Golden Gate Lab guide

Day 1. Golden gate Installation and Configuration

Day 2. Configuring Data Replication using Golden Gate

Day 3. Trail files, Initial load, Data Selection and Transformation

Day 4. Bi-directional Replication, DDL Replication, Encryption and Compression

Day 1. Golden Gate Installation

1.1 Objectives of the lab

- To learn golden gate installation.
- To learn golden gate configuration.
- To learn database configuration for golden gate.
- To get familiar with golden gate command prompt (ggsci).

1.2 Resources you must have before you start the labs:

a.) Atleast 1.5 GB utilizable RAM and 35 GB free disk space.
b.) Oracle VirtualBox or VMware installed on your machine.
c.) Oracle grid infrastructure installed in standalone mode for using ASM.
d.) Oracle database EE software installed.
e.) Two oracle database instances running **(orcl and dev)**
f.) Database **orcl** will use ASM for keeping its datafiles and database **dev** will use linux mount points to keep its datafiles.
g.) ORCL is the source and DEV will be used as target.

1.3 Pre-installation tasks:

1. Tell database to log more (supplemental logging ~10% of more redo writing) on source side.

   alter database add supplemental log data;

   alter system switch logfile;

2. GoldenGate user creation on source and target side. (Optional)

   useradd -d /data/home/gguser gguser

3. Create tablespace for golden gate user on both source and target.

   Create tablespace OGGDATA datafile size 50M autoextend on;

4. GoldenGate schema creation into source and target database

create user OGG identified by ogg default tablespace OGGDATA temporary tablespace TEMP profile DEFAULT;
alter user OGG QUOTA UNLIMITED ON OGGDATA;
grant CONNECT to OGG;
grant CREATE SESSION to OGG;
grant RESOURCE to OGG;
grant SELECT ANY TABLE to OGG;
grant ALTER SESSION to OGG;
grant CREATE TABLE to OGG;
grant FLASHBACK ANY TABLE to OGG;
grant SELECT ANY DICTIONARY to OGG;
grant DBA to OGG; (just for this lab purpose, we are giving the OGG user the DBA role. Else in real world scenario you would give selective privileges over the objects which you want to replicate using oracle Golden Gate)

1. Run package to grant Oracle Streams admin privilege.

exec dbms_streams_auth.grant_admin_privilege('OGG');

2. Grant INSERT into logmnr_restart_ckpt$.

grant insert on system.logmnr_restart_ckpt$ to ogg;

3. Grant UPDATE on streams$_capture_process.

grant update on sys.streams$_capture_process to ogg;

4. Grant the 'become user' privilege.

grant become user to ogg;

1.4 Golden Gate Installation

1. Create a directory for golden gate home. The directory should have permissions given to the oracle golden gate OS user and also, the Golden gate OS user must have read privileges over the redo logs and archivelogs of the database from where it wishes to get its data.

   On source:

   mkdir gg_orcl

   On target:

   mkdir gg_dev

2. Extract the golden gate software tar file into the golden gate home directories made in step 1.

   Use the following commands to unpack the golden gate bundle

   unzip <golden gate download>

   tar –xvf <golden gate download>

3. Set environmental variables

   On source:

   export ORACLE_HOME=/oracle/product/dbhome

   export ORACLE_SID=orcl

   export LD_LIBRARY_PATH=/oracle/product/dbhome /lib

   export PATH=$PATH:$ORACLE_HOME/bin

   On target:

   export ORACLE_HOME=/oracle/product/dbhome

   export ORACLE_SID=dev

   export LD_LIBRARY_PATH=/oracle/product/dbhome /lib

   export PATH=$PATH:$ORACLE_HOME/bin

1.5 Golden Gate configuration

1. Verify if the main GG tool is working or not. By changing the directory to Golden gate installation directory and then giving the command

   ./ggsci

   If the above command logs you into golden gate secured command prompt, it implies your installation was successful.

2. Configuring the source side

Create the sub-directories in the golden gate home by the following command.

GGSCI> create subdirs

View status of all golden gate processes using the following command.

GGSCI> status all

Edit the manager process's parameters by the following command.

GGSCI> edit params mgr

You may use the following values for golden gate manager process parameters

port 7809
lagreportminutes 5
laginfominutes 1
lagcriticalminutes 2
purgeoldextracts ./dirdat/a, minkeepdays 2, usecheckpoints*

Start the manager process

GGSCI > start mgr
GGSCI > status all

If manager is running then manager configuration is ok.

GGSCI > dblogin userid OGG, password ogg

If you see the list of tables then your configuration is good and you can continue:

*GGSCI > add trandata DATA.**
*GGSCI > info trandata DATA.**

3. Configure the target side

Create the sub-directories in the golden gate home by the following command.

GGSCI > create subdirs

Edit manager parameters on the target side.

GGSCI > edit params mgr
port 7810
dynamicportlist 7900-7950
lagreportminutes 5
laginfominutes 1
lagcriticalminutes 2
purgeoldextracts ./dirdat/a, minkeepdays 2, usecheckpoints*

Start the manager process

GGSCI > start mgr

GGSCI > status all

View the manager process report

GGSCI > view report mgr

1.6 Introduction to golden gate command interface

1. Log into ggsci

2. View a HELP summary for all commands

```
GGSCI (        )     > Help

GGSCI Command Summary:

Object:           Command:
SUBDIRS           CREATE
ER                INFO, KILL, LAG, SEND, STATUS, START, STATS, STOP
EXTRACT           ADD, ALTER, CLEANUP, DELETE, INFO, KILL,
                  LAG, REGISTER, SEND, START, STATS, STATUS, STOP
                  UNREGISTER
EXTTRAIL          ADD, ALTER, DELETE, INFO
GGSEVT            VIEW
MANAGER           INFO, SEND, START, STOP, STATUS
MARKER            INFO
PARAMS            EDIT, VIEW
REPLICAT          ADD, ALTER, CLEANUP, DELETE, INFO, KILL, LAG, SEND,
                  START, STATS, STATUS, STOP
REPORT            VIEW
RMTTRAIL          ADD, ALTER, DELETE, INFO
TRACETABLE        ADD, DELETE, INFO
TRANDATA          ADD, DELETE, INFO
SCHEMATRANDATA    ADD, DELETE, INFO
CHECKPOINTTABLE   ADD, DELETE, CLEANUP, INFO

Commands without an object:
(Database)        DBLOGIN, LIST TABLES, ENCRYPT PASSWORD, FLUSH SEQUENCE
                  MININGDBLOGIN
(DDL)             DUMPDDL
(Miscellaneous)   FC, HELP, HISTORY, INFO ALL, OBEY, SET EDITOR, SHELL,
                  SHOW, VERSIONS, ! (note: you must type the word
                  COMMAND after the ! to display the ! help topic.)
                  i.e.: GGSCI (sys1)> help ! command

For help on a specific command, type HELP <command> <object>.

Example: HELP ADD REPLICAT

GGSCI (        )     > Help All
```

3. View HELP for specific commands

   GGSCI> HELP ADD EXTRACT

4. View command history

    GGSCI > History

        GGSCI Command History

            1: Help

            2: Help All

            3: Help Add Extract

            4: Help Add ExtTrail

            5: History

5. View brief summary of golden gate processes

    GGSCI > **Info All**

```
Program     Status      Group       Lag at Chkpt  Time Since Chkpt

MANAGER     STOPPED
```

6. Exit from ggsci

Day 2.Configuring Data Replication using Golden Gate

2.1 Create the table structures on both the sides

On both source and the target

```
create user data identified by welcome1;
grant connect,resource,create table to data;
grant all on scott.emp to data;
```

On Source

```
conn data/welcome1
create table employees0 as select * from scott.emp;
create table employees1 as select * from scott.emp;
```

On Target

```
conn data/welcome1
create table empr0 as select * from scott.emp;
create table empr1 as select * from scott.emp;
CREATE table emp_dw (EMPNO NUMBER(4),emp_name VARCHAR2(10),HIREDATE
DATE,salary NUMBER(7,2));
```

2.2 Replication Using One Extract Process

On source Side

```
GGSCI>add trandata data.*
GGSCI>info trandata data.*

GGSCI>edit params xtst00
```

```
extract xtst00
userid OGG, password ogg
discardfile ./dirrpt/xtst00.dsc,purge
reportcount every 15 minutes, rate
rmthost ggdemo, mgrport 7810
rmttrail /oracle/golden_gate/gg_dev/dirdat/a0
tranlogoptions asmuser sys@ASM, asmpassword welcome1
table data.employees0;
```

Add the following entry in tnsnames.ora on the source Database home

```
ASM =
 (DESCRIPTION =
   (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = TCP)(HOST = ggdemo)(PORT = 1521))
   )
   (CONNECT_DATA =
    (SERVICE_NAME = +ASM)
   )
 )
```

Make sure that your ip is configured against your hostname in /etc/hosts file.

```
GGSCI>add extract xtst00, tranlog, begin now
GGSCI>add rmttrail /oracle/golden_gate/gg_dev/dirdat/a0, extract xtst00, megabytes 100
GGSCI>start xtst00
```

On Target Side

```
GGSCI>edit params rtst00

replicat rtst00
userid OGG, password ogg
discardfile ./dirrpt/rtst00.dsc, purge
assumetargetdefs
reportcount every 15 minutes, rate
batchsql
MAP data.employees0, TARGET data.empr0;
```

```
GGSCI> EDIT PARAMS ./GLOBALS
GGSCHEMA OGG
```

GGSCI>DBLOGIN USERID ogg, PASSWORD ogg
GGSCI>add checkpointtable ogg.checkpoint_gg
GGSCI>add replicat rtst00, exttrail ./dirdat/a0,checkpointtable ogg.checkpoint_gg
GGSCI>start rtst00

GGSCI>send replicat rtst00, status

Try the following sample transaction on the source and check the target

SQL>insert into employees0 values (101,'JOHN','MD',00,TO_DATE('04/02/2001', 'DD/MM/YYYY'),50000,1000,1);
SQL>commit;

2.3 Replication using Two Separate Extract Processes for Capture and Data Pump

---

On source Side

GGSCI>add trandata data.*
GGSCI>info trandata data.*

GGSCI>edit params xtst01

extract xtst01
userid OGG, password ogg
discardfile ./dirrpt/xtst01.dsc,purge
reportcount every 15 minutes, rate
tranlogoptions asmuser sys@ASM, asmpassword welcome1
exttrail ./dirdat/a1
table data.employees1;

GGSCI>add extract xtst01, tranlog, begin now
GGSCI>add exttrail ./dirdat/a1, extract xtst01, megabytes 100
GGSCI>start xtst01

GGSCI>edit params ptst01

extract ptst01

passthru
rmthost ggdemo, mgrport 7809
rmttrail /oracle/golden_gate/gg_dev/dirdat/a1
table data.employees1;

GGSCI>add extract ptst01, exttrailsource ./dirdat/a1
GGSCI>add rmttrail /oracle/golden_gate/gg_dev/dirdat/a1, extract ptst01, megabytes 100
GGSCI>start ptst01

On Target Side

GGSCI>edit params rtst01

replicat rtst01
userid OGG, password ogg
discardfile ./dirrpt/rtst01.dsc, purge
assumetargetdefs
reportcount every 15 minutes, rate
batchsql
MAP data.employees1, TARGET data.empr1;

GGSCI> EDIT PARAMS ./GLOBALS
GGSCHEMA OGG

GGSCI>DBLOGIN USERID ogg, PASSWORD ogg
GGSCI>add checkpointtable ogg.checkpoint_gg
GGSCI>add replicat rtst01, exttrail ./dirdat/a1,checkpointtable ogg.checkpoint_gg
GGSCI>start rtst01

GGSCI>send replicat rtst01, status

2.4 Replication Using Defgen for tables of different structures

---
On source Side

GGSCI> edit params defgen
defsfile ./dirdef/defemp1.sql
userid OGG password ogg
table data.employees1;

Run the defgen utility to generate the definition file.

./defgen paramfile ./dirprm/defgen.prm

Copy definition file from source to target

cp /oracle/golden_gate/gg_orcl/dirdef/defemp1.sql to /oracle/golden_gate/gg_dev/dirdef

---
On Target Side

GGSCI>edit params rtst02

replicat rtst02
userid OGG, password ogg
discardfile ./dirrpt/rtst02.dsc, purge
reportcount every 15 minutes, rate
batchsql
SOURCEDEFS ./dirdef/defemp1.sql
MAP data.employees1, TARGET data.emp_dw,
colmap (USEDEFAULTS,emp_name=ename, salary=sal);

GGSCI>DBLOGIN USERID ogg, PASSWORD ogg
GGSCI>add checkpointtable ogg.checkpoint_gg
GGSCI>add replicat rtst02, exttrail ./dirdat/a1,checkpointtable ogg.checkpoint_gg
---
GGSCI>send replicat rtst02, status

Day 3. Trail files, Initial load, Data Selection and Transformation

3.1 Exploring the Trail file using logdump

In Golden gate installation home

$ ./logdump

Oracle GoldenGate Log File Dump Utility for Oracle

Version 11.2.1.0.0 OGGCORE_11.2.1.0.0_PLATFORMS_120131.1910

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

Logdump 1 > ***help***

Logdump 2 > ***open dirdat/a1000000***

Logdump 3 > ***n***

```
2012/05/11 06:44:37.550.418 FileHeader Len 1030 RBA 0
Name: *FileHeader*
3000 01a3 3000 0008 4747 0d0a 544c 0a0d 3100 0002 | 0...0...GG..TL..1...
0003 3200 0004 2000 0000 3300 0008 02f1 e5f0 86ad | ..2... ...3.........
7552 3400 0027 0025 7572 693a 4544 5253 5233 3150 | uR4..'.%uri:EDRSR31P
313a 3a75 3031 3a61 7070 3a6f 7261 636c 653a 6767 | 1::u01:app:oracle:gg
5f61 6d65 7236 0000 1300 112e 2f64 6972 6461 742f | _amer6....../dirdat/
6577 3030 3030 3030 3700 0001 0138 0000 0400 0000 | ew0000007....8......
     0039 ff00 0800 0000 0000 0000 003a 0000 8107 3134 | .9..........:....14
```

Logdump 4 > ***fileheader on***

Logdump 5 > ***pos 0***

Reading forward from RBA 0

Logdump 6 > ***n***

Logdump 8 > ***pos 0***

Reading forward from RBA 0

Logdump 9 > ***ghdr on***

Logdump 10 > ***detail on***

Logdump 11 > ***n***

Logdump 12 > ***n***

3.2 Initial load

3.2.1 Initial load using direct load

On Source

1) **Create the Initial data extract process 'exti1'**

GGSCI> **ADD EXTRACT exti1, SOURCEISTABLE**
EXTRACT added.

2) **Create the parameter file for the extract group exti1**

GGSCI> **EDIT PARAMS exti1**

EXTRACT exti1
USERID ogg, PASSWORD ogg
RMTHOST ggdemo, MGRPORT 7810
RMTTASK replicat, GROUP repi1
TABLE data.employees1;

On Target

On the SQL Prompt, truncate the table empr1.

SQL> truncate table empr1;

Then, on GGSCI prompt:

3) **Create the initial data load task 'repi1'**

GGSCI> **ADD REPLICAT repi1, SPECIALRUN**
REPLICAT added.

4) **Create the parameter file for the Replicat group, repi1**

GGSCI (devu007) 2> **EDIT PARAMS repi1**

REPLICAT repi1
USERID ogg, PASSWORD ogg
ASSUMETARGETDEFS
MAP data.employees1, TARGET data.empr1;

On Source

SQL> **select count(*) from data.employees1**;

COUNT(*)
———-
72

On Target

SQL> **select count(*) from data.empr1;**

COUNT(*)
———-
0

5) **Start the initial load data extract task on the source system**

On Source

GGSCI > **START EXTRACT exti1**

Sending START request to MANAGER …
EXTRACT exti1 starting

GGSCI > **info extract exti1**

EXTRACT exti1 Last Started 2010-02-11 11:33 Status RUNNING
Checkpoint Lag Not Available
Log Read Checkpoint Table data.employees1
2010-02-11 11:33:16 Record 72
Task SOURCEISTABLE
GGSCI > **info extract exti1**

EXTRACT EXTI1 Last Started 2010-02-11 11:33 Status STOPPED
Checkpoint Lag Not Available
Log Read Checkpoint Table data.employees1
2010-02-11 11:33:16 **Record 72**
Task SOURCEISTABLE

On Target

SQL> **select count(*) from data.employees1;**

COUNT(*)
———-
72

3.3 Filtering and Mapping data

SQL prompt

Create the table structures and procedures on both the sides

On Source

conn data/welcome1

create table employees2 as select * from scott.emp;

On Target

conn data/welcome1

```
CREATE table emp_dw2 (EMPNO NUMBER(4),ENAME VARCHAR2(10),
JOB VARCHAR2(9),
MGR NUMBER(4),
HIREDATE DATE,
SAL NUMBER(7,2),
COMM NUMBER(7,2),
DEPTNO NUMBER(2));

CREATE OR REPLACE PROCEDURE DATE_LOOKUP
(DESC_PARAM OUT DATE) is
BEGIN
SELECT SYSDATE INTO DESC_PARAM
FROM DUAL;
END;
/
```

GGSCI Prompt

Using One Extract Process

On source Side

GGSCI>add trandata data.*

GGSCI>info trandata data.*

GGSCI>edit params xtst0f

```
extract xtst0f
userid OGG, password ogg
tranlogoptions asmuser sys@ASM, asmpassword welcome1
discardfile ./dirrpt/xtst0f.dsc,purge
reportcount every 15 minutes, rate
rmthost ggdemo, mgrport 7810
rmttrail /oracle/golden_gate/gg_dev /dirdat/af
table data.employees2, FILTER (SAL >1000);
```

GGSCI>add extract xtst0f, tranlog, begin now

GGSCI>add rmttrail /oracle/golden_gate/gg_dev /dirdat/af, extract xtst0f, megabytes 100

GGSCI>start xtst0f

On Target Side

GGSCI>edit params rtstf1

```
replicat rtstf1
userid OGG, password ogg
discardfile ./dirrpt/rtstf1.dsc, purge
assumetargetdefs
reportcount every 15 minutes, rate
batchsql
MAP data.employees2, TARGET data.emp_dw2, FILTER (@RANGE (1,3)),
SQLEXEC (id dlookup, spname date_lookup),
COLMAP (USEDEFAULTS, hiredate = @GETVAL(dlookup.desc_param));
```

GGSCI>edit params rtstf2

```
replicat rtstf2
userid OGG, password ogg
discardfile ./dirrpt/rtstf2.dsc, purge
assumetargetdefs
reportcount every 15 minutes, rate
batchsql
MAP data.employees2, TARGET data.emp_dw2, FILTER (@RANGE (2,3)), SQLEXEC (id dlooku
p,spname data.date_lookup), COLMAP (USEDEFAULTS, hiredate = @GETVAL(dlookup.desc_pa
ram));
```

GGSCI>edit params rtstf3

```
replicat rtstf3
userid OGG, password ogg
discardfile ./dirrpt/rtstf3.dsc, purge
assumetargetdefs
reportcount every 15 minutes, rate
batchsql
MAP data.employees2, TARGET data.emp_dw2, FILTER (@RANGE (3,3)), SQLEXEC (id dlooku
p,spname data.date_lookup), COLMAP (USEDEFAULTS, hiredate = @GETVAL(dlookup.desc_pa
ram));
```

GGSCI>DBLOGIN USERID ogg, PASSWORD ogg

GGSCI>add checkpointtable ogg.checkpoint_gg

GGSCI>add replicat rtstf1, exttrail ./dirdat/af,checkpointtable ogg.checkpoint_gg

GGSCI>add replicat rtstf2, exttrail ./dirdat/af,checkpointtable ogg.checkpoint_gg

GGSCI>add replicat rtstf3, exttrail ./dirdat/af,checkpointtable ogg.checkpoint_gg

---

GGSCI>send replicat rtstf1, status

GGSCI>send replicat rtstf2, status

GGSCI>send replicat rtstf3, status

3.4 Reverse utility

create the table on both source and target

```
create table data.test1(
id    number(10) primary key,
c_desc    varchar2(60),
c_date    date);
```

Only on Source SQL Prompt, set NLS_DATE_FORMAT

```
SQL>ALTER SESSION SET NLS_DATE_FORMAT='DD-MON-YYYY HH24:MI:SS';
```

select sysdate for BEGIN TIME

```
SQL> select sysdate from dual;
```

Insert data into source and target both (Ideally, in the real world we would have inserted the data only on the source and GG would have transferred it to the target but here just for demo purpose, since we are not configuring change synchronization for table test1 therefore we are manually inserting the values on both the sides)

```
SQL>insert into data.test1 values ( 1,'Test 1', sysdate);
SQL>insert into data.test1 values ( 2,'Test 2', sysdate);
SQL>insert into data.test1 values ( 3,'Test 3', sysdate);
SQL>commit;

SQL> select * from data.test1;
```

select sysdate for END TIME

```
SQL> select sysdate from dual;
```

On GGSCI Prompt, on source side configure extract

GGSCI> edit params extcap

extract extcap
userid ogg, password ogg
tranlogoptions asmuser sys@ASM, asmpassword welcome1
end <end time>
nocompressdeletes
getupdatebefores
rmthost ggdemo, mgrport 7810
rmttrail /oracle/golden_gate/gg_dev/dirdat/ar
table data.test1, keycols(id);

GGSCI> add extract extcap, tranlog, begin <begin time>

GGSCI> add rmttrail /oracle/golden_gate/gg_dev/dirdat/ar, extract extcap

Start extract

GGSCI> start extcap

On Target side configure replicat

GGSCI> edit params repcap

replicat repcap
end runtime
userid ogg, password ogg
assumetargetdefs
discardfile ./dirrpt/repcat.dsc, megabytes 4, purge
map data.test1, target data.test1;

GGSCI> add replicat repcap, exttrail ./dirdat/br, checkpointtable checkpoint_gg

Run the reverse utility from Golden gate Installation Home

$ reverse ./dirdat/ar000000,./dirdat/br000000

Run the replicat to reverse all the transactions

GGSCI> start replicat repcap

Day 4. Bi-directional Replication, DDL Replication, Encryption and Compression

4.1 Configure DDL replication

Run the scripts as SYSDBA

SQL> **@marker_setup**

Enter GoldenGate schema name: **OGG**

SQL> **alter session set recyclebin=OFF;**

SQL> **@ddl_setup**

Enter GoldenGate schema name: **OGG**

Enter mode of installation:INITIALSETUP

RECYCLEBIN must be empty.

This installation will purge RECYCLEBIN for all users.

To proceed, enter yes. To stop installation, enter no.

Enter yes or no: **yes**

SQL> **@role_setup**

Enter GoldenGate schema name: **OGG**

SQL> **grant ggs_ggsuser_role to ogg;**

SQL> **@ddl_enable**

SQL> **@ddl_pin GGS_OWNER**

4.2 Configuring Bi-directional Replication along with DDL replication

4.2.1 Create the table structures on both the sides

On both source and the target

create user data identified by welcome1;
grant connect,resource,create table to data;
grant all on scott.emp to data;

On Source

ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
Alter system switch logfile;
conn data/welcome1
create table employees4 as select * from scott.emp;
ALTER TABLE employees4 add CONSTRAINT pk_sou1 PRIMARY KEY (empno);

On Target

ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
Alter system switch logfile;
conn data/welcome1
create table empr4 as select * from scott.emp;
ALTER TABLE empr4 add CONSTRAINT pk_tar1 PRIMARY KEY (empno);

4.2.2 GGSCI Prompt

On source ORCL Side

GGSCI> dblogin userid ogg password ogg
GGSCI>add trandata data.*
GGSCI>info trandata data.*
GGSCI>edit params xtor01
    extract xtor01
    userid OGG, password ogg
    discardfile ./dirrpt/xtor01.dsc,purge
    reportcount every 15 minutes, rate
    rmthost ggdemo, mgrport 7810
    rmttrail /oracle/golden_gate/gg_dev/dirdat/b1
    tranlogoptions asmuser sys@ASM, asmpassword welcome1
    DDL INCLUDE MAPPED

tranlogoptions excludeuser ogg
table data.employees4;

GGSCI>add extract xtor01, tranlog, begin now

GGSCI>add rmttrail /oracle/golden_gate/gg_dev/dirdat/b1, extract xtor01, megabytes 100

GGSCI>start xtor01

GGSCI>edit params rtor01

replicat rtor01
userid OGG, password ogg
discardfile ./dirrpt/rtor01.dsc, purge
assumetargetdefs
reportcount every 15 minutes, rate
batchsql
MAP data.empr4, TARGET data.employees4;

GGSCI>DBLOGIN USERID ogg, PASSWORD ogg

GGSCI>add checkpointtable ogg.checkpoint_gg

GGSCI>add replicat rtor01, exttrail ./dirdat/b2,checkpointtable ogg.checkpoint_gg

On Target DEV Side

GGSCI> dblogin userid ogg password ogg
GGSCI>add trandata data.*

GGSCI>info trandata data.*

GGSCI>edit params xtde01

extract xtde01
userid OGG, password ogg
discardfile ./dirrpt/xtde01.dsc,purge
reportcount every 15 minutes, rate
rmthost ggdemo, mgrport 7809
rmttrail /oracle/golden_gate/gg_orcl/dirdat/b2
tranlogoptions excludeuser ogg

table data.empr4;

GGSCI>add extract xtde01, tranlog, begin now

GGSCI>add rmttrail /oracle/golden_gate/gg_orcl/dirdat/b2, extract xtde01, megabytes 100

GGSCI>start xtde01

GGSCI>edit params rtde01

replicat rtde01
userid OGG, password ogg
discardfile ./dirrpt/rtde01.dsc, purge
assumetargetdefs
reportcount every 15 minutes, rate
batchsql
MAP data.employees4, TARGET data.empr4;

GGSCI>DBLOGIN USERID ogg, PASSWORD ogg

GGSCI>add checkpointtable ogg.checkpoint_gg

GGSCI>add replicat rtde01, exttrail ./dirdat/b1,checkpointtable ogg.checkpoint_gg

4.3 Configuring Encryption

GoldenGate provides the following encryption options:

1) The data stored in extract and replicat trail files
2) Passwords used in the extract and replicat parameter files
3) Data send over TCP/IP networks

On Source

conn data/welcome1

create table employees5 as select * from scott.emp;

ALTER TABLE employees5 add CONSTRAINT pk_empl5 PRIMARY KEY (empno);

On Target

conn data/welcome1

create table empr5 as select * from scott.emp;

ALTER TABLE empr5 add CONSTRAINT pk_empr5 PRIMARY KEY (empno);

- To encrypt trail or extract files, GoldenGate uses 256-key byte substitution. All records going into those files are encrypted both across any data links and within the files themselves.
- To encrypt the database password or data sent across TCP/IP, GoldenGate uses Blowfish encryption.

4.3.1 Setting up encryption

Let us examine some of the steps involved in setting up the encryption with GoldenGate.

Generate Encryption Keys

Run the keygen command from the GoldenGate software installation home

KEYGEN (key length) (n)

Where:

(key length) is the encryption key length, up to 128 bits.

(n) represents the number of keys to generate.

$ ./keygen 128 4

0x0A0E5C624211E87040B50129726C0371

0x0D44A10F0A6A05101FCE1E2003F0B405

0x0F7AE63CD1C2222FFEE63B179373661A

0xBB5A266A0AFF58158771E5599E5AB84C

We will then create a text file called ENCKEYS and in this file for each key that has been generated we will provide a logical name as shown below

$ vi ENCKEYS

securekey1 0x0A0E5C624211E87040B50129726C0371

securekey2 0x0D44A10F0A6A05101FCE1E2003F0B405

securekey3 0x0F7AE63CD1C2222FFEE63B179373661A

securekey4 0xBB5A266A0AFF58158771E5599E5AB84C

We will then copy the ENCKEYS file to the GoldenGate software location on our target server

$ cp ENCKEYS /oracle/golden_gate/gg_dev/

4.3.2 Encrypt database passwords

Run GGSCI and issue the ENCRYPT PASSWORD command along with the logical name of the secure key (from the ENCKEYS file we have just created)to generate an encrypted password

GGSCI> encrypt password ogg encryptkey securekey1

Encrypted password: AACAAAAAAAAAAAJAUEUGODSCVGJEEIUGKJDJTFNDKEJFFFTC

Let us now test this encrypted password

GGSCI> dblogin userid ogg, password AACAAAAAAAAAAAJAUEUGODSCVGJEEIUGKJDJTFNDKEJFFFTC, encryptkey securekey1

Successfully logged into database.

4.3.3 Encrypt data sent over TCP/IP

You can encrypt captured data before GoldenGate sends it across the TCP/IP network to the target system.

On the target system, GoldenGate decrypts the data before writing it to the GoldenGate trail files. By default, data sent across the network is not encrypted.

For example in our extract parameter file we will add the ENCRYPT BLOWFISH along with the logical name for our secure encryption key (taken from the ENCKEYS file) as shown below

RMTHOST ggdemo, MGRPORT 7810, ENCRYPT BLOWFISH, KEYNAME securekey1

4.3.4 Encrypt Trail and Extract files

We can also encrypt the data in any local or remote trail or file.

In the Extract parameter file we use the keyword ENCRYPTTRAIL before all trails or files that you want to be encrypted.

In the Replicat parameter file, include the parameter DECRYPTTRAIL so that Replicat decrypts the data for processing.

Let us now look at an example of an Extract and Replicat parameter file where we have used all three encryption features where the OGG database password has been encrypted,

the trail files have been enabled for encryption and decryption and the TCP/IP network connectivity to the remote site also has encryption enabled.

On the Source ORCL side

EXTRACT xtsec
USERID ogg, PASSWORD <encrypted password generated by you >, ENCRYPTKEY securekey1
tranlogoptions asmuser sys@ASM asmpassword welcome1
RMTHOST ggdemo, MGRPORT 7810, ENCRYPT BLOWFISH, KEYNAME securekey1
ENCRYPTTRAIL RMTTRAIL /oracle/golden_gate/gg_dev/dirdat/a5
TABLE data.employees5;

GGSCI>add extract xtsec, tranlog, begin now

GGSCI>add rmttrail /oracle/golden_gate/gg_dev/dirdat/a5, extract xtsec, megabytes 100

GGSCI>start xtsec

On the target DEV side

REPLICAT rtsec
HANDLECOLLISIONS
DECRYPTTRAIL
ASSUMETARGETDEFS
USERID ogg, PASSWORD <encrypted password generated by you>, ENCRYPTKEY securekey1
MAP data.employees5, target data.empr5;

GGSCI>DBLOGIN USERID ogg, PASSWORD ogg

GGSCI>add checkpointtable ogg.checkpoint_gg

GGSCI>add replicat rtsec, exttrail ./dirdat/a5,checkpointtable ogg.checkpoint_gg

---

End of the Lab Guide