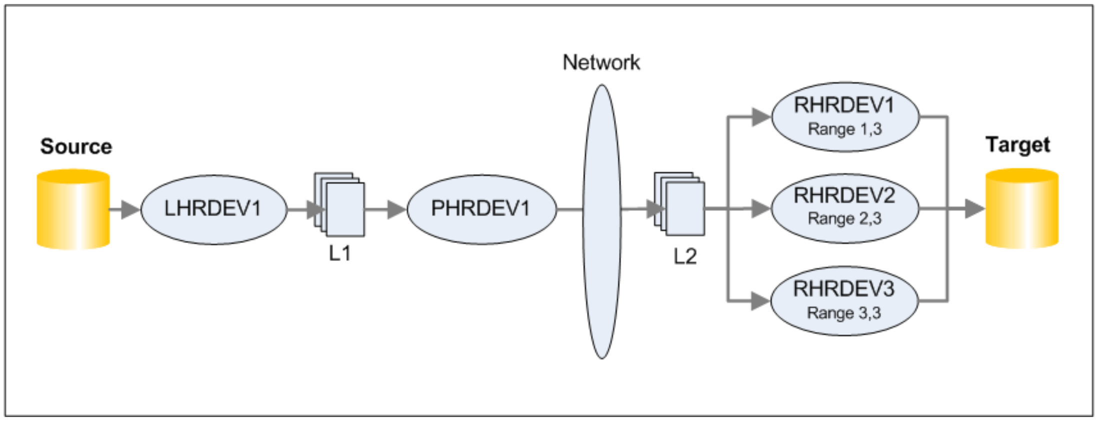
14. Verify the set up is running as expected. Make changes on the data and check the following:

```
-- make sure the RBA is increasing:
info RHRDEV*

-- make sure changes made on specific table are processed:
stats extract <tablename>
stats replicat <tablename>
```

Tutorial: Implementing Parallel Extracts and Replicates Using Key Ranges

This tutorial is a modification on the replication environment created by the "Tutorial: Basic Replication between two Oracle Databases".

It implements parallel Replicates using *key ranges* as in the following figure:



Note: Always make a backup of the GoldenGate parameter files before you make any change.

Note: The modification includes renaming the remote trail files from L2 to R1. This looks more convenient that linking L2 to RHRDEV1.

1. Backup the parameter files of all process groups in both source and target systems

2. Make sure all the process groups are up and running

```
info all
```

3. Edit the existing Replicate parameter file RHRDEV1 to add the RANGE filtering:

```
Replicat RHRDEV1
-------------------------------------------------------------------
-- Replicat for HR Schema
-------------------------------------------------------------------
USERID GGS_OWNER@ORA11TRG, PASSWORD g
HANDLECOLLISIONS
AssumeTargetDefs
ReportCount Every 30 Minutes, Rate
Report at 01:00
ReportRollover at 01:15
DiscardFile dirrpt/RHRDEV1.dsc, Append
DiscardRollover at 02:00 ON SUNDAY

Map HR.EMPLOYEES, Target HR.EMPLOYEES,COLMAP (USEDEFAULTS), FILTER (@RANGE
(1,3));
Map HR.JOBS, Target HR.JOBS,COLMAP (USEDEFAULTS), FILTER (@RANGE (1,3));
Map HR.JOB_HISTORY, Target HR.JOB_HISTORY,COLMAP (USEDEFAULTS),
FILTER (@RANGE (1,3));
Map HR.LOCATIONS, Target HR.LOCATIONS,COLMAP (USEDEFAULTS), FILTER (@RANGE
(1,3));
Map HR.DEPARTMENTS, Target HR.DEPARTMENTS,COLMAP (USEDEFAULTS),
FILTER (@RANGE (1,3));
Map HR.COUNTRIES, Target HR.COUNTRIES,COLMAP (USEDEFAULTS), FILTER (@RANGE
(1,3));
Map HR.REGIONS, Target HR.REGIONS,COLMAP (USEDEFAULTS), FILTER (@RANGE (1,3));
```

4. Add the parameter files for the new Replicates: RHRDEV2 and RHRDEV3:

```
edit params RHRDEV2

Replicat RHRDEV2
-------------------------------------------------------------------
-- Replicat for HR Schema
-------------------------------------------------------------------
USERID GGS_OWNER@ORA11TRG, PASSWORD g
HANDLECOLLISIONS
AssumeTargetDefs
ReportCount Every 30 Minutes, Rate
Report at 01:00
ReportRollover at 01:15
DiscardFile dirrpt/RHRDEV2.dsc, Append
DiscardRollover at 02:00 ON SUNDAY

Map HR.EMPLOYEES, Target HR.EMPLOYEES,COLMAP (USEDEFAULTS), FILTER (@RANGE
(2,3));
Map HR.JOBS, Target HR.JOBS,COLMAP (USEDEFAULTS), FILTER (@RANGE (2,3));
Map HR.JOB_HISTORY, Target HR.JOB_HISTORY,COLMAP (USEDEFAULTS),
FILTER (@RANGE (2,3));
Map HR.LOCATIONS, Target HR.LOCATIONS,COLMAP (USEDEFAULTS), FILTER (@RANGE
(2,3));
Map HR.DEPARTMENTS, Target HR.DEPARTMENTS,COLMAP (USEDEFAULTS),
FILTER (@RANGE (2,3));
Map HR.COUNTRIES, Target HR.COUNTRIES,COLMAP (USEDEFAULTS), FILTER (@RANGE
(2,3));
Map HR.REGIONS, Target HR.REGIONS,COLMAP (USEDEFAULTS), FILTER (@RANGE (2,3));


edit params RHRDEV3

Replicat RHRDEV3
-------------------------------------------------------------------
-- Replicat for HR Schema
-------------------------------------------------------------------
USERID GGS_OWNER@ORA11TRG, PASSWORD g
HANDLECOLLISIONS
AssumeTargetDefs
ReportCount Every 30 Minutes, Rate
Report at 01:00
ReportRollover at 01:15
DiscardFile dirrpt/RHRDEV3.dsc, Append
DiscardRollover at 02:00 ON SUNDAY

Map HR.EMPLOYEES, Target HR.EMPLOYEES,COLMAP (USEDEFAULTS), FILTER (@RANGE
(3,3));
Map HR.JOBS, Target HR.JOBS,COLMAP (USEDEFAULTS), FILTER (@RANGE (3,3));
Map HR.JOB_HISTORY, Target HR.JOB_HISTORY,COLMAP (USEDEFAULTS),
FILTER (@RANGE (3,3));
Map HR.LOCATIONS, Target HR.LOCATIONS,COLMAP (USEDEFAULTS), FILTER (@RANGE
(3,3));
Map HR.DEPARTMENTS, Target HR.DEPARTMENTS,COLMAP (USEDEFAULTS),
FILTER (@RANGE (3,3));
```

```
Map HR.COUNTRIES, Target HR.COUNTRIES,COLMAP (USEDEFAULTS), FILTER (@RANGE
(3,3));
Map HR.REGIONS, Target HR.REGIONS,COLMAP (USEDEFAULTS), FILTER (@RANGE (3,3));
```

5. Stop the data-pump Extract, PHRDEV1 and make sure the lag on the Replicate goes down to zero:

```
stop ext PHRDEV1
info ext PHRDEV1

info RHRDEV1
```

6. Record the SEQ (sequence number of the file Log Read Checkpoint) and RBA output from info RHRDEV1. You need those values when you add the new Replicats so they begin processing at the correct location in the data-pump Extract trail file l2.

```
GGSCI (WIN11TRG)> info rhrdev1
GGSCI (WIN11TRG)> stop rhrdev1
```

7. Add the other replicates: RHRDEV2 and RHRDEV3. Make them start from the recorded point.

```
ADD REPLICAT RHRDEV2, EXTTRAIL dirdat/l2,EXTSEQNO 1, EXTRBA 1186,
CHECKPOINTTABLE HR.GG_CHKPT
ADD REPLICAT RHRDEV3, EXTTRAIL dirdat/l2, EXTSEQNO 1, EXTRBA 1186,
CHECKPOINTTABLE HR.GG_CHKPT
info rep RHRDEV*
```

8. In the source system, start the data-pump Extract:

```
start ext PHRDEV1
```

9. Start all the three Replicates

```
start rep RHRDEV*
```

10. Verify the set up is running as expected. Make changes on the data and check the following:

```
-- make sure the RBA is increasing:
info RHRDEV*

-- make sure changes made on specific table are processed:
stats extract <tablename>
stats replicat <tablename>
```

Using the Replicate Parameter BATCHSQL

- Use BATCHSQL parameter to instruct the Replicate to operate in BATCHSQL mode. When operating in this mode, the Replicate organizes similar SQL statements into batches within a memory queue, and then it applies each batch in one database operation. A batch contains SQL statements that affect the same table, operation type (insert, update, or delete), and column list.
- Refer to its limitations in the Reference Guide documentation
- The parameter will add further statistics information to the Replicate generated report
- When you configure BATCHSQL, you should review your Replicate report file and compare the processing rates to your baseline reports and determine if you're getting performance improvements.

Using the Replicate Parameter GROUPTRANSOPS

- The GROUPTRANSOPS parameter controls the number of SQL operations in the Replicate transaction when operating in normal mode. Increasing its value may improve the performance of the Replicate by:
    - Reducing the number of transactions executed by the Replicate
    - Reducing I/O activity to the checkpoint file/table
- Setting it to too high value may increase the latency though.
- Its default value is 1000

Tuning Network Performance Issues

To make sure you have network performance issues:

- Compare current network performance statistics with you baseline
- While the data is being changed, run "info ext LHRDEV1" and monitor the value RBA (if you would like to see the info history, you can use the command "info ext LHRDEV1, showch 10")
- Check the progress of the Replicate using the command "send replicat RHRDEV1, status"
- If RBA is increasing in the Extract side and in a much slower rate (or even not increasing at all) in the Replicate side, this is usually a symptom of a network bottleneck
- In such a situation, the data-pump Extract backs up with transactions due to the network bottleneck and eventually runs out of internal memory and then abends

Possible Remedies:

- Parallel data pumps alleviate network bandwidth limitations caused by using a single data-pump process.
- Adjust some additional network tuning options on the RMTHOST parameter. You see those next.

Tuning the RMTHOST Parameter

- Consider increasing the TCP buffer size. This size is limited by the OS though. In Linux, you can check the following settings to determine the maximum TCP transmit and receive buffer sizes:

```
net.core.rmem_max = 262144
net.core.wmem_max = 262144
```

- Consult the system administrator to increase them, if they are not set to sufficient values
- In gg, you can utilize larger TCP buffer sizes using RmtHost parameter:

```
RmtHost targetserver, MgrPort 7809, TCPBUFSIZE 100000
```

- You can also compress the packets (more CPU consumption):

```
RmtHost targetserver, MgrPort 7809, TCPBUFSIZE 100000, COMPRESS
```

More Oracle GoldenGate Tuning Tips

Beside the tools mentioned in the previous sections, consider the following:

- Place GoldenGate files on the fastest available storage and storage controller.
- RAID 0+1 is recommended over RAID 5.
- Some commands run by the Replicate may perform badly in the target system database. Usually, this required some performance tuning actions in the target database. In this case, consider the following actions:
    - Verify that the target tables used by GoldenGate have unique keys and SQL indexes are created on those keys.
    - Verify that database optimizer statistics are kept up to date to ensure that the database chooses the most efficient SQL execution plans.
    - Make sure the database tables that are the target of replication aren't overly fragmented. If so, they should be reorganized to eliminate the fragmentation.
    - Run database performance reports such as AWR for Oracle database and verify there are no major performance issues.

Monitoring Oracle GoldenGate

Monitoring Strategy Targets

The main target of monitoring any system is to achieve the following:

- Pro-act to an issue before it occurs (based on alerting facility)
- Facilitate detecting and identifying errors

Monitoring Points

Any component (internal or external for gg) that may affect GoldenGate operation should be monitored. Basically, the following monitoring points should be included in your monitoring strategy:

- GoldenGate processes: Manager, Extract, Data Pump and Replicate processes
- System components: CPU, server memory and storage

The following section shows the list of some monitoring factors, the monitoring tool to use for it and what to look at for troubleshooting.

Trail Files Storage

The storage should tolerate a defined network downtime.

If the Extract process has been down for long time, data might be lost.

Monitoring Tool:

- Consult system administrator about the tool to monitor the trail files storage space

What to look at:

- Storage growth rate history
- Free space

Getting the Maximum Network Throughput

If GoldenGate has to be down for 5 hours due to a scheduled outage, how long GoldenGate takes to catch up to zero lag.

Monitoring Tool:

- Stop the data pump for a few hours and then restart it.

What to look at:

- Replicate activity rate and time it needed to catch up

Monitoring All Running Processes

Indicator of GoldenGate's overall health

Monitoring Tool:

- info all

What to look at:

- Lag should be 0 and the "Time Since Chkpt" should be less than 10 seconds on normal operations

Monitoring Extract Sources

Extract possible sources: online redo log, archived redo log and copies of archived log stored in ALTARCHIVELOGDEST

Monitoring Tool:

- in gg side:
```
send extract extggs1 status
info extract extggs1, detail
```
- in db side:
```
SELECT A.MEMBER, B.SEQUENCE#,B.STATUS, B.FIRST_CHANGE#,
 B.NEXT_CHANGE#, B.ARCHIVED
FROM V$LOGFILE A, V$LOG B
WHERE A.GROUP# = B.GROUP#;

SELECT SEQUENCE#, NAME, FIRST_CHANGE#, NEXT_CHANGE#
FROM V$ARCHIVED_LOG;
```

What to look at:

- Log Read Checkpoint Seqno

Monitoring the GoldenGate Activity Rate

You can customize the gg activity rate

Monitoring Tool:

- `stats lhrdev1`
- `stats phrdev1 totalsonly *`
- `stats rhrdev1 totalsonly *`

What to look at:

- output figures of the commands

Monitoring Lag of Each Group Process

Monitoring Tool:

- `info lhrdev1`
- `info extract extggs1, showch 10`

- `lag extract lhrdev1`
- `send extract lhrdev1, getlag`
- Use the Manager parameters LAGINFOSECONDS, LAGINFOMINUTES, or LAGINFOHOURS parameter to specify a lag threshold; if lag exceeds the specified value, Oracle GoldenGate reports lag information to the error log.

What to look at:

- output figures of the commands

Automating the Monitoring

- Best run in a separate system to avoid losing monitoring scripts and output if the production system goes down
- Some scripts below are written in ksh to be run in Linux systems. For Windows platforms, you can use PowerShell for the same purpose.

Verifying GoldenGate Processes are Up

- You can automate the following script to run every every 5 minutes
- If GoldenGate hangs, this script won't catch the error. Checking lags should catch it though.

```
#!/bin/ksh
EMAIL_LIST="myemail@testme.com myemail2@testme.com"
gghome=/ggs
host=`hostname`
check=`ps -ef|grep lhrdev1|grep -v "grep lhrdev1"|wc -l`;
check_num=`expr $check`
if [ $check_num -le 0 ]
then
 echo "GoldenGate Extract LHRDEV1 is down on $host." > $gghome/extggs1.out
 tail -200 $gghome/dirrpt/eextggs1.rpt >> $gghome/extggs1.out
 mailx -s "GoldenGate LHRDEV1 down on $host" $EMAIL_LIST < $gghome/extggs1.out
fi
check=`ps -ef|grep phrdev1|grep -v "grep phrdev1"|wc -l`;
check_num=`expr $check`
if [ $check_num -le 0 ]
then
 echo "GoldenGate Extract PHRDEV1 is down on $host." > $gghome/dpggs1.out
 tail -200 $gghome/dirrpt/dpggs1.rpt >>$gghome/dpggs1.out
 mailx -s "GoldenGate Data Pump PHRDEV1 down on $host" $EMAIL_LIST <
$gghome/dpggs1.out
fi
```

Monitoring Lag Scripts

- This lag-monitoring script reports the overall lag from the source database to target database
- You must run this script on the target database
- The concept is to have two dummy identical tables: one in the source and one in the target system. These tables are added to the GoldenGate setup. There is a job in the source which periodically keeps updating the table with the current time. On the target database, the monitoring job compares the lag between the maximum time received in the target and current time and issues an alert if it exceeds specific threshold.

```
/* Original Code Source: Expert Oracle GoldenGate, by: Ben Prusinski, Steve
Phillips, and Richard Chung, Apress
Code modified from source */

-- will hold lag history (Target)
CREATE TABLE GGS.GG_LAG_HISTORY
( AUDIT_TIME DATE CONSTRAINT GG_LAG_HISTORY_PK PRIMARY KEY,
  LAG NUMBER
);

-- Dummy Source table
-- Map this table in the Extract that you are monitoring
CREATE TABLE GGS.SOURCE_TABLE
(UPDATE_DATE DATE,
 UPDATE_ID NUMBER
);

-- Dummy Target table. Map this in the replicat that you are replicating
CREATE TABLE GGS.TARGET_TABLE
(UPDATE_DATE DATE,
 UPDATE_ID NUMBER
);


-- Schedule this to run every five or ten seconds.
CREATE OR REPLACE PROCEDURE GGS.GET_LAG
IS
 L_MAX_TIME DATE;
 L_LAG NUMBER;
BEGIN
SELECT MAX (UPDATE_DATE)
INTO L_MAX_TIME
FROM GGS.TARGET_TABLE;

-- if target and source are in different timezone,
--  add timezone offset value in this calculation
SELECT (SYSDATE - L_MAX_TIME) * 24 * 60
INTO L_LAG
FROM DUAL;
INSERT INTO GGS.GG_LAG_HISTORY (AUDIT_TIME, LAG)
VALUES (SYSDATE, L_LAG);
COMMIT;

------ email alerts if the lag is more than 5 minutes.
IF L_LAG > 5
THEN
---- if you want Oracle to send email alert, you can use this.
GGS.SEND_EMAIL ('myemail@testme.com myemail2@testme.com',
 'Golden Gate Admin',
 'ggs@mycompany.com',
 'Golden Gate Data Is Behind',
 'The Lag Is ' || to_char( L_LAG ) || ' Minutes');
END IF;
END GET_LAG;
/


---- this is the send_email procedure for Oracle
CREATE OR REPLACE PROCEDURE GGS.SEND_EMAIL (I_TO_EMAIL IN VARCHAR2,
 I_FROM_NAME IN VARCHAR2,
 I_FROM_EMAIL IN VARCHAR2,
 I_SUBJECT IN VARCHAR2,
 I_MESSEGE IN VARCHAR2)
IS
 C UTL_SMTP.CONNECTION;
 L_EMAIL_START_POS NUMBER (5) := 1;
 L_EMAIL_END_POS NUMBER (5) := 0;
 L_EMAIL_CNT NUMBER (5) := 0;
 L_EMAIL VARCHAR2 (250);
 L_LAST_EMAIL_ADDR BOOLEAN := FALSE;
PROCEDURE SEND_HEADER (NAME IN VARCHAR2, HEADER IN VARCHAR2)
AS
BEGIN
 UTL_SMTP.WRITE_DATA (C, NAME || ': ' || HEADER || UTL_TCP.CRLF);
END;

BEGIN
 C := UTL_SMTP.OPEN_CONNECTION ('your.mail.relay.server');
 UTL_SMTP.HELO (C, 'foo.com');
 UTL_SMTP.MAIL (C, I_FROM_EMAIL);
LOOP
 L_EMAIL_CNT := L_EMAIL_CNT + 1;
 L_EMAIL_END_POS := INSTR (STR1 => I_TO_EMAIL, STR2 => ',',
 POS => 1, NTH => L_EMAIL_CNT);
 L_LAST_EMAIL_ADDR := (L_EMAIL_END_POS <= 0);
 L_EMAIL_END_POS := CASE
   WHEN (L_EMAIL_END_POS <= 0) THEN LENGTH (I_TO_EMAIL) + 1
   WHEN (L_EMAIL_START_POS = 0) THEN L_EMAIL_END_POS - 1
   ELSE L_EMAIL_END_POS
   END;
 L_EMAIL := SUBSTR (STR1 => I_TO_EMAIL,
  POS => L_EMAIL_START_POS,
  LEN => (L_EMAIL_END_POS - L_EMAIL_START_POS));
 UTL_SMTP.RCPT (C, L_EMAIL);
EXIT WHEN L_LAST_EMAIL_ADDR;
 L_EMAIL_START_POS := L_EMAIL_END_POS + 1;
END LOOP;
 UTL_SMTP.OPEN_DATA (C);
 SEND_HEADER ('From', '"' || I_FROM_NAME || '" <' || I_FROM_EMAIL || '>');
 SEND_HEADER ('To', '"Recipient" <' || I_TO_EMAIL || '>');
 SEND_HEADER ('Subject', I_SUBJECT);
 UTL_SMTP.WRITE_DATA (C, UTL_TCP.CRLF || I_MESSEGE);
 UTL_SMTP.CLOSE_DATA (C);
 UTL_SMTP.QUIT (C);
EXCEPTION
 WHEN UTL_SMTP.TRANSIENT_ERROR OR UTL_SMTP.PERMANENT_ERROR THEN
 BEGIN
  UTL_SMTP.QUIT (C);
  EXCEPTION
  WHEN UTL_SMTP.TRANSIENT_ERROR OR UTL_SMTP.PERMANENT_ERROR
  THEN
   NULL;
 END;
 RAISE_APPLICATION_ERROR ( -20000, 'Failed: ' || SQLERRM );
END SEND_EMAIL;
/
```

Monitoring

- The following scripts reports CPU and memory usage of GoldenGate Extract processes.

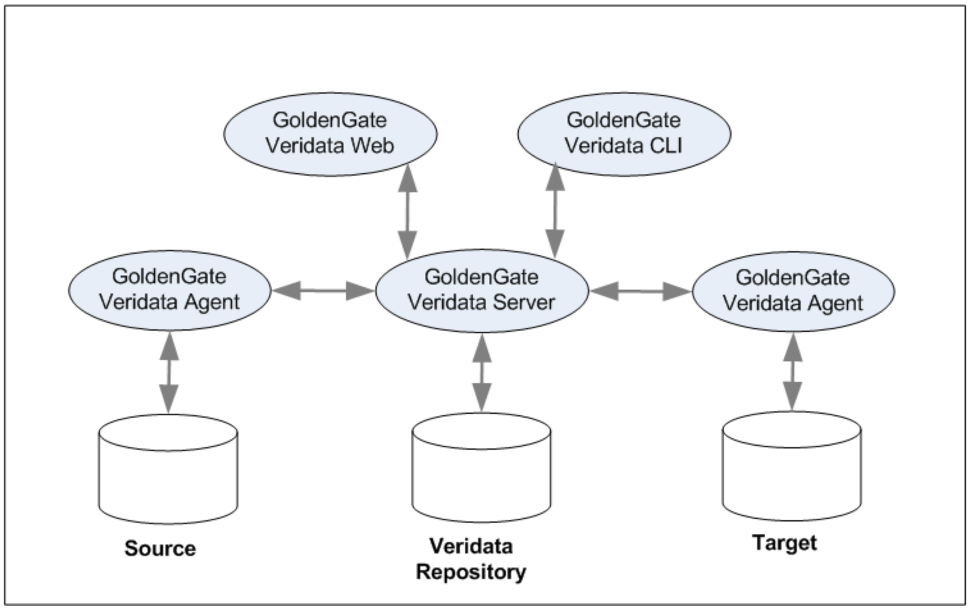
```
Checking Memory and CPU Scripts

#!/bin/ksh
EMAIL_LIST="myemail@testme.com myemail2@testme.com"
```

```
host=`hostname`
cd /ggs/scripts
rm top.log
rm top.rpt
top -h -f top.log -n 50
cat top.log | grep extract > top.rpt
echo "" >> top.rpt
echo "Free Memory: " >> top.rpt
vmstat 1 2 | tail -1 | awk '{printf "%d%s\n", ($5*4)/1024, "MB" }' >> top.rpt
mailx -s "GoldenGate Memory Usage on $host" $EMAIL_LIST < /ggs/scripts/top.rpt


# demo:
For the output:
19 ? 14000 goldengate 152 20 2331M 1201M run 602:09 31.64 31.59 extract
Its description:
31.59% of one CPU
Memory: 2331MB total and 1201MB in resident memory
```

Oracle GoldenGate Veridata

Oracle GoldenGate Veridata is a high-speed data comparison solution that identifies and reports on data discrepancies between heterogeneous databases without interrupting ongoing business processes. The comparison made by this tool can be performed on changing data.

Oracle GoldenGate Veridata Components



Tutorial: Installing Oracle GoldenGate Veridata Server

The following procedure installs the Windows platform version of the software using an Oracle repository. To comply with the tutorials in this document, the installation procedure was applied on a Windows XP system named as GENERALPC.

Note: For installing GoldenGate Veridata in other platforms, refer to the documentation "Oracle GoldenGate Veridata Administrator's Guide".

1. Download the software from the following link:

   http://www.oracle.com/technetwork/middleware/goldengate/downloads/index.html?ssSourceSiteId=ocomen

   or

   http://edelivery.oracle.com

2. Install Oracle software in the server to connect to the GoldenGate Veridata repository or create an Oracle database in the same server. Make sure the tnsnames.ora is configured to connect to the database repository

3. Create a database user to host the repository:

```
create tablespace ggv_tbs
datafile 'E:\ORADATA\ORADB\GGV_DATA.DBF' size 100m
autoextend on maxsize 4g;

create user ggv identified by g
default tablespace ggv_tbs;

create role VERIDATA_ROLE;

grant CREATE SESSION to VERIDATA_ROLE ;
grant CREATE TABLE to VERIDATA_ROLE ;
grant CREATE VIEW to VERIDATA_ROLE ;
grant CREATE PROCEDURE to VERIDATA_ROLE ;
grant CREATE SYNONYM to VERIDATA_ROLE ;
grant VERIDATA_ROLE to ggv;
alter user ggv QUOTA UNLIMITED on ggv_tbs;

-- Connection configuration returns error till this grant is issued:
grant dba to ggv;
```

4. Run the GoldenGate Veridata software setup and follow the Wizard steps

5. If you chose to create GoldenGate Veridata Services, the Web interface services is by default created with its startup set to "Manual". You can change it to "Automatic".

Tutorial: Installing Oracle GoldenGate Veridata C-agent

The following procedure installs the Oracle GoldenGate Veridata C-agent on Windows system. Perform the installation procedure on the source and target systems (WIN11SRC and WIN11TRG).

Note: For installation on other platforms, refer to the documentation "Oracle GoldenGate Veridata Administrator's Guide".

1. Download the software "Oracle GoldenGate Veridata V3.0.0.11.2 C Agent for Oracle 11g on Windows (32-bit)" from the following link:

   http://edelivery.oracle.com

2. Create a directory for the agent and unzip the software files into it.

```
E:\mkdir ggv
```

3. Change the current directory to the software directory, run ggsci and issue the "CREATE SUBDIRS" command in its prompt:

```
E:\ggv>ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 10.4.1.0.1_01 13559475 OGGVDT_3.0.0.11.2_PLATFORMS_120626.0940
Windows (optimized), Oracle 11g on Jun 26 2012 12:41:44

Copyright (C) 1995, 2012, Oracle and/or its affiliates. All rights reserved.


GGSCI (WIN11SRC) 1> CREATE SUBDIRS

Creating subdirectories under current directory E:\ggv

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

4. Specify a custom name for the Manager process different from the one used by the "Oracle GoldenGate replication software". Then perform the following:

```
EDIT PARAMS ./GLOBALS
MGRSERVNAME GGVMGR

Save the file and close it.
```

5. Install the Manager as a Windows service. In command prompt execute the following:

```
E:\ggv>install ADDSERVICE AUTOSTART
Service 'GGVMGR' created.
Install program terminated normally.
```

6. Configure the GoldenGate Veridata Manager process. In ggsci command prompt:

```
EDIT PARAMS MGR
port 7808

stop mgr
start mgr
info mgr
```

Setting Up the Veridata Compares

The procedure assumes Oracle GoldenGate Veridata server software has been installed. It also assumes the agent software has been installed on both the source and target systems. An installation procedure is presented in the previous section.

Following are the high-level steps to setup the Veridata compares:

1. Create database connections, source and target.
2. Create group, a group of comparing pairs.
3. Create compare pairs, what tables to compare.
4. Create job, to run one or more groups.
5. Create profile (how you want it to run.) This is optional, as you can use the default profile.
6. Run job.

Tutorial: Setting Up Basic Veridata Compares

Following are further details for performing the procedure:

1. Create database connections: in Oracle GoldenGate Veridata Web, click on **Connection Configuration**> **New button.** Follow the steps to connection two connections: WIN11SRC and WIN11TRG to connect to the source and target systems. When asked about the database username, use GGV.

2. In Group Configuration, create a group and add the source and target connections to it.

   Group Configuration> **Name**: GROUP1> **Next**> **Source**: WIN11SRC, **Destination**: WIN11TRG> **Finish**> **OK**

3. Create compare pairs.

   select GROUP1> **Edit**> click **Go to Compare Pair Configuration**> click **Pattern Mapping**> **Source,Schema**:HR **Target,Schema**:HR> **Table Mapping Method**, **select Map Source and Target Tables using SQL % ..**: Employees> **Generate Mappings**> **Save**

4. Create a Veridata job

   Job Configuration> **New**> follow the instructions. For this tutorial, you use the default profile. Name job: Compare_Employees_Job

5. Run the Veridata job. You can run the compare job by Veridata Web or you can use the command "vericon.exe –job <job name>"

   Job Configuration> select "Compare_Employees_Job"> **Run**> **Retrieve Compare List**> **Run Job**

   When the job is finished, see **Compare Paris**> **Comparison Status** column> see **Comparison Reports** column

Using Vericom Command Line

Vericom command-line scripts are usually used to automate the scheduled jobs.

```
C:\gg_veridata\server\bin>vericom.exe -job <jobname>

# to capture details message:
vericom.exe -job <jobname> -w
```

You can also redirect the output to a file and email to the monitoring team. The –w or –wp arguments will return the correct Exit codes.

There are six exit statues:

0: The Vericom command finished successfully. If –w or –wp was not specified, the exit status is 0 if the job started successfully. The actual compare errors will not be captured.

1: Invalid vericom syntax was used.

2: Vericom could not find the connection information by looking in the veridata.loc or veridata.cfg, and the connection parameter "{ -url | -u } <url>" is not specified with the command arguments.

3: Syntax is correct, but missing required flags. For example, if –j or –c is specified, then –g <group> input is required.

4: Job ran successfully, but the compare status has something other than in–sync.

5: Communication error with Oracle Veridata Server.

Setting Up Role-Based Security in Veridata

- Veridata role-based security is very easy to set up. It already defines a fixed number of roles. You just need to add users or groups to the pre-defined roles, then Veridata will apply the security for you.

`http://<hostname>:<port>/admin`

- The default Apache Tomcat Web Server port is 8830. To change the default port, modify the server.xml file in the `<Veridata_install_dir>/server/web/conf` directory. Search for 8830 and replace it with another port number.
- Here are the descriptions of the role names:
  - **Administrator**: This role can perform all functions.
  - **DetailReportView**: This is the ReportViewer role, but it can also view the out-ofsync report information through the Web interface or at the file level. Some business-sensitive information may be in the detail reports.
  - **PowerUser**: As with the admin role, you can do everything except any configuration function on the server side.
  - **ReportViewer**: With this role, you only can view the reports, jobs, and configurations.

Using Oracle GoldenGate Directory

About Oracle GoldenGate Director

Oracle GoldenGate Director is a multi-tiered GUI application that enables you to administer and monitor GoldenGate. With Director, you can:

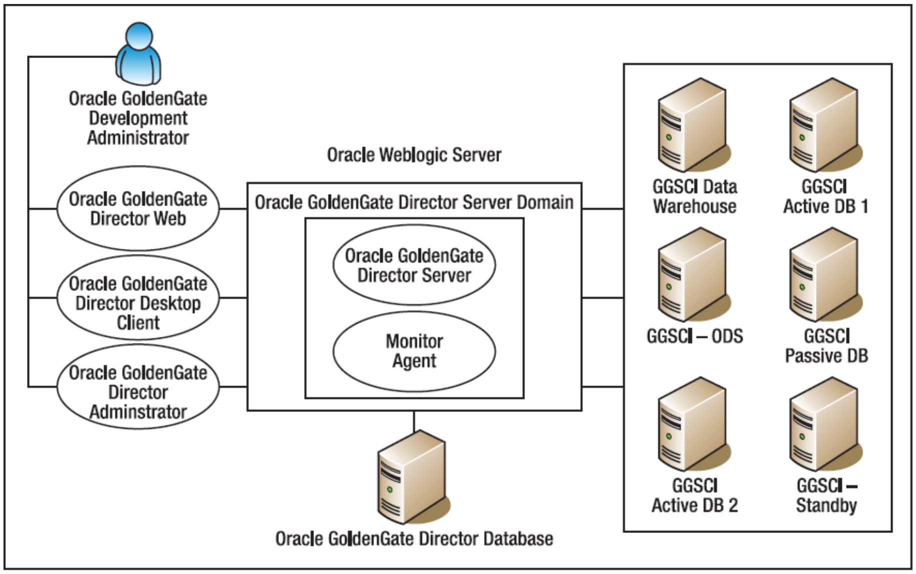
- manage the GoldenGate processes, without the need to login to the GoldenGate server, and easier than ggsci command line
- view the near real-time lag on each process
- easily setup intial load task

Note: Director does not support macros defined within a parameter file.

About Oracle GoldenGate Director Components

Oracle GoldenGate Director has the following components:

- Director Administrator: that is used to configure the GoldenGate Director
- Director Server/database: that manages all the jobs and runs as a WebLogic domain
- Director Web: that is used to monitor and control the Extract and Replicate processes (you cannot change their parameters though)
- Desktop Client: that is used to do all the Director tasks (except the Manager's parameters because it actually uses the Manager process in its operation)
- GoldenGate GGSCI instances: that have Manager processes running



Oracle GoldenGate Development Administrator
Oracle GoldenGate Director Web
Oracle GoldenGate Director Desktop Client
Oracle GoldenGate Director Adminstrator
Oracle Weblogic Server
Oracle GoldenGate Director Server Domain
Oracle GoldenGate Director Server
Monitor Agent
Oracle GoldenGate Director Database
GGSCI Data Warehouse
GGSCI Active DB 1
GGSCI – ODS
GGSCI Passive DB
GGSCI Active DB 2
GGSCI – Standby

Tutorial: Installing Oracle GoldenGate Director on Windows

Reference: Oracle GoldenGate Director Administrator's Guide 11g Release 1 (11.1.1)

Following is a list of the used software:

- Oracle GoldenGate Director V11.1.1.1.0 Server on Windows
- Oracle GoldenGate Director V11.1.1.1.0 Client on Windows
- Oracle GoldenGate Monitor V11.1.1.1.0 on Windows

They can be used from Oracle site. At time of this writing, it can be downloaded from the following link.

The tutorial was tested on a Windows XP 32-bit system of 1 GB RAM.

The installation procedure is as follows:

1. Confirm Java JRE version 1.6 is installed in the system. You can check it from Control Panel. If not, install it. Remember, Java JRE versions later than 1.6 is not supported. This tutorial was tested using "jre-6u34".

   Note: Even if you do not have Java JRE version 1.6 installed and even if a JRE of a later version is installed, download and install JRE version 1.6. At time of this writing, it can be downloaded from Oracle site on this link.

2. If not installed, install **Oracle Weblogic Server 11g** (10.3.1, 10.3.2, 10.3.3) Standard Edition

   2.1 Make sure JDK is installed. If not, download and install it. For this tutorial, JDK 7u7 was used.

   2.2 Download Oracle WebLogic Server software (version 11g R1 10.3.3 used for this tutorial) from Oracle site, on this link. When you are given multiple file options to download for the software, select "Oracle WebLogic Server 11gR1 (10.3.3)+Coherence+OEPE-Package Installer"

   2.3 Run the software installer **Next>**

   2.4 In this tutorial, the software was installed on the folder: C:\Oracle\Middleware **Next>**

   2.5 Choose install Type: **Typical> Next>**

   2.6 Select the installation folder for the software components. The WebLogic server component was installed on "C:\oracle\Middleware\wlserver_10.3"

   2.7 Choose Shortcut Location> All Users Start Menu folder Next> Next> Done>

   Start> Programs> Oracle WebLogic> WebLogic Server 11gR1> Tools> Configuration Wizard

   2.8 Create a new WebLogic domain next> Generate a domain configured automatically to ..> **Next>**

   Enter the name and location for the domain

   Domain name: base_domain

   Domain location: C:\domains\wldomain

   2.9 Configure Administrator User Name and Password **Next>**

   2.10 Configure Server Start Mode and JDK **Next>**

   2.11 Select Production Mode and JRockit **Next> Create>**

3. In the repository database, create the username that will be used to store GoldenGate Director data. The password must contain at least eight alphanumeric characters, with at least one alphabet character and one number.:

```
create user ggdir identified by ***
default tablespace ggs_data;
grant connect, dba to ggdir;
```

4. Install "Oracle GoldenGate Director Server":

   4.1 Extract the software zip file of into a directory

   4.2 Check the Release Notes document and the README file shipped with the software files

   4.3 Run the setup executable file from the installation directory >Next

   4.4 Enter the GoldenGate Director directory like: C:\orgg-director >Next

   4.5 Enter the WebLogic directory: "C:\oracle\Middleware\" (not "C:\oracle\Middleware\wlserver_10.3") >Next

   4.6 Enter HTTP Port 7001 Next>

   4.7 Select a Database: select Oracle Next>

   4.8 Enter Database Driver Configuration

   4.9 Enter Database User: ggdir Next>

   4.10 Install as a service Next> Next> Finish

   4.11 Start the Oracle GoldenGate Director Server:

   Start> Programs> Oracle GoldneGate Director Server> Start Oracle GoldenGate Director

   Test it on Web browser: http:<hostname>/7001 (do not use localhost)

5. Install "Oracle GoldenGate Director Client":

   5.1 Extract the software zip file of into a directory

   5.2 Check the Release Notes document and the README file shipped with the software files

   5.3 Follow the simple instructions

Tutorial: Adding a Data Source in Oracle GoldenGate Director

Note: This tutorial assumes that Oracle Director Server and Client software were installed. It also assumes that there are source and target GoldenGate instances ready to manage.

Note: In Director terms, Data Source is any GoldenGate instance accessed by the Director.

1. Start the Admin Tool and login into it as Admin:

   Start> Oracle GoldenGate Director Client> Oracle GDSC Admin Tool

   Enter the following:

   Username: admin

   Password: admin

   Server: <hostname>:7001

2. Add a GoldenGate data source.

   2.1 Start the Manager for the GoldenGate instance that you want to add as a data source.

   2.2 In Oracle GDSC Admin Tool, Click the **Data Sources** tab

   At the bottom of the Data Sources tab, click **New/Clear** button

   Fully Qualified Domain Name: WIN11SRC

   Manager Port: 7840

   Click **Check Connection** to make certain a connection can be made

   2.3 **Data Source Name**: A name for the GoldenGate instance: WIN11SRC

   This name appears as the data source name in the Director Client interface.

   Note: Once a GoldenGate instance has been added as a data source, the fully qualified domain name and port number cannot be changed without deleting it and then adding it back again. The diagrams that reference it should be recreated.

   2.4 Under GoldenGate Info, type or select the following:

   Host Operating System: WU for Windows or UNIX,
   NSK for NonStop Server,
   IBM for an IBM z/OS system running UNIX

   Database: The type of database against which the Manager is running

   GoldenGate Version: 11.2.1.0 (only first four digits)
   can be obtained from GGSCI prompt

   2.5 Under Default DB Credentials, enter the following:

   DSN: Only needs to be completed for ODBC databases; otherwise leave it blank.

   Username: The name of a default user for connecting to the database.

   Password: The password for the user.

   Under Access Control, select the following:

   Owner: One of the users listed on the Accounts tab that always will have full control of this data source.

   If no owner is specified, then all users can see and access the processes associated with this data source.

Host is **Observable** Non-owners will be able to observe this host, but will not be able to alter anything on the host.

3. Click **Save**

Repeated the procedure to add WIN11TRG instance.

Tutorial: Using Oracle GoldenGate Director Client

Create New Diagram

1. Start> Programs> Oracle GoldenGate Director Client> Oracle GoldenGate-Director
2. In the OGG Director window, **File> Login**

   enter the login you created in the Oracle GDSC Admin Tool
3. Create **New Diagram** button> enter Diagram name
4. From Data Sources tab> drage WINTRG and WINSRC into the diagram area
5. Click on any component> modify its configuration values from the tabs underneath

Setting Up the Initial Load using GG Director Client

1. Select (click) the Data Source in the Diagram (if you don not select the Data Source icon in the diagram, the commands under the menu Actions> New> will be disabled)
2. Choose Actions> Add New> Initial Load Task>
3. In "New Initial Load Task" window, fill in the fields OK>
4. Configure the table mappings. For Extract (Capture) process, define the source tables only. For Replicate (Delivery) process, define the source tables, target tables and (if required) any column mappings.

   4.1 In the Diagram area, click on the Map icon for the Data Source of the source database

   4.2 **Parameters** tab> **Insert...** button> **Mapping Type**: Table> enter **source table name**, like hr.employees >**Insert** button

   4.3 Repeat the step above for all the tables included in the initial load

   4.4 Done button

   4.5 When prompted for Saving Changes, click on **Save** button

   4.6 click on the Map icon for the Data Target of the source database

   4.7 **Parameters** tab> select the table(s) to map> **Change** button> enter target table> add Column Mappings if required

   4.8 Mark Handle Collisions option

   4.9 Done button

   4.10 When prompted for Saving Changes, click on **Save** button
5. Start the Initial Loading: right click the Extract process> Start

Setting Up Basic Replication using GG Director Client

1. Perform the actions required in the database level, like enabling supplement logging. For further details, refer to the steps in the section "Tutorial: Basic Replication between two Oracle Databases"
2. Add trandata for the Extract tables: right-click the data source> View Details> Click the Tran/Redo Log Data tab> click Get Info.
3. Enter the login information in the Database Access section> click Get Info to refresh the table list> Highlight the tables> click the Add Trandata> you have the option to choose the trandata columns. If you want to capture all columns, select all the columns> click OK.

Note: In Oracle database, if you login to the database using a schema that does not have access to DBA dictionary views, you will receive the error: "ORA-00942 table or view does not exist". To resolve it, grant "SELECT ANY DICTIONARY" to the schema.

4. Create the Extract process: Right-click the data source> select **New Capture**. Enter the Extract name EXT1> Change the database access information to your own> **OK** button

Note: If your capture is on Oracle RAC (4 indicates four-node RAC):

```
GGSCI> alter extract ext1, translog, threads 4, begin now
```

5. Create Extract process trail file: right click EXT1 icon in the diagram> select New Trail> Fill in the info> OK button

6. Create Data Pump process: right click Trail icon in the diagram> select New Capture> Fill in the info> OK button

7. Create Remote Trail: right click the Data Pump process (PUM1) icon in the diagram> select New Trail> select the target system (WIN11TRG) Fill in the info> OK button

8. Create the Replicate Process: right click on the Remote Trail file icon (R1)> Create Delivery> Fill in the info> OK button

9. Change the mappings in all the Map icons

   Note: it is a little bit tricky to change the mappings. For mappings of a Delivery Process, mark the Table name you want to map, type the table name in the target side, press Enter (do not click outside the field), then click on Save button (do not click outside the row). Also, do not click on Save before pressing on Enter!

10. Start the processes: Extract, Data Pump and the Replicate

11. Check the status of the replication established

Further Available Operations

Operation	Action to do it
Alter Extract or Replicate RUN Options	click the Extract/Replicat object in the diagram, and choose Controls> Run Options
Changing the Trail-File Size	click the Trail object in the diagram, and choose Controls> Run Options
Generating Definition Files	select the data source object in the diagram> Control> View Details> click the Definitions Files tab> Enter the Database Access information for each table you need to add to the source/target definition files. Select one or more tables in the "Select Table" box, and click "Transfer Definitions" when you're finished. The tables automatically transfer to the target/source location via the Manager process. You may have more than one definition files for each schema on either the source or target; if so, concatenate the files, or append them to the existing sourcedef file. Remember to add the sourcedef/targetdef file to the

	Extract or Replicate parameter file.
Adding Parameters that are unavailable in Director	You have to use Edit Param feature to add these parameters manually. Examples of unavailable parameters or attributes: `TRANSMEMORY, FETCHOPTIONS, COMPRESSDELETES, NOCOMPRESSDELETES.`

Setting Up Alerts using GG Director Client

- You can configure Director to send email notification alerts of GoldenGate events. There are only two types of alerts: Checkpoint Lag and Event Text in the ggserr.log file.
- To configure or edit an alert, in the **Multipurpose** pane, click the **My Alerts** tab.
- From the Select an Alert list, select Add new alert to create an alert or select the name of an alert to edit an existing one
- Fill in the fields which are self-explanatory
- Specify the condition that triggers the alert. Depending on the type of alert, the condition is one of the following:
    - For an event alert, select the Event Type and Event Text (contained in the log file) that triggers an alert
    - For a checkpoint lag alert, enter a lag threshold in the form of HH:MM:SS.
- Configure the email settings:
    - In the recipList field, enter at least one email address. Separate multiple addresses with commas
    - In the msgFrom field, enter an email address to display as the sender
- Not all SMTP server configurations are supported. GoldenGate Director supports basic authentication and TLS.
- A custom email option with which you can specify a custom Java class with which to handle the alert.

Troubleshooting Oracle GoldenGate

Troubleshooting Tools

Command/Tool	Description

`INFO ALL`
`INFO <group>`
`INFO <group>, detail`
	display the status and lag (where relevant) for all processes

`INFO EXTTRAIL *`
`INFO RMTTRAIL`
	retrieve configuration information for a local/remote trail files

`VIEW GGSEVT`	view the Oracle GoldenGate error log (ggserr.log file)
`VIEW REPORT`	view the process report that is generated by Extract or Replicate

Using Trace Commands with Oracle GoldenGate

Note: Tracing tools are best used in coordination with Oracle support.

Parameter	Description	Usage
TLTRACE	It traces DB transaction log activity	add the parameter to the Extract or Replicate parameter file and restart the process
TRACE and TRACE2	allow you to grab all details regarding Extract and Replicate processing	

Process Failures

Symptom	Possible Root Cause/Action Plan

Manager Process failure	Manager process fails when network interruption occurs in the source or target system
	`start mgr`
	`info all`

Extract Process failure	Examine the output of the report:
	`VIEW REPORT <extract name>`
	Check the logged problem and resolve it
	start the process again
	`START EXTRACT <extract name>, DETAIL`

	If the process fails even before it writes to the report file: start the process from the operating system's command shell (not GGSCI): `C:\ggs_src>extract paramfile C:\ggs_src\dirprm\extora.prm` On Linux, look for the generated core dump file in gg home directory

Trail Files Issues

Symptom	Possible Root Cause / Action Plan
Trail files that don't empty	If it is local: check for TCP/IP network latency issues If it is remote: check that Replicate process is running OK. If yes: check if there are large transactions going on. Consider partitioning (parallelizing) using Range function

Trail files that don't roll over to a new trail file after it fills up completely	Check the maximum file size:
	`info exttrail *`
	You can change its size:
	`alter exttrail aa megabytes 20, extract extora`

Trail file aren't purged correctly, and thus causing errors between source and target	Absence of PURGEOLDEXTRACTS in mgr parameter file
	Trail files are referenced by an unused (obsolete) processes. In this case, the Trail files are never purged. Get rid of the inactive process:
	`DBLOGIN USERID <user>, PASSWORD <pw>`
	`DELETE REPLICAT <group>`

Trail Files that Purge Too Soon	multiple Replicates and data pumps reading from the same trail file Using the option USECHECKPOINTS of the Manager parameter PURGEOLDEXTRACTS Also, consider using the parameter options: MIN and MAX (both, not one of them)

Discard File Issues

Symptom	Possible Root Cause / Action Plan
Discard File Not Created	the discard file's location wasn't provided in the parameter file (DISCARDFILE) DISCARDFILE references invalid/privileges file or path
Discard File Is Too Big	You should use the following parameters: DISCARDROLLOVER MAXDISCARDRECS PURGE option for DISCARDFILE EGABYTES option for DISCARDFILE

Oracle GoldenGate Configuration Issues

Symptom	Possible Root Cause / Action Plan
Incorrect Software Versions with Oracle GoldenGate	There a unique GG build based on platform, database, and version (32-bit versus 64-bit) `ggsci -v`
Missing Oracle GoldenGate Trail	INFO EXTRACT <group> or the INFO REPLICAT <group>

<table>
<tr><td>Files</td><td></td></tr>
<tr><td>Oracle GoldenGate Parameter File Configuration Issues</td><td>Obtain the parameter file location:
INFO EXTRACT <group>, DETAIL
If the configuration parameters for the Extract and Replicate are listed in wrong order, then processing fails.
The following parameters are required for an Extract:
EXTRACT <group name>
USERID <ID>, PASSWORD <pw>
RMTHOST <hostname>, MGRPORT <port>
RMTTRAIL <trail name> | EXTTRAIL <trail name> |
RMTFILE <filename> | EXTFILE <filename>
TABLE <source table>;
A Replicate requires the following:
REPLICAT <group name>
SOURCEDEFS <file name> | ASSUMETARGETDEFS
USERID <ID>, PASSWORD <pw>
MAP <source table>, TARGET <target table>;
Use the CHECKPARAMS parameter to verify the syntax in a process's parameter file whenever the process starts. It writes results to the report file and then stops the process. Remember to remove CHECKPARAMS from the parameter file after you fix the parameter file.
Common typing errors:
- The Extract TABLE parameter or Replicate MAP parameter isn’t terminated with a semicolon.
- Commas aren’t followed by a space in parameter files.
- Missing commas, quotes, or parentheses in nested clauses exist, such as a COLMAP.</td></tr>
</table>

Operating System Related Issues

Symptom	Possible Root Cause / Action Plan
Missing system libraries	Check the values of LD_LIBRARY_PATH and PATH in the .profile file
Function stack is low	Use FUNCTIONSTACKSIZE parameter. Test it on a testing environment first!
Error message "Bad parameter:Group name invalid"	the process can’t open the checkpoint file INFO * VIEW PARAMS <group> Make sure group name matches the one in the EXTRACT or REPLICAT parameter file

Network Configuration Issues in Oracle GoldenGate

Reference: document ID 966097.1

Symptom	Possible Root Cause / Action Plan
ERROR: sending message to EXTRACT EATAA (TCP/IP error: Connection reset)	Check the port number used by the Manager `INFO MGR` Make sure the Extract parameter RMTHOST use the same port
Extract returns "No Dynamic ports available"	ports are reserved or used ports are occupied by zombie processes. Kill them! consider reserving ports for GG
Network latency issues	Refer to monitoring section in this document Tools to monitor network: OS utilities, HP OpenView or Tivoli

Oracle Database Issues with GoldenGate

Symptom	Possible Root Cause / Action Plan

Extract Can't Access Oracle Database Archive and Redo Logs
`ERROR OGG-00446 Error 5 (Access is denied.) opening log file`
`C:\..WIN11SRC\REDO01.LOG for sequence nn`
	- the file doesn't exist, restore it
	- the archive logs aren't stored in the default Oracle location. Use the parameter `ALTARCHIVELOGDEST <path name>`
	- when a very long transaction is running, and meanwhile the Extract process was restarted, the process starts searching in log files and this might take hours and the process looks like hanging
	- Check long running open transaction from `V$TRANSACTION`:
	You can set a threshold for long-running Oracle database transactions using the parameter `WARNLOGTRANS`

Data-Pump Errors	PASSTHRU can only be used when both source and target table names and structures are the same, and no filtering operations can be performed on the data. When PASSTHRU is used, the sure the parameters USERID or SOURCEDB are not used
Replicate Hangs on the Target System	Note the trail file name and the checkpoint: INFO REPLICAT Make sure the Replicate is reading data from the same trail file the Extract process group is writing to
Replicate Failures	- checkpoint table is not available Resolutions: - drop and re-create the checkpoint table on the target system for the Replicate - use the convchk utility and then restart the Replicate convchk <replicat group> <schema>.<checkpoint table name> The utility causes the checkpoint file to become the new master for Replicate checkpoints. The Replicate resynchronizes the truncated checkpoint table with data from the checkpoint file
Replicate Fails on Large Transactions	If the target system can't support large transactions, the Replicate process groups will fail with an out-of-memory condition.

	To solve this error, you can set the MAXTRANSOPS parameter to split up the Replicate transactions into smaller transactions or use multiple Replicate groups to process these in parallel.
Incompatible Record Errors with a Replicate Incompatible record in C:\GGS_TRGT\DIRDAT\AA000000, rba 5486 (getting header).	Possible root causes: - This occurs if the data gets corrupted in transit from the source system to the target system - Another cause is a data format that isn't compatible with Replicate processing - If you use the FORMATASCII, FORMATSQL, or FORMATXML parameter for the Extract during the initial load, it will cause the failure to occur. - misconfiguration of trail files like having two Extracts writing to the same trail file Resolution: - use the Logdump utility to investigate error 509 conditions with Oracle and GoldenGate - re-initialize and rebuild the processes

Data-Synchronization Issues

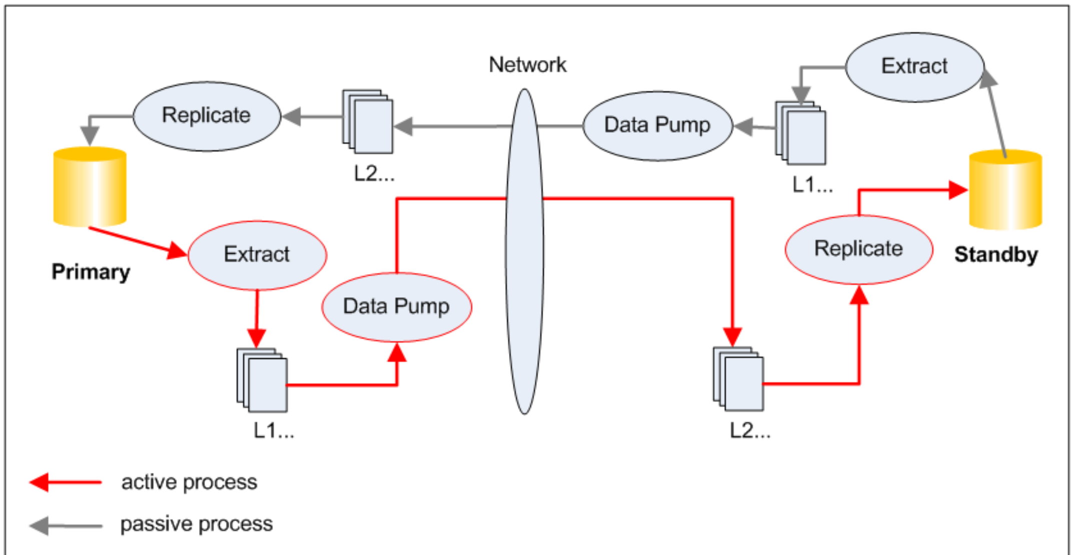
- You should perform out-of-sync data checks on a regular basis.
- Veridata provides a suite of tools to check for data-synchronization

Root Cause	Description / Resolution
Tables Missing Key Constraints	If the table has no pk, two rows may be affected by an update. In this case, you can set the KEYCOLS option for the TABLE and MAP statements in the Extract and Replicate parameter files along with the LIMITROWS.
Character Set Configuration Issues	The source NLS_LANG value must be set to the same exact character set as the target system. In addition, the target Oracle database character set needs to be set as a superset of the source Oracle database system.
Fetch Failures	The Extract is returned "Snapshot Too Old" error Resolutions: - add the FETCHOPTIONS NOUSESNAPSHOT parameter which makes the Extract fetch data from the table instead of going to the undo tablespace. - after you verify that Oracle GoldenGate has processed the record, remove FETCHOPTIONS so the Extract returns to fetching data from the undo tablespace - increase UNDO_RETENTION in source and target

Disaster Recovery Replication using GoldenGate

Considerations

- The best advantage of using GoldenGate as a replication solution is that it works across different databases and platforms.
- GoldenGate can handle both planned and unplanned outage scenarios.
- In replication, the source database is referred to as the "**primary** database" and the target database as the "**standby** or disaster recovery database".
- After failover (unplanned primary failure) or switchover (planned primary downtime) to the standby database, the database roles are reversed. As needed, the database roles can be reversed back to their original configuration at a later time.
- When used for replication, you should have the following:
  - A clear understanding of the business and technical replication requirements for the disaster recovery
  - Backups of your GoldenGate software files and working directories on the source and target servers
  - Processes and scripts to move your application and users from the source to the target server and back again
- GoldenGate replication is configured in a similar way as in the bi-directional replication. The main difference is, in normal operation, only the processes included in one-way direction are active and running. The processes included in the opposite direction are there but passive and doing nothing. The roles are reversed in case of switchover or failover.



Standby replication using GoldenGate in normal operation

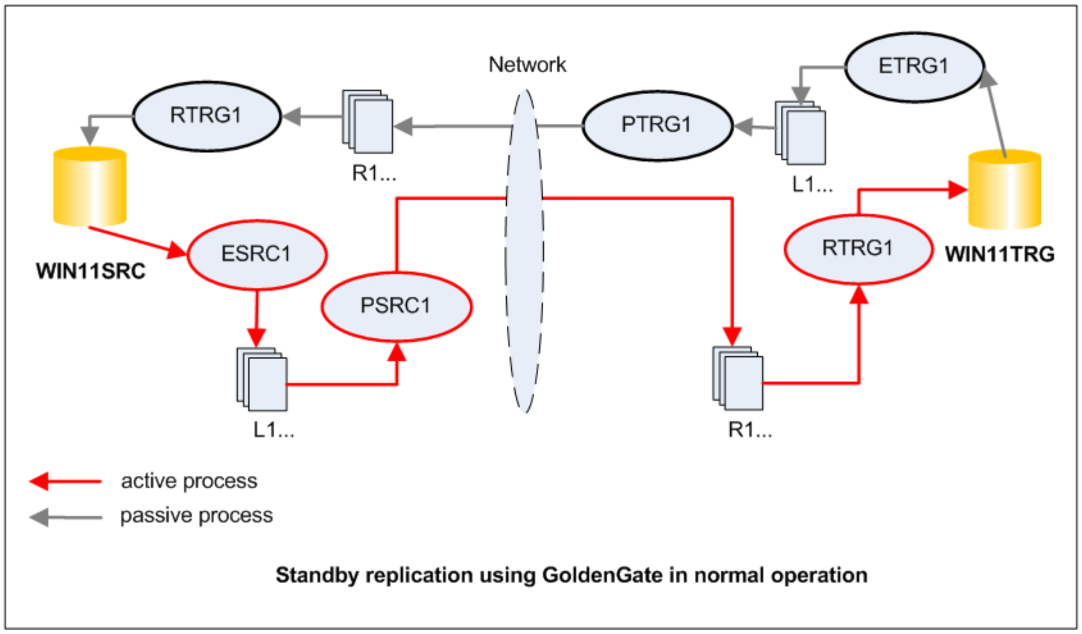
Tutorial: Setting Up Disaster Recovery Replication

This tutorial shows how to configure GoldenGate to set up a disaster replication between two Oracle databases. This configuration will replicate the tables in HR schema.

The environment used for this tutorial was configured as follows:

	Environment 1	**Environment 2**
Hostname	WIN11SRC	WIN11TRG
Database Software	Oracle Database 11g R2	Oracle Database 11g R2
OS	Windows XP 32-bit	Windows XP 32-bit
Extract Capture Mode	classic	
Replicated Tables	HR.COUNTRIES HR.DEPARTMENTS HR.EMPLOYEES HR.JOB_HISTORY HR.JOBS HR.LOCATIONS HR.REGIONS	HR.COUNTRIES HR.DEPARTMENTS HR.EMPLOYEES HR.JOB_HISTORY HR.JOBS HR.LOCATIONS HR.REGIONS
Database Role	Primary (initialization source)	Standby
Conflict Resolution	based on DML_TIMESTAMP column	based on DML_TIMESTAMP column

The processes are configured as follows:



Standby replication using GoldenGate in normal operation

This tutorial has the following parts:

Part 1: Setting up the replication

Part 2: Performing a Planned Switchover

Part 3: Performing an Unplanned Failover

Part 1: Setting up the replication

1. Install Oracle GoldenGate. Refer to its section in this document.
2. Prepare both the source and target databases for GoldenGate replication. Refer to the section "Preparing Oracle Database for GoldenGate".
3. Review the section "Notes on Tables to Replicate" and check its influence on your case.
4. Carry out the following:

```
/* in the BOTH databases: */

-- Make sure all the tables to replicate have key constraint
select t.table_name from dba_tables t where t.owner='HR' and
t.table_name in
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','LOCATIONS','REGIO
NS')
minus
select c.table_name from dba_constraints c where c.owner='HR' and
c.constraint_type in ('P','U')


-- the LOGGING option in the table level should be ENABLED
select table_name from dba_tables where owner='HR' and logging='NO'
and table_name in
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','LOCATIONS','REGIO
NS')

-- to fix those with NOLOGGING:
begin
for r in ( select table_name from dba_tables where owner='HR' and logging='NO'
   and table_name in
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','LOCATIONS','REGIO
NS')) loop
 execute immediate 'alter table hr.'|| r.table_name ||' LOGGING';
end loop;
end;



-- enable table level supplemental logging
Note: If you are using the Oracle GoldenGate DDL replication feature, use the
ADD SCHEMATRANDATA command to log the required supplemental data instead of
ADD TRANDATA command. You can use ADD TRANDATA even when using ADD
SCHEMATRANDATA if you need to use the COLS option to log any non-key columns,
such as those needed for FILTER statements and KEYCOLS clauses in the TABLE
and MAP parameters.

-- to generate its code:
select 'add trandata HR.'||t.table_name  || ' COLS(DML_TIMESTAMP)' AS
SCRIPT_COMMAND
from dba_tables t where t.owner='HR' and
```

```
t.table_name in
('COUNTRIES','DEPARTMENTS','EMPLOYEES','JOB_HISTORY','JOBS','LOCATIONS','REGIO
NS')

-- save the output in a file, name it something like: hr_add_trandata.obey
-- to run the script in gg:
GGSCI > dblogin userid ggs_owner password g
GGSCI > obey diroby\hr_add_trandata.obey



-- in gg: to confirm the table logging is enabled:
dblogin userid ggs_owner password g
info trandata hr.*
```

5. In both databases, depending on the code in the triggers, you may need to modify the triggers to ignore transactions that are applied by Replicate. In the standby database, you may consider disabling the triggers.

   Note: you will need to enable those triggers in switchover or failover situations. Therefore, save the code of enabling them in a script file. This file will be used during switchover or failover procedure.

   Alternatively, for Oracle versions 10.2.0.5, 11.2.0.2 and later (11gR1 not included), you can use the DBOPTIONS parameter with the SUPPRESSTRIGGERS option to disable the triggers for the Replicate session. Refer to GoldenGate Reference documentation for details.

```
-- the following trigger prevents changing data on specific timings in the day
--  this doesn't apply to gg sessions. The code thus changed to the following:

CREATE OR REPLACE TRIGGER secure_employees
  BEFORE INSERT OR UPDATE OR DELETE ON employees
DISABLE
BEGIN
 IF SYS_CONTEXT ('USERENV', 'CURRENT_USER') != 'GGS_OWNER' THEN
  secure_dml;
 END IF;
END secure_employees;
/
```

6. On both databases, handle ON DELETE CASCADE constraints, if they exist.

   Do as what stated in Oracle documentation: "Disable ON DELETE CASCADE constraints and use a trigger on the parent table to perform the required delete(s) to the child tables. Create it as a BEFORE trigger so that the child tables are deleted before the delete operation is performed on the parent table. This reverses the logical order of a cascaded delete but is necessary so that the operations are replicated in the correct order to prevent "table not found" errors on the target."

```
select * from dba_constraints where owner='HR' and DELETE_RULE ='DELETE_RULE';
```

7. Make sure the steps of Installing Support Sequences have been implemented.

8. On both systems: create a Replicate checkpoint table.

```
-- on source:
dblogin userid ggs_owner@win11src password g
ADD CHECKPOINTTABLE HR.GG_CHKPT
```

```
-- on target:
dblogin userid ggs_owner@win11trg password g
ADD CHECKPOINTTABLE HR.GG_CHKPT
```

9. On both systems, configure the Manager process:

```
/* on Primary */
edit params mgr

port 7840
-- autostart Extract
autostart E *
BOOTDELAYMINUTES 4


/* on Standby */
edit params mgr

port 7840
-- autostart Replicate
autostart R *
BOOTDELAYMINUTES 4


-- delete old trail files (optional)
PURGEOLDEXTRACTS dirdat/*, USECHECKPOINTS, MINKEEPDAYS 3
```

10. In Source: configure the local Extract and the Data Pump.

```
1. verify the manager is running
GGSCI (WIN11SRC) 2> info manager
Manager is running (IP port WIN11SRC.7840)


2. create and edit the Extract parameter file (8 characters max):
GGSCI (WIN11SRC) 2>edit params ESRC1

-- add the following to it:

Extract ESRC1
SETENV (NLS_LANG = AMERICAN_AMERICA.WE8MSWIN1252)
-- this is needed, if you have more than one instance:
SETENV (ORACLE_HOME = "E:\oracle\product\11.2.0\dbhome_1")
SETENV (ORACLE_SID = "ORADB")

-- handling process report parameters (optional)
REPORT AT 8:00
REPORTCOUNT EVERY 60 MINUTES, RATE
REPORTROLLOVER AT 08:00 ON MONDAY

-- discard file PURGE or APPEND (optional)
DISCARDFILE discard.dsc, PURGE, MEGABYTES 200
DiscardRollover at 02:00 ON SUNDAY

USERID GGS_OWNER@WIN11SRC, PASSWORD g
ExtTrail dirdat/l1
```

```
Table HR.REGIONS;
Table HR.LOCATIONS;
Table HR.JOB_HISTORY;
Table HR.JOBS;
Table HR.EMPLOYEES;
Table HR.DEPARTMENTS;
Table HR.COUNTRIES;



3. add a primary Extract group:
ADD EXTRACT ESRC1, TRANLOG, BEGIN NOW
ADD EXTTRAIL dirdat/l1, EXTRACT ESRC1, MEGABYTES 100
START EXTRACT ESRC1

-- verify the extract:
info all
info extract ESRC1, DETAIL



4. create the Data Pump parameter file:
-- no env variables should be defined
edit params PSRC1

Extract PSRC1
PassThru

ReportCount Every 10000 Records, Rate
Report at 00:30

DiscardFile dirrpt/PSRC1.dsc, Append
DiscardRollover at 02:00

RmtHost WIN11TRG, MgrPort 7840
RmtTrail dirdat/r1

Table HR.* ;



5. create the Data Pump process:
ADD EXTRACT PSRC1, EXTTRAILSOURCE dirdat/l1
ADD RMTTRAIL dirdat/r1, EXTRACT PSRC1, MEGABYTES 100
START EXTRACT PSRC1

-- verify the process:
info extract PSRC1
info extract PSRC1, detail
```

11. On the Target: create the Replicate group.

```
1. create the parameter file:
GGSCI (WIN11TRG) 1> edit params RTRG1

Replicat RTRG1
SETENV (NLS_LANG = AMERICAN_AMERICA.WE8MSWIN1252)
```

```
-- Specifies the discard file.
DISCARDFILE ./dirrpt/rtrg1_discards.dsc, APPEND, MEGABYTES 100
DiscardRollover at 02:00 ON SUNDAY

USERID GGS_OWNER@WIN11TRG, PASSWORD g

-- the source and target are identical
ASSUMETARGETDEFS

-- Specify error handling rules:
-- writes operations that could not be processed to a discard file
REPERROR (DEFAULT, DISCARD)

Map HR.*, Target HR.* ;



2. add the Replicate
ADD REPLICAT RTRG1, EXTTRAIL dirdat/r1, CHECKPOINTTABLE HR.GG_CHKPT
START REPLICAT rtrg1



3. verify the Replicate is running
INFO REPLICAT rtrg1
INFO REPLICAT rtrg1, DETAIL
```

12. In Secondary: configure the local Extract and the Data Pump.

```
1. create and edit the Extract parameter file (8 characters max):
GGSCI (WIN11TRG)> edit params ETRG1

Extract ETRG1
SETENV (NLS_LANG = AMERICAN_AMERICA.WE8MSWIN1252)
SETENV (ORACLE_HOME = "E:\oracle\product\11.2.0\dbhome_1")
SETENV (ORACLE_SID = "ORADB")

-- handling process report parameters(optional)
REPORT AT 8:00
REPORTCOUNT EVERY 60 MINUTES, RATE
REPORTROLLOVER AT 08:00 ON MONDAY

-- discard file PURGE or APPEND (optional)
DISCARDFILE discard.dsc, PURGE, MEGABYTES 200
DiscardRollover at 02:00 ON SUNDAY

-- Exclude Replicate trans (Oracle)
TRANLOGOPTIONS EXCLUDEUSER ggs_owner COMPLETEARCHIVEDLOGTIMEOUT 600

USERID GGS_OWNER@WIN11TRG, PASSWORD g

ExtTrail dirdat/L1

Table HR.REGIONS;
Table HR.LOCATIONS;
Table HR.JOB_HISTORY;
Table HR.JOBS;
Table HR.EMPLOYEES;
Table HR.DEPARTMENTS;
Table HR.COUNTRIES;



3. add a primary Extract group (without starting it):
ADD EXTRACT ETRG1, TRANLOG, BEGIN NOW
ADD EXTTRAIL dirdat/L1, EXTRACT ETRG1, MEGABYTES 100

/* verify the extract:  */
info extract ETRG1
info extract ETRG1, DETAIL



4. create the Data Pump parameter file:
-- no env variables should be defined
edit params PTRG1

Extract PTRG1
PassThru

ReportCount Every 10000 Records, Rate
Report at 00:30

DiscardFile dirrpt/PTRG1.dsc, Append
DiscardRollover at 02:00

RmtHost WIN11SRC, MgrPort 7840

RmtTrail dirdat/R1

Table HR.* ;



5. create the Data Pump process:
ADD EXTRACT PTRG1, EXTTRAILSOURCE dirdat/l1
ADD RMTTRAIL dirdat/R1, EXTRACT PTRG1, MEGABYTES 100

-- verify the process:
info extract PTRG1, DETAIL
```

13. On the Primary, create the Replicate group.

```
1. create the parameter file :
Note: it can be copy of the one in the Secondary but the USERID must be
modified to connect to the source
GGSCI (WIN11SRC) > edit params RSRC1

Replicat RSRC1
SETENV (NLS_LANG = AMERICAN_AMERICA.WE8MSWIN1252)

-- Specifies the discard file.
DISCARDFILE ./dirrpt/rsrc1_discards.dsc, APPEND, MEGABYTES 100
DiscardRollover at 02:00 ON SUNDAY

USERID GGS_OWNER@WIN11SRC, PASSWORD g
```

```
ASSUMETARGETDEFS

-- Specify error handling rules:
-- writes operations that could not be processed to a discard file
REPERROR (DEFAULT, DISCARD)

Map HR.*, Target HR.* ;



2. add the Replicate (do not start it)
ADD REPLICAT RSRC1, EXTTRAIL dirdat/R1, CHECKPOINTTABLE HR.GG_CHKPT


3. verify the Replicate is running
INFO REPLICAT RSRC1, DETAIL
```

14. Test the one way replication from the primary to the standby database.

Part 2: Performing a Planned Switchover

1. Stop the application connected to the primary database
2. Verify there is no lag in the replicated data
3. After making sure each process has no lag, stop the local Extract, Data Pump, and the remote Replicate processes:

```
-- in Primary
LAG EXTRACT esrc1
-- make sure no records to process
STOP EXTRACT esrc1

LAG EXTRACT psrc1
-- make sure no records to process
STOP EXTRACT psrc1

-- in Standby
LAG EXTRACT rtrg1
STOP REP rtrg1
```

4. In the standby system, start the Extract processes:

```
ALTER EXTRACT etrg1, BEGIN NOW
START EXTRACT etrg1
INFO etrg1
```

5. Prepare the standby database for the application:

   5.1 Enable any triggers or cascade delete constraints you may have stopped while the standby was the replication target.

6. Start the application connected to the standby database
7. Start the required maintenance operation on the primary database system
8. When the maintenance is concluded, disable any triggers or cascade delete constraints on the primary (just as you did on the Standby in the initial set up)
9. Stop the application running against the standby database

10. Start the Replicate on the primary database:

```
START REPLICAT rsrc1
info rsrc1
```

11. Start the data-pump Extract on the standby:

```
START EXT ptrg1
info ptrg1
```

12. After verifying there is no lag, stop the following processes:

```
-- on the standby server:
LAG EXTRACT etrg1
STOP EXTRACT etrg1

LAG EXTRACT ptrg1
STOP EXTRACT ptrg1


-- on the primary database:
LAG EXTRACT rsrc1
STOP REP rsrc1
```

13. Alter the Local Extract for the primary database to begin capturing changes beginning now and then start the Extract:

```
ALTER EXTRACT esrc1, BEGIN NOW
START EXTRACT esrc1
info esrc1
```

14. Prepare the primary database for the application:

    14.1 Enable any triggers or cascade delete constraints you may have stopped while the standby was the replication target (just as you did on the standby server when it worked as a primary)

15. Start the applications against the primary database

16. Start the Data-pump Extract on the primary database:

```
START EXTRACT psrc1
info psrc1
```

17. Start the Replicate on the standby database as shown in the following example:

```
START REPLICAT rtrg1
info rtrg1
```

Part 3: Performing an Unplanned Switchover

The scenario here assumes the primary server fails with no recovery window. The applications will be switched to the standby server and the primary database will be restored in a new machine.

1. After verifying there is no lag, stop the Replicate:

```
LAG REPLICAT rtrg1
STOP REPLICAT rtrg1
```

2. In the standby system, start the Extract processes:

```
ALTER EXTRACT etrg1, BEGIN NOW
START EXTRACT etrg1
INFO etrg1
```

3. Prepare the standby database for the application:

   3.1 Enable any triggers or cascade delete constraints you may have stopped while the standby was the replication target.

4. Start the application connected to the standby database

5. After the primary server is built, restore the GoldenGate software directories and files from your offline server backups.

6. Delete the Local Extract and the source trails and then add the Local Extract back:

```
DELETE EXTRACT ESRC1
-- this removes the GoldenGate checkpoints:
DELETE EXTTRAIL dirdat/l1

-- delete the dirdat/l1 files

-- start the Manager
start mgr

-- add the process back again:
ADD EXTRACT ESRC1, TRANLOG, BEGIN NOW
ADD EXTTRAIL dirdat/l1, EXTRACT ESRC1, MEGABYTES 100

-- Do NOT start the Extract process for now
```

7. Disable any triggers or cascade delete constraints on the primary database.

8. Perform the initial data load from the standby to the primary database. Using GoldenGate, this step can be done in "hot" mode. Check the example for the steps to do it.

9. Stop the application running against the standby database

10. Make sure the Extract is up and running in the standby server:

```
info etrg1
```

11. In the primary database, start the Replicate:

```
start rep rsrc1
```

12. In the standby server, start the Data Pump process:

```
start ptrg1
```

13. Once the replication is complete and the lag is showing "At EOF, no more records to process", you can stop the Extracts and Replicates:

```
lag replicat rsrc1
...
stop rsrc1

-- on the standby:
stop etrg1
stop ptrg1
```

14. Verify the data is valid and synchronized. You can use SQL queries or GoldenGate Veridata.

15. In the primary database, start the Extract:

```
ALTER EXTRACT LHREMD2, BEGIN NOW
START EXTRACT LHREMD2
```

16. Enable any triggers or cascade delete constraints you may have stopped while the primary database was the replication target.

17. Connect the applications to primary database

18. In the primary server, start the Data Pump process:

```
start psrc1
info psrc1
```

19. In the standby server, start the Replicate process:

```
start rtrg1
info rtrg1
```

Zero-Downtime Migration Replication

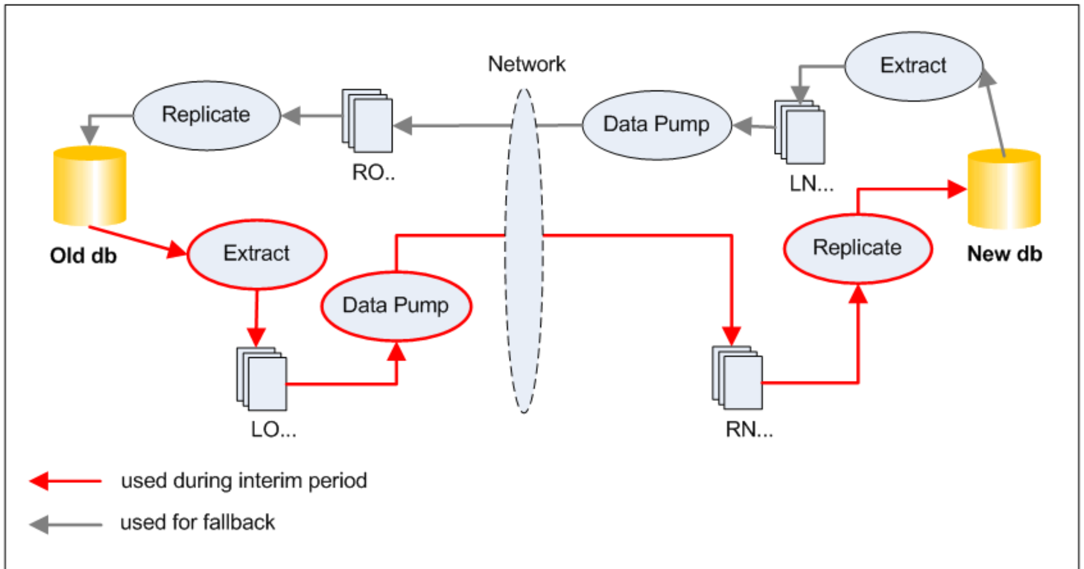
- When you need to migrate a production RDBMS to another new one (regardless of RDBMS product or platform), GoldenGate can help on making the migration with almost no downtime.
- Your plan should cover how to handle both *cutover* to the new database and *fallback* from the new database to the old database if there are problems.

Prerequisites

- The Oracle GoldenGate software installed on the new and old
- The Manager process up and running on both servers
- An understanding of the business and technical replication requirements of the planned migration
- Backups of your Oracle GoldenGate software and working directories on both servers

Considerations

- The old database keeps synchronized with the new one. After cutover, the new database is still kept synchronized with the old one till it is certain no fallback is required.
- Data structure could be different.
- The applications may need reconfiguration to connect to the new database
- The time period permitted for the cutover and fallback procedures should be well understood.



GoldenGate configuration for zero-time migration

Zero-Downtime Migration High Level Plan

Setup Stage: configure active one-way replication from old db to new db

1. In the old database, configure the local Extract and the Data Pump Extract processes and start them
2. Initial load the data from old database to the new database
3. In the new database, configure the Replicate process and start it. Initially keep it working using the parameter HANDLECOLLISIONS. When it catches up with the changes made on the data after the initial load was finished, stop it and restart it using the parameter NOHANDLECOLLISIONS.
4. Verify the data in the old and new databases are valid and synchronized

Fallback Preparation: configure passive one-way replication from new db to old db

5. In the new database, add the local Extract and the Data Pump Extract processes, without starting them
6. In the old database, add the Replicate process, without starting it. Make sure HANDLECOLLISIONS is **not** used in its parameter file

Migration Cutover: switch your applications to the new db

7. Stop the applications connected to the old db
8. After making sure there is no lag, stop the one-way replication from the old db to the new db.
9. Start the local Extract process in the new db, as in the following example:

```
ALTER EXTRACT enewdb, BEGIN NOW
START EXTRACT enewdb
```

10. Prepare the new db: enable any triggers or cascade-delete constraints you may have stopped and configure the applications to connect to the new db
11. Start the applications
12. In the new db server, start the Data Pump process
13. Prepare the old db: disable any triggers or cascade-delete constraints
14. In the old db server, start the Replicate process

Migration Fallback: if issues raised that require fallback to the old db

15. Stop the applications connected to the new db
16. In the new db, after verifying that there is no lag, stop the Local Extract and Data Pump processes
17. In the old db server, after verifying that there is no lag, stop the Replicate process
18. In the old db server, start the local Extract process, as in the following example:

```
ALTER EXTRACT eolddb, BEGIN NOW
START EXTRACT eolddb
```

19. Prepare the old db: enable any triggers or cascade-delete constraints you may have stopped

20. Start the applications against the old db

21. In the old db server, start the Data Pump process

22. Prepare the new db: disable any triggers or cascade-delete constraints

23. In the new db server, start the Replicate process

24. Perform the cutover again, when the issues are resolved

Using Oracle GoldenGate Guidelines

Understanding the Requirements

- **Source and target configurations:**
  - Describe the existing or planned source and target databases.
  - What are the software releases for the database and operating system?
  - Are the servers stand-alone or clustered?
  - How is the data storage configured?
  - What are the character sets for the source and target databases?

- **Data currency:**
  - How quickly and how often does the data need to be updated?
  - Can any lag in the data replication be tolerated? If no lag can be tolerated and there is a high volume of changes, you may need to devote more time in the project to designing and tuning the replication to avoid any lag.

- **Data volume:**
  - How much data needs to be replicated?
  - How often is it updated? You may be able to examine the existing database-transaction log files to determine the amount of data change that is occurring in the source database. The data-change volume impacts the network bandwidth requirements between the source and target and the amount of disk space required for trail files.

- **Data requirements**:
  - You should understand any unique or special data requirements so they can be handled properly in GoldenGate. For example, if there are triggers, they may need to be disabled in the target database.
  - Sequences need special parameters added if they need to be replicated.
  - Certain data types may need special handling or may not be supported.

- **Security requirements**:
  - How much security is required? For example, is it necessary to encrypt the data GoldenGate is replicating? GoldenGate can encrypt both the data stored in trail files as well as the data being sent from the source to the target.
  - Also, is the data in the database encrypted? For example, Oracle can encrypt database data using a feature called Transparent Data Encryption (TDE). If the database is using TDE, GoldenGate needs special parameters to unencrypt the data for replication.
  - GoldenGate itself requires security to install and own the GoldenGate software and execute the replication inside the database.
- **Network requirements**:
  - Describe the network between the source and the target.
  - Are there any firewalls that need to be opened for the replication?
  - What is the distance?
  - How much bandwidth is required for the replication, and how much is available?

Installation and Setup

- Create dedicated operating-system and database users for GoldenGate
- Always encrypt your passwords in the parameter files
- Use a checkpoint table in the target database. Consider adding CHECKPOINTTABLE parameter in the GLOBALS file
- Set the database character set inside the parameter file using SETENV
- It is always highly recommended to configure a Data Pump process

Management and Monitoring

- Save your commands in obey files and store them in /diroby
- Include the REPORT parameter with REPORTCOUNT and RATE options to generate interim runtime statistics
- Always use DiscardFile and DiscardRollover parameters
- Consider using the Manager parameters DOWNREPORT, DOWNCRITICAL, LAGCRITICAL and LAGINFO
- Consider using the parameter PurgeOldExtracts in the Manager level
- Consider using the Manager parameter AUTOSTART. Be aware that in some GoldenGate configuration you have inactive Extract or Replicate processes that you do not want to automatically start when the Manager starts.

Tuning Performance

- The best method for performance testing is to test the normal workload against a testing environment that is identical or so close to the production environment
- You do not need to parallelize the Extract process. Only Replicate may get advantage from it.
- Tune transactions applied by the Replicate on the target database.
- Whenever possible, use the PASSTHRU parameter with the Pump process
- Store the trail files on a dedicated disk in separate mount points

Best Practices and References

My Oracle Support (MOS) White Papers

- Database specific bundle patch for Integrated Extract 11.2.x (Doc ID 1411356.1)
- Oracle GoldenGate Best Practices: Instantiation from an Oracle Source Database (Doc ID 1276058.1)
- Oracle GoldenGate Best Practice: NFS Mount options for use with GoldenGate (Doc ID 1232303.1)
- Oracle GoldenGate Best Practices: Oracle GoldenGate high availability using Oracle Clusterware (Doc ID 1313703.1)
- Oracle GoldenGate Best Practice - sample parameter files (Doc ID 1321696.1)