

HANDS ON LAB ORACLE GOLDENGATE 19C & MULTITENANCY

DATE: 15-09-2020

=== Hands On Lab : Oracle GoldenGate 19c & Multitenancy ===

Change History

VERSION	DATE	DESCRIPTION	AUTHOR
0.1	05-09-2020	Initial version	Rob Lasonder

Sources

SOURCE

Contents

1.	ABOUT THIS LAB	4
2.	PREPARE THE GOLDENGATE DATABASE ENVIRONMENT	5
2.1.	PREREQUISITES.....	5
2.2.	PREPARE THE CDB\$ROOT DATABASE.....	5
2.3.	CREATE THE SOURCE PDB4 AND THE TARGET PDB5 DATABASES.....	8
2.4.	CONFIGURE SQLNET CONNECTIVITY FOR THE PLUGGABLE DATABASES PDB4 AND PDB5	8
2.5.	CREATE A CUSTOM HEARTBEAT TABLE IN SOURCE DATABASE PDB4.....	10
2.6.	CHECK THE CUSTOM HEARTBEAT TABLE IN SOURCE DATABASE PDB4	12
3.	INSTALL GOLDENGATE 19C	13
3.1.	DOWNLOAD AND STAGE GOLDENGATE	13
3.2.	CREATE A DIRECTORY FOR THE GOLDENGATE SOFTWARE (ON THE VIRTUAL SERVER).....	13
3.3.	ADD THE GOLDENGATE HOME DIRECTORY TO THE .BASH_PROFILE.....	13
3.4.	UNZIP THE SOFTWARE IN THE GOLDENGATE HOME.....	13
3.5.	RUN THE GOLDENGATE INSTALLER	14
4.	CONFIGURE THE GOLDENGATE PROCESSES	17
4.1.	CREATE THE GOLDENGATE SUBDIRECTORIES	17
4.2.	CONFIGURE THE CREDENTIAL STORE	17
4.3.	ADD SUPPLEMENTAL LOGGING FOR THE PDB TABLES.....	18
4.4.	CONFIGURE THE MANAGER PROCESS.....	19
4.5.	CONFIGURE THE EXTRACT PROCESS.....	21
4.6.	CONFIGURE THE PUMP PROCESS	23
4.7.	INSTANTIATE THE TABL/SCHEMA IN THE TARGET DATABASE	26
4.8.	CONFIGURE THE REPLICAT PROCESS	28
5.	ENCOUNTERED ERRORS DURING THE SETUP	31

1. About this Lab

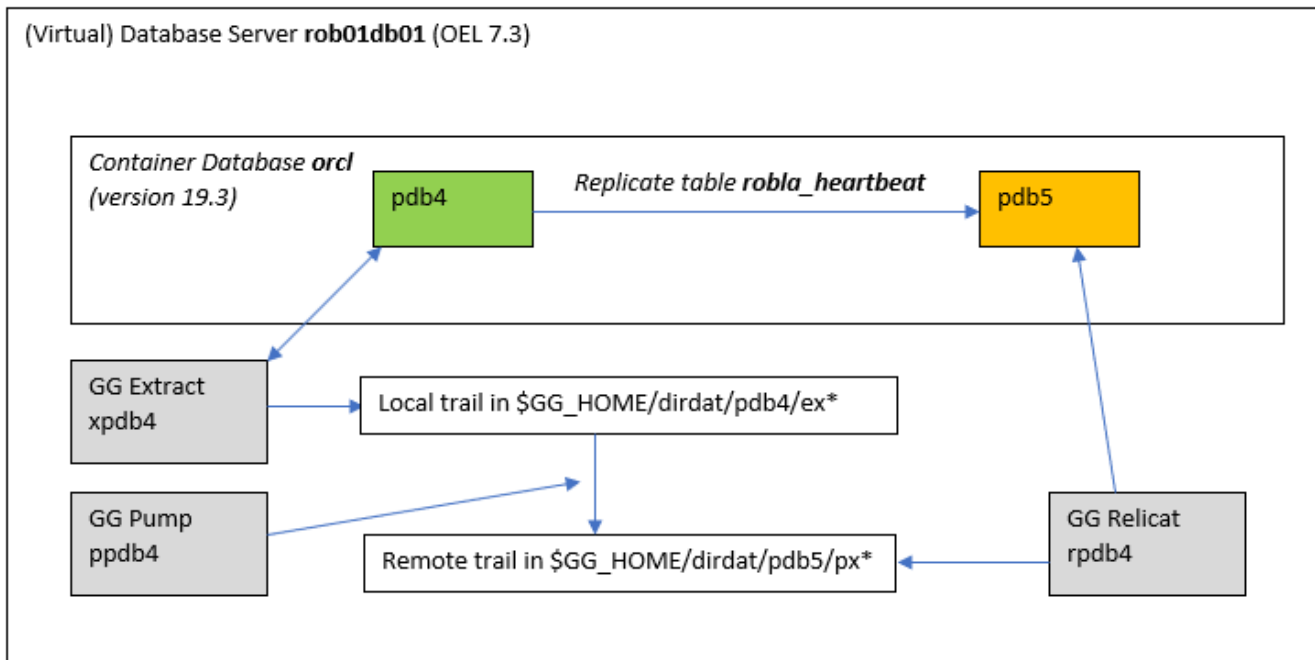
In this lab I configure a GoldenGate 19c replication of table ROBLA.RL_HEARTBEAT.

This table is located inside pluggable database pdb4 of cdb orcl and it will be replicated to pluggable database pdb5 of the same cdb. Table robla.rl_heartbeat has a record inserted every minute (via a DBMS_SCHEDULER-JOB), so we can trace these records from source to target.

This lab assumes that you have a 19c Container Database up and running, but other than that, all necessary step to setup the replication will be performed:

- Create pluggable databases pdb4 and pdb5
- Create the heartbeat table plus related objects in pdb4
- Install GoldenGate 19c
- Prepare the source and target pluggables for the replication
- Setup the GoldenGate replication of the heartbeat table between pdb4 and pdb5

We will build the setup as shown below:



2. Prepare the GoldenGate database environment

2.1. Prerequisites

We already have a 19.3 Container database installed:

```
SQL> show con_name
CON_NAME
-----
CDB$ROOT
SQL> select banner_full from v$version;
BANNER_FULL
-----
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
SQL> show parameter db_create_file_dest
NAME                                TYPE          VALUE
-----
db_create_file_dest                 string        /u01/app/oracle/oradata
```

2.2. Prepare the CDB\$ROOT Database

2.2.1. Configure the streams pool

The integrated extract uses the streams pool. So it needs to be configured. The default is 0.

```
SQL> show parameter sga_target
NAME                                TYPE          VALUE
-----
sga_target                           big integer 1184M
SQL> show parameter streams
NAME                                TYPE          VALUE
-----
streams_pool_size                     big integer 0
SQL> alter system set streams_pool_size=300M;
System altered.
```

2.2.2. Enable archive log and force logging

To make sure all the changes are recored in the online redo log files:

```
SQL> show con_name
CON_NAME
-----
CDB$ROOT
SQL> select log_mode, force_logging from v$database;
LOG_MODE      FORCE_LOGGING
-----
ARCHIVELOG    NO
SQL> alter database force logging;
Database altered.
SQL> select log_mode, force_logging from v$database;
LOG_MODE      FORCE_LOGGING
-----
ARCHIVELOG    YES
```

2.2.3. Add supplemental logging

If at least minimal supplemental logging is enabled in CDB\$ROOT, then additional supplemental logging levels can be enabled at the PDB level. Enable supplemental logging at the root level as follows:

```
SQL> alter session set container = cdb$root;
Session altered.
SQL> select * from dba_supplemental_logging ;
MIN PRI UNI FOR ALL PRO SUB
--- --- --- --- ---
NO NO NO NO NO NO NO
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
Database altered.
SQL> select * from dba_supplemental_logging ;
MIN PRI UNI FOR ALL PRO SUB
--- --- --- --- ---
YES NO NO NO NO NO NO
```

=== Hands On Lab : Oracle GoldenGate 19c & Multitenancy ===

2.2.4. Create the GoldenGate common user

The GoldenGate user needs to be created in the CDB. The GoldenGate db package is used to grant most of the privileges required and then a few additional rights are also granted.

```
SQL> create user c#gg_admin identified by gg_admin;
User created.
SQL> exec dbms_goldengate_auth.grant_admin_privilege('c###gg_admin','capture',container=>'all');
PL/SQL procedure successfully completed.
SQL> grant dba, create session, alter session, set container to c###gg_admin container=all;
SQL> grant select any dictionary to c###gg_admin container=all;
SQL> select privilege from dba_sys_privs where grantee = 'C##GG_ADMIN';

PRIVILEGE
-----
CREATE JOB
ALTER SESSION
LOGMINING
CREATE EVALUATION CONTEXT
DEQUEUE ANY QUEUE
CREATE RULE
FLASHBACK ANY TABLE
SELECT ANY TABLE
ALTER ANY TABLE
CREATE SESSION
EXECUTE ANY RULE SET
CREATE RULE SET
SET CONTAINER
13 rows selected.
```

2.2.5. Enable GoldenGate Replication

```
SQL> alter system set enable_goldengate_replication=TRUE;
System altered.
```

2.3. Create the source PDB4 and the target PDB5 databases

```
SQL> create pluggable database pdb4 admin user pdbadmin identified by pdbadmin roles = (DBA, CONNECT);
Pluggable database created.

SQL> create pluggable database pdb5 admin user pdbadmin identified by pdbadmin roles = (DBA, CONNECT);
Pluggable database created.

SQL> alter pluggable database pdb4 open;
Pluggable database altered.

SQL> alter pluggable database pdb4 save state;
Pluggable database altered.

SQL> alter pluggable database pdb5 open;
Pluggable database altered.

SQL> alter pluggable database pdb5 save state;
Pluggable database altered.
```

2.4. Configure SQLNet connectivity for the pluggable databases PDB4 and PDB5

Navigate to \$ORACLE_HOME/network/admin and add the following entries to the tnsnames.ora:

```
PDB4 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = rob01db01.robdomain)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = pdb4.robdomain)
    )
  )
)

PDB5 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = rob01db01.robdomain)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = pdb5.robdomain)
    )
  )
)
```


=== Hands On Lab : Oracle GoldenGate 19c & Multitenancy ===

Test out SQL*Net connectivity as follows (also for pdb5):

```
[oracle@rob01db01 admin]$ sqlplus pdbadmin/pdbadmin@pdb4
SQL*Plus: Release 19.0.0.0.0 - Production on Wed Sep 16 09:11:33 2020
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

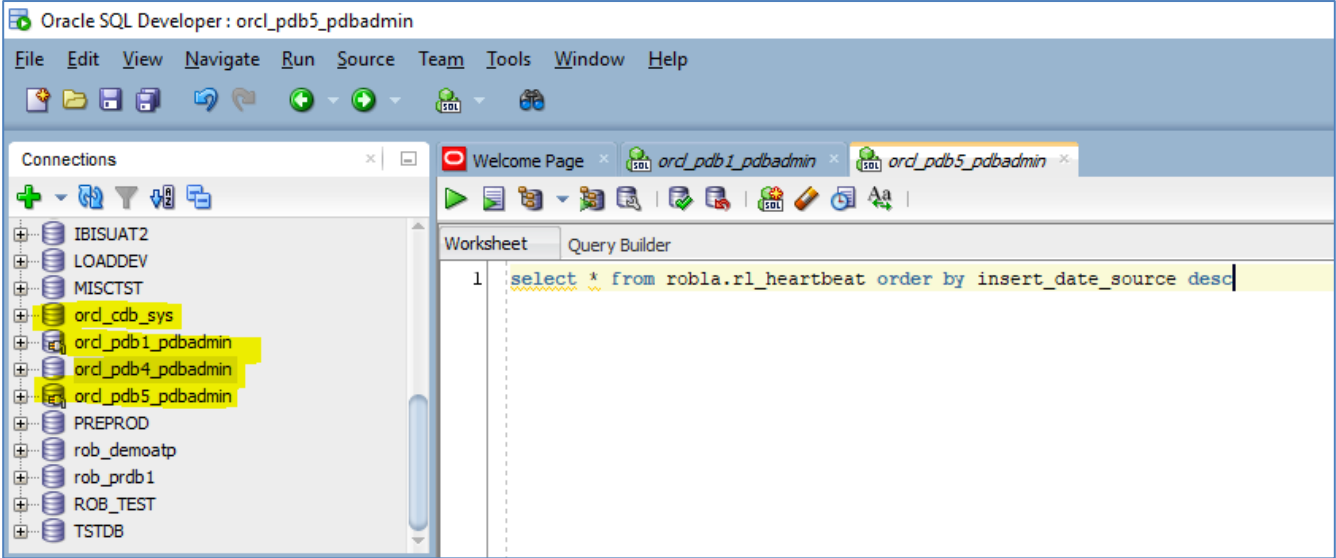
Last Successful login time: Wed Sep 16 2020 09:11:23 +01:00

Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0

SQL> show con_name

CON_NAME
-----
PDB4
```

You can optionally configure SQL Developer connections as well,. But this is not required for this lab:



Connection Name:

Username:

Password:

Save Password Connection Color

Oracle

Connection Type: Role:

Hostname:

Port:

SID

Service name:

2.5. Create a custom heartbeat table in source database PDB4

Execute the code below to create a custom heartbeat table and job to populate this table. This table has a record inserted every minute. This will be the table we will replicate from PDB4 to PDB5 in this GoldenGate demo.

```
SQL> alter session set container=pdb4;
```

```
Session altered.
```

```
SQL> create user robla identified by robla;
```

```
User created.
```

```
SQL> grant dba to robla;
```

```
Grant succeeded.
```

```
SQL> set echo on
```

```
SQL> @cr_heartbeat.sql
```

```
SQL> spool /home/oracle/dbadir/robla/scripts/cr_heartbeat.log
```

```
SQL> -- create the heartbeat table
```

```
SQL> drop table robla.rl_heartbeat;
```

```
Table dropped.
```

```
SQL>
```

```
SQL> create table robla.rl_heartbeat (  
 2 id NUMBER PRIMARY KEY,  
 3 colour varchar2(10),  
 4 insert_date_source date,  
 5 insert_date_target date,  
 6 update_date_source date,  
 7 update_date_target date  
 8 )  
 9 PARTITION BY RANGE (insert_date_source)  
10 INTERVAL(NUMTOYMINTERVAL(1, 'MONTH'))  
11 (  
12     PARTITION part_01 values LESS THAN (TO_DATE('01-NOV-2019','DD-MON-YYYY'))  
13 );
```

```
Table created.
```

```
SQL>
```

```
SQL>
```

```
SQL> -- create a sequence for the table
```

```
SQL> drop sequence ROBLA.RL_SEQ_HEARTBEAT;
```

```
Sequence dropped.
```

```
SQL>
```

```
SQL> CREATE SEQUENCE ROBLA.RL_SEQ_HEARTBEAT  
 2 START WITH 1  
 3 INCREMENT BY 1  
 4 CACHE 100  
 5 NOCYCLE;
```

```
Sequence created.
```

```
SQL>
```

```
SQL> -- create a trigger for the table to insert the sequence value
```

```
SQL> CREATE OR REPLACE TRIGGER ROBLA.rl_trg_heartbeat  
 2 BEFORE INSERT ON ROBLA.rl_heartbeat  
 3 FOR EACH ROW
```

=== Hands On Lab : Oracle GoldenGate 19c & Multitenancy ===

```
4 BEGIN
5     SELECT ROBLA.rl_seq_heartbeat.nextval
6         INTO :new.id
7         FROM dual;
8 END;
9 /
```

Trigger created.

SQL>

SQL>

SQL> -- create a job to insert records in the table

SQL> begin

```
2     dbms_scheduler.drop_job(job_name => 'ROBLA.insert_rl_heartbeat');
3 end;
4 /
```

PL/SQL procedure successfully completed.

SQL>

SQL> BEGIN

```
2     DBMS_SCHEDULER.CREATE_JOB (
3         job_name           => 'ROBLA.insert_rl_heartbeat',
4         job_type            => 'PLSQL_BLOCK',
5         job_action          => 'BEGIN insert into ROBLA.rl_heartbeat (colour,insert_date_source)
values ('GREEN',sysdate); commit; END;',
6         repeat_interval    => 'freq=MINUTELY;interval=1',
7         enabled             => TRUE,
8         comments            => 'insert records in manual heartbeat table every minute');
9 END;
10 /
```

PL/SQL procedure successfully completed.

SQL>

SQL> spool off

2.6. Check the custom heartbeat table in source database PDB4

After some time, we can check the heartbeat table in PDB4. With the query below we can see the last 6 inserted rows:

```
SQL> alter session set container=pdb4;
SQL> set echo on
SQL> @check_heartbeat_source.sql
SQL> select * from (
  2  select id, colour, sysdate, insert_date_source, insert_date_target from robla.rl_heartbeat order
by INSERT_DATE_SOURCE desc
  3  ) where rownum < 7;
```

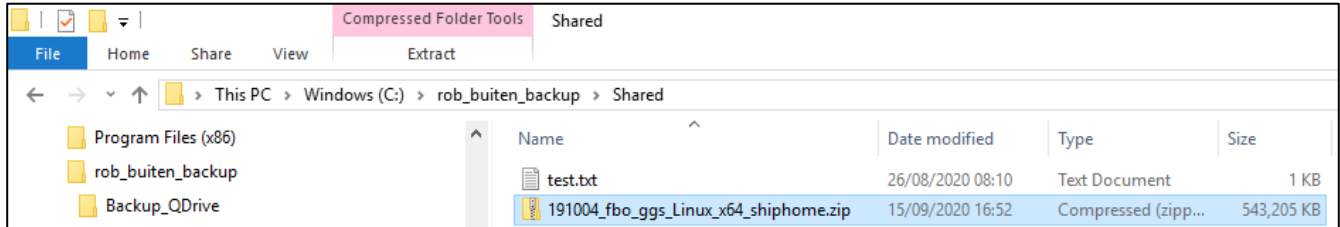
ID	COLOUR	SYSDATE	INSERT_DATE_SOURCE	INSERT_DATE_TARGET
24	GREEN	2020-SEP-16 09:15:05	2020-SEP-16 09:14:25	
23	GREEN	2020-SEP-16 09:15:05	2020-SEP-16 09:13:25	
22	GREEN	2020-SEP-16 09:15:05	2020-SEP-16 09:12:25	
21	GREEN	2020-SEP-16 09:15:05	2020-SEP-16 09:11:25	
20	GREEN	2020-SEP-16 09:15:05	2020-SEP-16 09:10:25	
19	GREEN	2020-SEP-16 09:15:05	2020-SEP-16 09:09:25	

6 rows selected.

3. Install GoldenGate 19c

3.1. Download and stage GoldenGate

The software can be downloaded from Oracle and staged on the host, in the folder that is shared with the virtual machine:



3.2. Create a directory for the GoldenGate software (on the virtual server)

```
[oracle@rob01db01 product]$ mkdir -p /u01/app/oracle/product/ogg/19.1.0
```

3.3. Add the GoldenGate Home directory to the .bash_profile

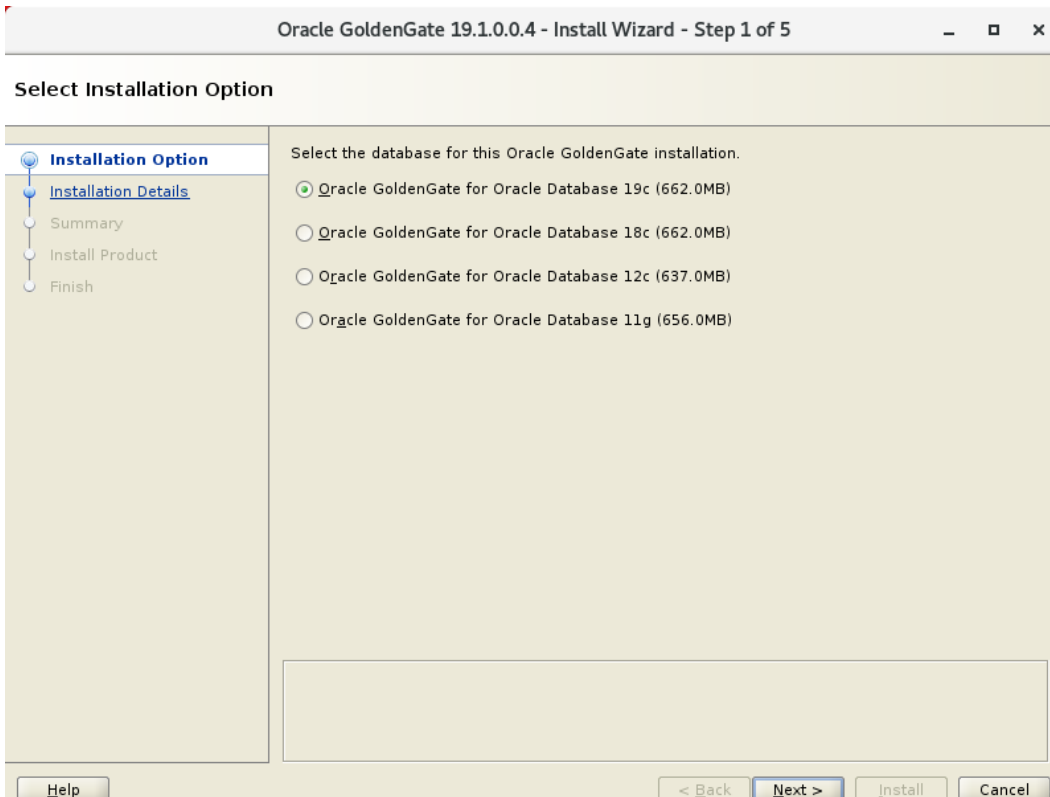
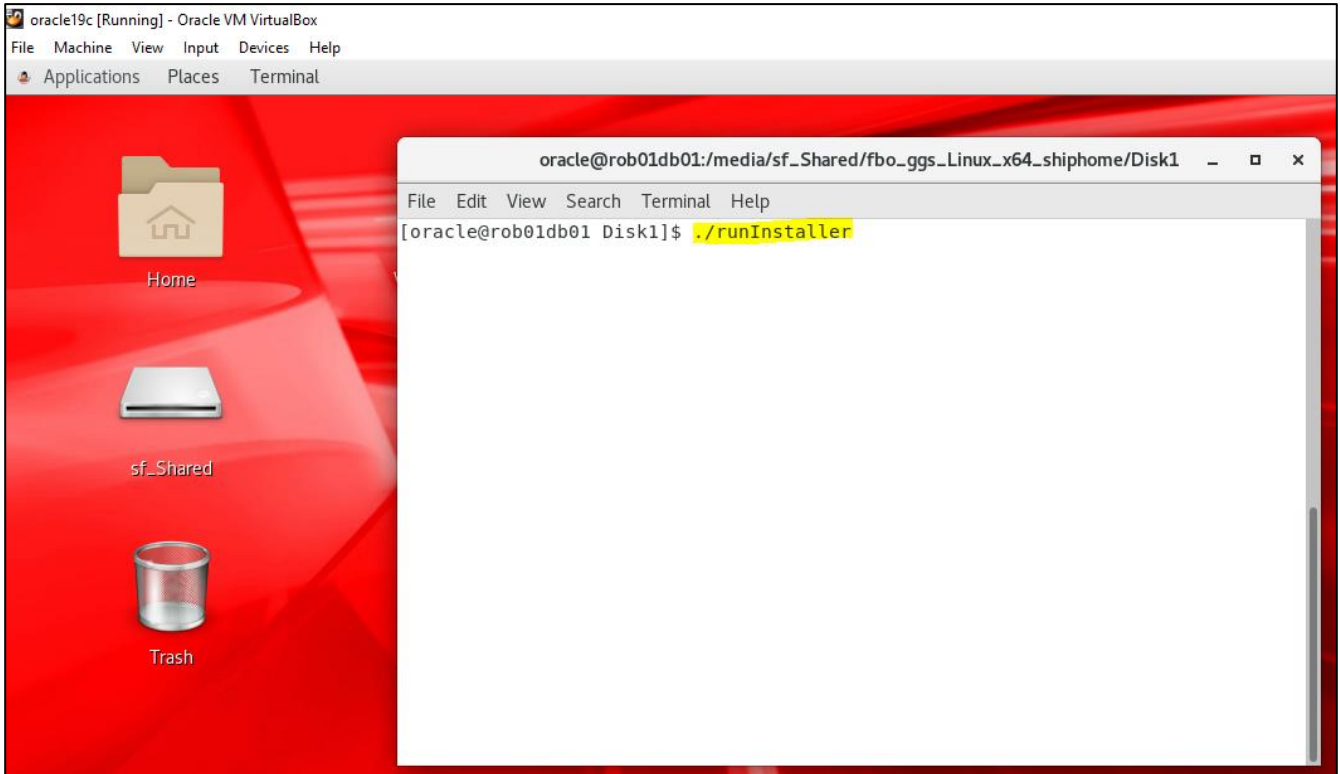
```
[oracle@rob01db01 ~]$ vi .bash_profile
--add:
export GG_HOME=/u01/app/oracle/product/ogg/19.3.0
```

3.4. Unzip the software in the GoldenGate home

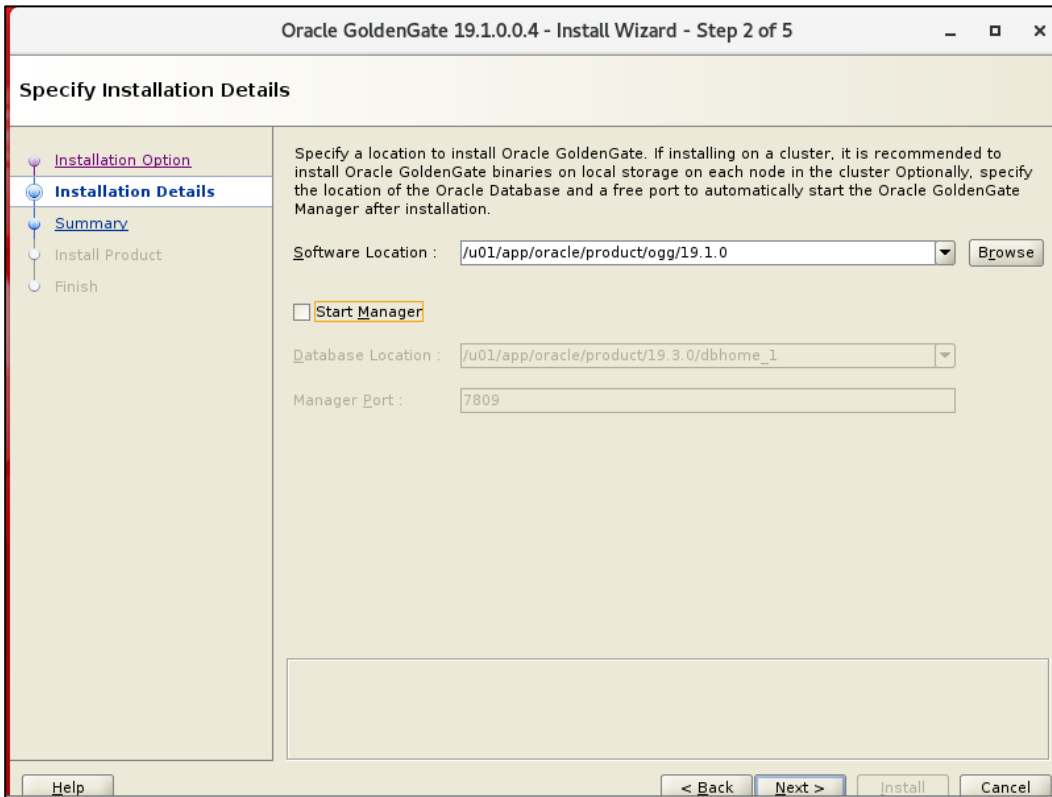
```
[oracle@rob01db01 ~]$ su -
Password: robroot
[root@rob01db01 ~]# cd /media/sf_Shared
[root@rob01db01 sf_Shared]# ls
191004_fbo_ggs_Linux_x64_shiphome.zip test.txt
[root@rob01db01 sf_Shared]# unzip 191004_fbo_ggs_Linux_x64_shiphome.zip
[root@rob01db01 sf_Shared]# exit
```

3.5. Run the GoldenGate Installer

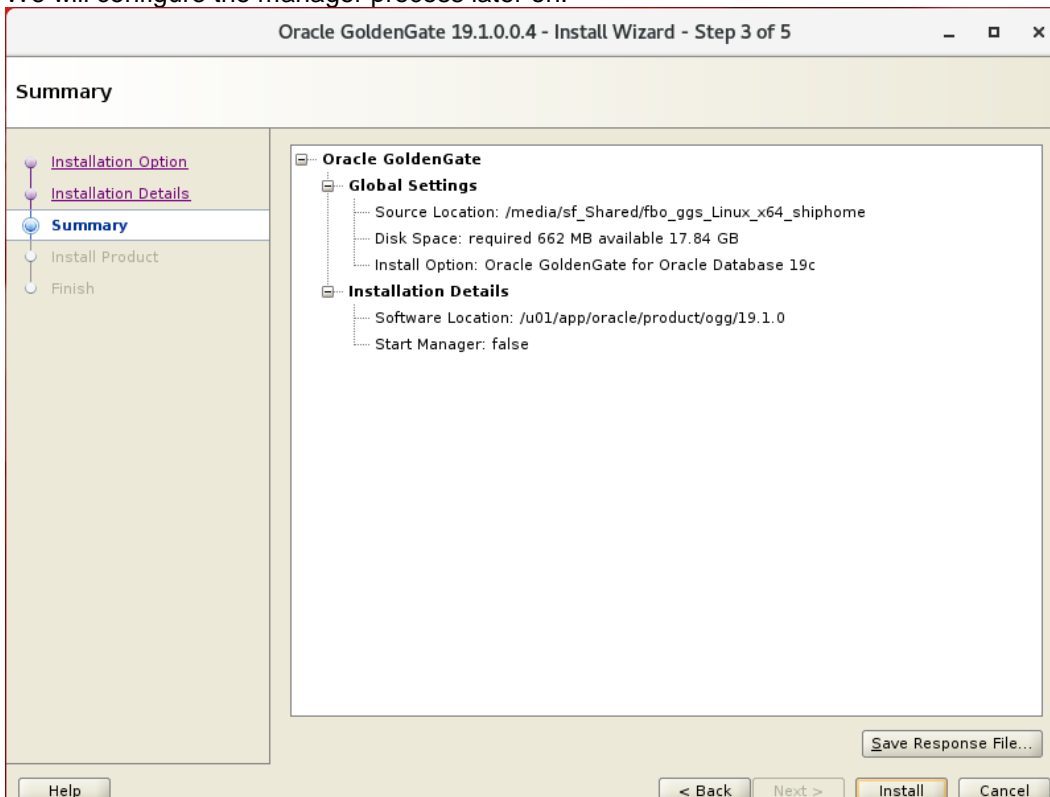
This operation needs to be done as the oracle user, via a GUI connection. The location of the installer is /media/sf_Shared/fbo_ggs_Linux_x64_shiphome/Disk1



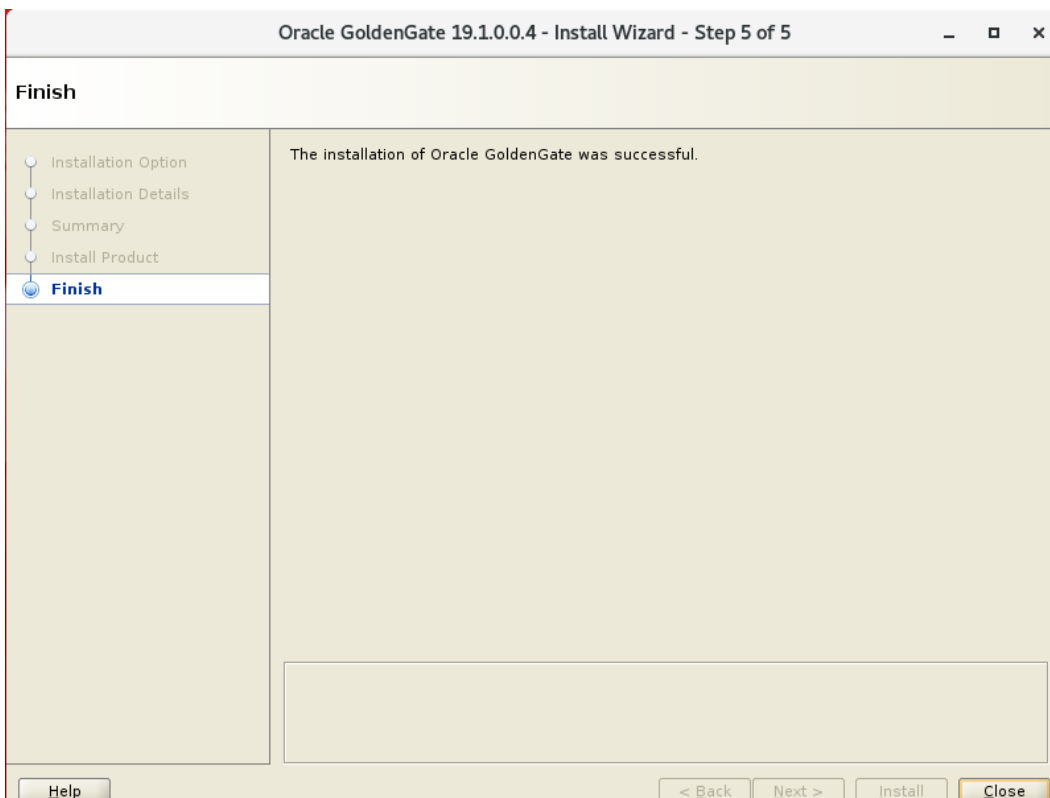
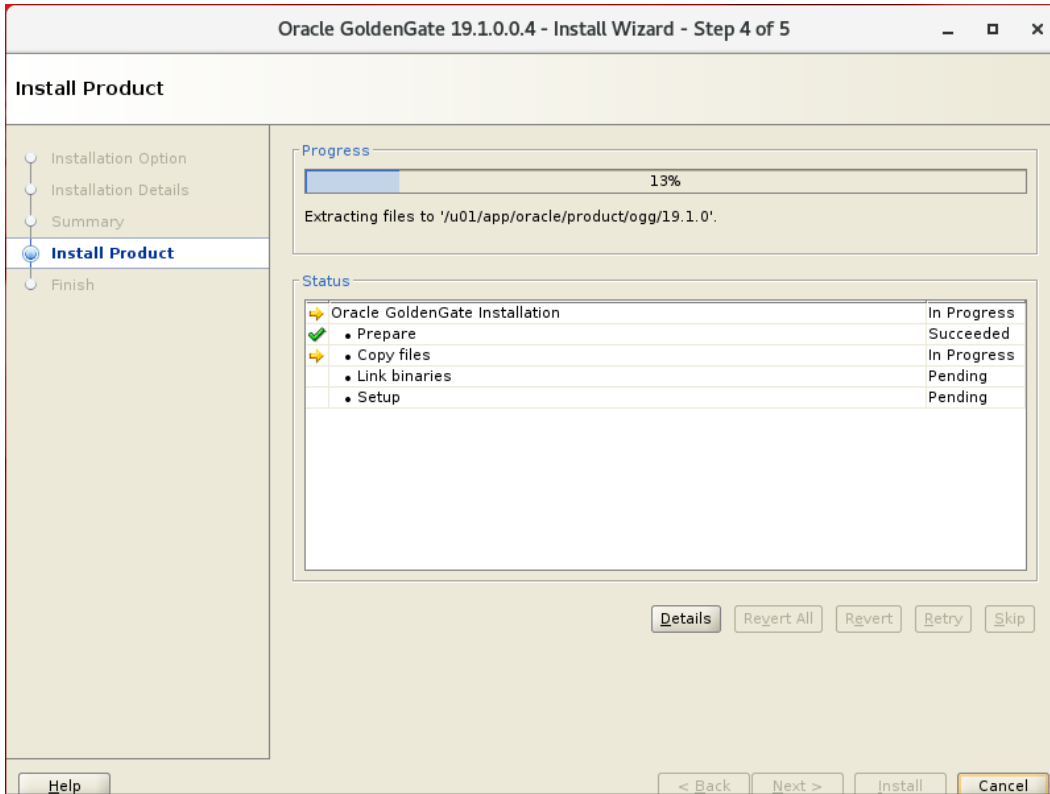
=== Hands On Lab : Oracle GoldenGate 19c & Multitenancy ===



We will configure the manager process later on.



=== Hands On Lab : Oracle GoldenGate 19c & Multitenancy ===



4. Configure the GoldenGate processes

4.1. Create the GoldenGate subdirectories

```
[oracle@rob01db01 ~]$ cd $GG_HOME
[oracle@rob01db01 19.1.0]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 19.1.0.0.4 OGGCORE_19.1.0.0.0_PLATFORMS_191017.1054_FBO
Linux, x64, 64bit (optimized), Oracle 19c on Oct 17 2019 21:16:29
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2019, Oracle and/or its affiliates. All rights reserved.

GGSCI (rob01db01.robdomain) 1> create subdirs

Creating subdirectories under current directory /u01/app/oracle/product/ogg/19.1.0

Parameter file           /u01/app/oracle/product/ogg/19.1.0/dirprm: created.
Report file              /u01/app/oracle/product/ogg/19.1.0/dirrpt: created.
Checkpoint file          /u01/app/oracle/product/ogg/19.1.0/dirchk: created.
Process status files     /u01/app/oracle/product/ogg/19.1.0/dirpcs: created.
SQL script files         /u01/app/oracle/product/ogg/19.1.0/dirsq: created.
Database definitions files /u01/app/oracle/product/ogg/19.1.0/dirdef: created.
Extract data files       /u01/app/oracle/product/ogg/19.1.0/dirdat: created.
Temporary files          /u01/app/oracle/product/ogg/19.1.0/dirtmp: created.
Credential store files    /u01/app/oracle/product/ogg/19.1.0/dircred: created.
Masterkey wallet files   /u01/app/oracle/product/ogg/19.1.0/dirwlt: created.
Dump files                /u01/app/oracle/product/ogg/19.1.0/dirdmp: created.
```

4.2. Configure the Credential Store

We will configure a credential store, so we don't have to list the passwords in the parameter files:

```
[oracle@rob01db01 ~]$ $GG_HOME/ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 19.1.0.0.4 OGGCORE_19.1.0.0.0_PLATFORMS_191017.1054_FBO
Linux, x64, 64bit (optimized), Oracle 19c on Oct 17 2019 21:16:29
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2019, Oracle and/or its affiliates. All rights reserved.

GGSCI (rob01db01.robdomain) 1> add credentialstore

Credential store created.

GGSCI (rob01db01.robdomain) 1> alter credentialstore add user c##gg_admin password gg_admin alias
c##gg_admin_root

Credential store altered.

GGSCI (rob01db01.robdomain) 2> alter credentialstore add user c##gg_admin@pdb4 password gg_admin alias
c##gg_admin_pdb4

Credential store altered.

GGSCI (rob01db01.robdomain) 3> alter credentialstore add user c##gg_admin@pdb5 password gg_admin alias
c##gg_admin_pdb5

Credential store altered.
```

=== Hands On Lab : Oracle GoldenGate 19c & Multitenancy ===

```
GGSCI (rob01db01.robdomain) 4> info credentialstore
```

```
Reading from credential store:
```

```
Default domain: OracleGoldenGate
```

```
Alias: c##gg_admin_pdb4  
Userid: c##gg_admin@pdb4
```

```
Alias: c##gg_admin_root  
Userid: c##gg_admin
```

```
Alias: c##gg_admin_pdb5  
Userid: c##gg_admin@pdb5
```

4.3. Add supplemental logging for the PDB tables

I will configure supplemental logging at the schema level. That is the easiest way, plus it has the advantage that all schema tables, including any future tables, will be supplementally logged. (Of course, if you only want to replicate a subset of your schema, you need to enable supplemental logging at the table level):

```
GGSCI (rob01db01.robdomain) 2> dblogin useridentialias c##gg_admin_pdb4
```

```
Successfully logged into database PDB4.
```

```
GGSCI (rob01db01.robdomain as c##gg_admin@orcl/PDB4) 3> info schematrandata robla
```

```
2020-09-16 13:43:00 INFO OGG-01786 Schema level supplemental logging is disabled on schema "ROBLA".
```

```
GGSCI (rob01db01.robdomain as c##gg_admin@orcl/PDB4) 4> add schematrandata robla
```

```
2020-09-16 13:43:09 INFO OGG-01788 SCHEMATRANDATA has been added on schema "robla".
```

```
2020-09-16 13:43:10 INFO OGG-01976 SCHEMATRANDATA for scheduling columns has been added on schema "robla".
```

=== Hands On Lab : Oracle GoldenGate 19c & Multitenancy ===

```
GGSCI (rob01db01.robdomain as c##gg_admin@orcl/PDB4) 2> info schematrandata robla
2020-09-16 14:29:32 INFO      OGG-06480  Schema level supplemental logging, excluding non-validated
keys, is enabled on schema "ROBLA".
2020-09-16 14:29:32 INFO      OGG-01980  Schema level supplemental logging is enabled on schema "ROBLA"
for all scheduling columns.
2020-09-16 14:29:32 INFO      OGG-10462  Schema "ROBLA" have 1 prepared tables for instantiation.
GGSCI (rob01db01.robdomain as c##gg_admin@orcl/PDB4) 6> info trandata robla.RL_HEARTBEAT
2020-09-16 14:33:34 INFO      OGG-06480  Schema level supplemental logging, excluding non-validated
keys, is enabled on schema "ROBLA".
2020-09-16 14:33:34 INFO      OGG-01980  Schema level supplemental logging is enabled on schema "ROBLA"
for all scheduling columns.
2020-09-16 14:33:34 INFO      OGG-10471  ***** Oracle Goldengate support information on table
ROBLA.RL_HEARTBEAT *****
Oracle Goldengate support native capture on table ROBLA.RL_HEARTBEAT.
Oracle Goldengate marked following column as key columns on table ROBLA.RL_HEARTBEAT: ID.

Logging of supplemental redo log data is enabled for table PDB4.ROBLA.RL_HEARTBEAT.

Columns supplementally logged for table PDB4.ROBLA.RL_HEARTBEAT: "ID".

Prepared CSN for table PDB4.ROBLA.RL_HEARTBEAT: 3698047
```

4.4. Configure the manager process

The Golden Gate Manager process is responsible for managing all the Golden Gate processes and operations within a given Golden Gate instance, including the Extract, Data Pump and Replicat process. It provides the following functionality:

- Starting Logger, Extract, Replicat and Syncfile processes.
- Monitoring and reporting the status of Oracle GoldenGate processing
- Starting the dynamic Collector process on the target
- Automatically restarting critical processes
- Threshold reporting, such as when Extract falls behind the TMF-audit trail
- Managing resources for the TMF audit trail, such as maintaining copies of audit trails on backup volumes
- Purging trails when Extract and Replicat has finished with them
- Pre-allocating log trail space for Logger processing

Every Golden Gate instance has only 1 Manager process running. This means that we will have one Manager process on the compute node, which will direct all the different extract and replicat processes. Note that if you have more than 1 Golden Gate instances up and running on the same database server (as may be the case when you host the database on a consolidated platform such as Exadata), you may require more physical Golden Gate installations.

=== Hands On Lab : Oracle GoldenGate 19c & Multitenancy ===

4.4.1. Configure the manager process

```
[oracle@rob01db01 gghome_1]$ cd $GG_HOME
[oracle@rob01db01 19.1.0.0]$ ./ggsci
Oracle GoldenGate Command Interpreter for Oracle
Version 19.1.0.0.4 OGGCORE_19.1.0.0.0_PLATFORMS_191017.1054_FBO
Linux, x64, 64bit (optimized), Oracle 19c on Oct 17 2019 21:16:29
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2019, Oracle and/or its affiliates. All rights reserved.

GGSCI (rob01db01.robdomain) 2> edit param mgr
-- Manager process file
PORT 7812
DYNAMICPORTLIST 7810-7815
-- Five ports is an advised minimum.
-- By naming it, you have better control and fix the numbers for the firewall security.
-- Set it to a higher number when for example change management process is long.
PURGEOLDEXTRACTS ./dirdat/*, USECHECKPOINTS, MINKEEPDAYS 2
AUTOSTART ER *
AUTORESTART extract *, retries 10, WAITMINUTES 5, RESETMINUTES 60
ACCESSRULE, PROG *, IPADDR 192.168.0.56, ALLOW
```

4.4.2. Start the manager process

```
GGSCI (rob01db01.robdomain) 2> start mgr
Manager started.
```

4.4.3. Check the manager process

```
GGSCI (rob01db01.robdomain) 3> info mgr
Manager is running (IP port TCP:rob01db01.robdomain.7812, Process ID 3576).
```

You can also review the ggserr.log file in \$GG_HOME:

```
-----
2020-09-17T17:36:45.191+0100 INFO OGG-00987 Oracle GoldenGate Command Interpreter for Oracle: GGSCI command (oracle): edit param mgr.
2020-09-17T17:38:03.637+0100 INFO OGG-00987 Oracle GoldenGate Command Interpreter for Oracle: GGSCI command (oracle): start mgr.
2020-09-17T17:38:03.934+0100 WARNING OGG-02286 Oracle GoldenGate Manager for Oracle, mgr.prm: Parameter port is incorrectly specified multiple times.
2020-09-17T17:38:03.934+0100 WARNING OGG-01877 Oracle GoldenGate Manager for Oracle, mgr.prm: Missing explicit accessrule for server collector.
2020-09-17T17:38:03.954+0100 INFO OGG-00983 Oracle GoldenGate Manager for Oracle, mgr.prm: Manager started (port 7812).
2020-09-17T17:38:39.649+0100 INFO OGG-00987 Oracle GoldenGate Command Interpreter for Oracle: GGSCI command (oracle): info mgr.
-----
```

You can check the report file \$GG_HOME/dirrpt/MGR.rpt for details of the process and process state.

4.5. Configure the extract process

4.5.1. Configure and Start the extract

Oracle Multitenant only supports the integrated Extract, which is far better anyway than the classical extract. In integrated capture mode, the GoldenGate Extract process no longer reads the online redo log files (as is the case with the classical capture mode), but instead, the GoldenGate Extract process interacts directly with a database logmining server to receive data changes in the form of logical change records (LCR).

Create a directory for the extract:

```
[oracle@rob01db01 dirrpt]$ mkdir -p $GG_HOME/dirdat/pdb4
```

Create a parameter file for the extract:

```
GGSCI (rob01db01.robdomain) 1> edit param xpb4

extract xpdb4
USERIDALIAS c##_gg_admin_root
-- set the environment
SETENV (ORACLE_SID='orcl')
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
SOURCECATALOG pdb4
discardfile ./dirrpt/pdb4.dsc, purge
exttrail ./dirdat/pdb4/ex
-- include DDL
DDL INCLUDE MAPPED
DDLOPTIONS REPORT
LOGALLSUPCOLS
UPDATERECORDFORMAT COMPACT
TRANLOGOPTIONS INTEGRATEDPARAMS (MAX_SGA_SIZE 1024)
-- replication objects
-- This is how to exclude a particular table from a schema
-- tableexclude hr.tablename
-- this rule must be before the "table" rule.
table robla.*;
-- sequence hr.*;
-- list how many records are processed every half an hour
reportcount every 30 minutes, rate
```

The one new parameter when replicating PDBs is the SOURCECATALOG parameter. The SOURCECATALOG identifies what PDB to replicate. A single extract is able to replicate many PDBs so by adding additional SOURCECATALOG and TABLE sections additional PDBs can be replicated.

We can see by the dbalias that we need to connect to the root container for the extract process. Otherwise the below error get's thrown:

```
2020-09-18 16:55:59 ERROR OGG-06206 The database connection must be to the root level database for user c##gg_admin.
```

=== Hands On Lab : Oracle GoldenGate 19c & Multitenancy ===

We need to connect to the root container when adding and registering the extract.

```
GGSCI (rob01db01.robdomain) 1> dblogin user=alias c##gg_admin_root
Successfully logged into database CDB$ROOT.

GGSCI (rob01db01.robdomain as c##gg_admin@orcl/CDB$ROOT) 2> register extract xpdb4 database container
(pdb4)

2020-09-17 19:46:21 INFO OGG-02003 Extract XPDB4 successfully registered with database at SCN
4232902.

GGSCI (rob01db01.robdomain as c##gg_admin@orcl/CDB$ROOT) 3> add extract xpdb4, integrated tranlog,
begin now
EXTRACT (Integrated) added.

GGSCI (rob01db01.robdomain as c##gg_admin@orcl/CDB$ROOT) 4> ADD EXTTRAIL ./dirdat/pdb4/ex, EXTRACT
xpdb4 megabytes 10
EXTTRAIL added.

GGSCI (rob01db01.robdomain as c##gg_admin@orcl/CDB$ROOT) 5> start extract xpdb4

Sending START request to MANAGER ...
EXTRACT XPDB4 starting
```

4.5.2. Check the extract

```
GGSCI (rob01db01.robdomain as c##gg_admin@orcl/CDB$ROOT) 4> info extract xpdb4

EXTRACT      XPDB4      Last Started 2020-09-18 18:41      Status RUNNING
Checkpoint Lag      00:00:00 (updated 00:00:01 ago)
Process ID          20615
Log Read Checkpoint Oracle Integrated Redo Logs
                    2020-09-18 18:43:38
                    SCN 0.4548558 (4548558)
```

Also check the extract report file:

```
$ cd $GG_HOME/dirrpt

[oracle@rob01db01 dirrpt]$ tail -f XPDB4.rpt
*****

2020-09-18 18:41:48 INFO OGG-06508 Wildcard MAP (TABLE) resolved (entry pdb4.robla.*): table
"PDB4"."ROBLA"."RL_HEARTBEAT".

2020-09-18 18:41:49 INFO OGG-06509 Using the following key columns for source table
PDB4.ROBLA.RL_HEARTBEAT: ID.

2020-09-18 18:41:49 INFO OGG-06508 Wildcard MAP (TABLE) resolved (entry pdb4.robla.*): table
"PDB4"."ROBLA"."RL_HEARTBEAT".

2020-09-18 18:41:49 INFO OGG-06509 Using the following key columns for source table
PDB4.ROBLA.RL_HEARTBEAT: ID.
```

=== Hands On Lab : Oracle GoldenGate 19c & Multitenancy ===

Also check if the trail files are being generated in the designated directory

```
[oracle@rob01db01 19.1.0]$ ls -altr $GG_HOME/dirdat/pdb4
total 96
drwxr-x--- 3 oracle oinstall  18 Sep 18 19:51 ..
-rw-r----- 1 oracle oinstall 33929 Sep 19 13:57 ex000000000
drwxr-xr-x 2 oracle oinstall  44 Sep 19 13:57 .
-rw-r----- 1 oracle oinstall 58211 Sep 19 15:05 ex000000001
```

4.6. Configure the Pump process

The Data pump process is an optional process but Oracle Best Practices recommend this setup. The Pump process reads the local trail files and ship them over to the Collector process on the target server, which writes the remote trail file. The Pump process is basically just like the Extract process, but instead of reading the online redo log files the Pump process reads the local trail files. Note that the Pump process does not wait until the trail file is full, before processing the file, it pumps over transactions as we go along (much like real time apply in Oracle Dataguard).

To increase performance of the Data Pump when table names and table structures are not altered or data is being filtered, use the PASSTHRU parameter in the Data Pump parameter file. This prevents the Data Pump from looking up table definitions from either the database or from a data definitions file. The PASSTHRU parameter is table specific and can be used with a wildcard to apply to multiple tables. Tables listed with the NOPASSTHRU parameter must be specified after the PASSTHRU parameter. Doing this increases Data Pump performance and reduces CPU usage.

4.6.1. Create a directory for the remote trail file:

```
[oracle@rob01db01 dirdat]$ mkdir -p /u01/app/oracle/product/ogg/19.1.0/dirdat/pdb5
```

4.6.2. Create a pump parameter file

```
[oracle@rob01db01 dirdat]$ cd $GG_HOME
[oracle@rob01db01 19.1.0]$ ./ggsci

Oracle GoldenGate Command Interpreter for Oracle
Version 19.1.0.0.4 OGGCORE_19.1.0.0.0_PLATFORMS_191017.1054_FBO
Linux, x64, 64bit (optimized), Oracle 19c on Oct 17 2019 21:16:29
Operating system character set identified as UTF-8.

Copyright (C) 1995, 2019, Oracle and/or its affiliates. All rights reserved.

GGSCI (rob01db01.robdomain) 1> edit param ppdb4
```

=== Hands On Lab : Oracle GoldenGate 19c & Multitenancy ===

Enter the following parameters:

```
EXTRACT ppdb4
-- PASSTRHU indicates that the data does not need to be filtered or transformed
PASSTRHU
-- RMTTHOST points to the remote system
RMTTHOST rob01db01, MGRPORT 7812, TCPBUFSIZE 1048576, TCPFLUSHBYTES 1048576
-- RMTTRAIL is the remote trail on the target database server
RMTTRAIL dirdat/pdb5/px
DDL INCLUDE MAPPED
-- objects that are replicated to the target system
table rob1a.*;
REPORTCOUNT EVERY 15 MINUTES, RATE
```

Remark: to find out the required port, you can query the manager process:

```
GGSCI (rob01db01.robdomain) 2> info mgr
Manager is running (IP port TCP:rob01db01.robdomain.7812, Process ID 19848).
```

4.6.3. Register and Start the pump process

```
GGSCI (rob01db01.robdomain) 3> dblogin user=alias c##gg_admin_root
Successfully logged into database CDB$ROOT.

GGSCI (rob01db01.robdomain as c##gg_admin@orcl/CDB$ROOT) 5> register extract ppdb4 database container (pdb4)
2020-09-19 15:19:08 INFO OGG-02003 Extract PPDB4 successfully registered with database at SCN 4804291.

GGSCI (rob01db01.robdomain as c##gg_admin@orcl/CDB$ROOT) 6> add extract ppdb4, extrailsources dirdat/pdb4/ex
EXTRACT added.

GGSCI (rob01db01.robdomain as c##gg_admin@orcl/CDB$ROOT) 7> add rmttrail dirdat/pdb5/px, extract ppdb4, megabytes 10
RMTTRAIL added.

GGSCI (rob01db01.robdomain as c##gg_admin@orcl/CDB$ROOT) 8> start extract ppdb4
Sending START request to MANAGER ...
EXTRACT PPDB4 starting
```


4.6.4. Verify the pump process

```
GGSCI (rob01db01.robdomain as c##gg_admin@orcl/CDB$ROOT) 4> info extract ppdb4
EXTRACT      PPDB4      Last Started 2020-09-19 15:33      Status RUNNING
Checkpoint Lag      00:00:00 (updated 00:00:07 ago)
Process ID         26060
Log Read Checkpoint File /u01/app/oracle/product/ogg/19.1.0/dirdat/pdb4/ex000000001
                  2020-09-19 17:23:26.000000 RBA 91970
```

You can also view the remote trail files. Note that the remote trailfiles do not have the same sequence numbers as the local trailfiles necessarily. There is no 1:1 relationship.

```
[oracle@rob01db01 19.1.0]$ cd $GG_HOME/dirdat/pdb5
[oracle@rob01db01 pdb5]$ ls -ltr
total 124
-rw-r----- 1 oracle oinstall 124560 Sep 19 17:25 px000000000
```

And you can view the pump report file in \$GG_HOME/dirrpt:

```
2020-09-19 15:33:05 INFO      OGG-02243  Opened trail file
/u01/app/oracle/product/ogg/19.1.0/dirdat/pdb4/ex000000000 at 2020-09-19 15:33:05.641606.
2020-09-19 15:33:05 INFO      OGG-01478  Output file dirdat/pdb5/px is using format RELEASE 19.1.
2020-09-19 15:33:05 INFO      OGG-02262  Passthru wildcard MAP (TABLE) resolved (entry pdb4.rob1a.):
table "PDB4"."ROBLA"."RL_HEARTBEAT".
2020-09-19 15:33:05 INFO      OGG-02232  Switching to next trail file
/u01/app/oracle/product/ogg/19.1.0/dirdat/pdb4/ex000000001 at 2020-09-19 15:33:05.658029 due to EOF.
with current RBA 33,929.
2020-09-19 15:33:35 INFO      OGG-01971  The previous message, 'INFO OGG-02262', repeated 1 times.
2020-09-19 15:48:29 INFO      OGG-08549  402 records processed as of 2020-09-19 15:48:29.461837 (rate 0,
delta 0).
2020-09-19 16:04:31 INFO      OGG-08549  418 records processed as of 2020-09-19 16:04:31.265097 (rate 0,
delta 0).
2020-09-19 16:20:30 INFO      OGG-08549  434 records processed as of 2020-09-19 16:20:30.113564 (rate 0,
delta 0).
2020-09-19 16:36:29 INFO      OGG-08549  450 records processed as of 2020-09-19 16:36:29.914585 (rate 0,
delta 0).
2020-09-19 16:51:30 INFO      OGG-08549  465 records processed as of 2020-09-19 16:51:30.663370 (rate 0,
delta 0).
2020-09-19 17:07:31 INFO      OGG-08549  481 records processed as of 2020-09-19 17:07:31.483909 (rate 0,
delta 0).
2020-09-19 17:23:29 INFO      OGG-08549  497 records processed as of 2020-09-19 17:23:29.303634 (rate 0,
delta 0).
```

4.7. Instantiate the tabl/schema in the target database

There are several ways to perform the target instantiation. Please refer to document Oracle GoldenGate Best Practices: Instantiation from an Oracle Source Database (Doc ID 1276058.1)

I will use Oracle Datapump for the target instantiation. This is an easy and the fastest way.

The export can be performed directly on the production system by using the export with the FLASHBACK_SCN option. Then the FLASHBACK_SCN used as the GoldenGate CSN value used for the Replicat. Note that you have to take the entire export using the same value for FLASHBACK_SCN for your entire export, even if you use multiple export files (e.g. you run multiple sessions in parallel, or in the case of Data Pump Export, you use Oracle's parallelism.). However, in this example, only 1 table will need to be instantiated.

4.7.1. Check the current SCN in the source database

```
SQL> col current_scn format 999999999999
SQL> select current_scn from v$database;

CURRENT_SCN
-----
          5007688
```

4.7.2. Create a directory for the export

```
SQL> create directory dir_pdb4 as '/home/oracle/pdb4';
SQL> !mkdir /home/oracle/pdb4
```

4.7.3. Perform the export

```
Starting "C##GG_ADMIN"."SYS_EXPORT_TABLE_01": c##gg_admin/*****@pdb4 directory=dir_pdb4 parallel=1
dumpfile=exp_pdb4_4.dmp flashback_scn=5007688 logfile=exp_pdb4_4.log tables=robla.rl_heartbeat
exclude=TRIGGER
Processing object type TABLE_EXPORT/TABLE/TABLE_DATA
Processing object type TABLE_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type TABLE_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
Processing object type TABLE_EXPORT/TABLE/STATISTICS/MARKER
Processing object type TABLE_EXPORT/TABLE/PROCACT_INSTANCE
Processing object type TABLE_EXPORT/TABLE/TABLE
Processing object type TABLE_EXPORT/TABLE/CONSTRAINT/CONSTRAINT
.. exported "ROBLA"."RL_HEARTBEAT":"SYS_P268"          67.01 KB      2790 rows
.. exported "ROBLA"."RL_HEARTBEAT":"PART_01"          7.187 KB         0 rows
Master table "C##GG_ADMIN"."SYS_EXPORT_TABLE_01" successfully loaded/unloaded
*****
Dump file set for C##GG_ADMIN.SYS_EXPORT_TABLE_01 is:
/home/oracle/pdb4/exp_pdb4_4.dmp
Job "C##GG_ADMIN"."SYS_EXPORT_TABLE_01" successfully completed at Sun Sep 20 11:55:11 2020 elapsed 0
00:00:51
```

=== Hands On Lab : Oracle GoldenGate 19c & Multitenancy ===

4.7.4. Import the table in the target database

We do not need to transfer the dumpfile because we are running it on the same system. Perform the import as follows.

First create the directory and the user in the target PDB:

```
SQL> create directory dir_pdb4 as '/home/oracle/pdb4';
SQL> create user robla identified by robla;
SQL> grant dba to robla
```

Now import the dumpfile:

```
[oracle@rob01db01 19.1.0]$ impdp c##gg_admin/gg_admin@pdb5 directory=dir_pdb4 dumpfile=exp_pdb4_4.dmp
logfile=imp_pdb4_4.log tables=robla.rl_heartbeat

Import: Release 19.0.0.0.0 - Production on Sun Sep 20 12:05:55 2020
Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Master table "C##GG_ADMIN"."SYS_IMPORT_TABLE_01" successfully loaded/unloaded
Starting "C##GG_ADMIN"."SYS_IMPORT_TABLE_01": c##gg_admin/*****@pdb5 directory=dir_pdb4
dumpfile=exp_pdb4_4.dmp logfile=imp_pdb4_4.log tables=robla.rl_heartbeat
Processing object type TABLE_EXPORT/TABLE/PROCACT_INSTANCE
Processing object type TABLE_EXPORT/TABLE/TABLE
Processing object type TABLE_EXPORT/TABLE/TABLE_DATA
.. imported "ROBLA"."RL_HEARTBEAT": "SYS_P268" 67.01 KB 2790 rows
.. imported "ROBLA"."RL_HEARTBEAT": "PART_01" 7.187 KB 0 rows
Processing object type TABLE_EXPORT/TABLE/CONSTRAINT/CONSTRAINT
Processing object type TABLE_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type TABLE_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
Processing object type TABLE_EXPORT/TABLE/STATISTICS/MARKER
Job "C##GG_ADMIN"."SYS_IMPORT_TABLE_01" successfully completed at Sun Sep 20 12:06:11 2020 elapsed 0
00:00:14
```

4.8. Configure the replicat process

We need to configure integrated replicat for the target database pdb5. This also offers some benefits:

- Integrated Replicat leverages the apply process functionality that is available inside the database. Referential integrity and DDL operations are automatically applied in the correct order. This alleviates the database administrator from having to understand how to partition tables between Replicat processes based on foreign key constraints, or from having to ensure that the correct Replicat handles the DDL for tables.
- Integrated Replicat also offers automatic parallelism which automatically increases or decreases the number of apply processes based on the current workload and database performance. Management and tuning of Replicat performance is simplified since you do not have to manually configure multiple Replicat processes to distribute the tables between them.
- Integrated Replicat automatically enables the asynchronous commit feature so processing can continue immediately after each COMMIT command is issued. The Replicat process will read all the changes in the remote trail files and will apply those changes to the target database.
- The Replicat process maintains checkpoints that provide a known position in the trail file for recovery and restart. By default, this checkpoint information is stored in a checkpoint file for the Replicat process.
- Integrated Replicat automatically records checkpoint information within the target database so checkpoint table creation is not necessary.
- Integrated Replicat uses dynamic allocation of apply server processes to distribute the work of applying transactions between the processes. New apply processes are created or removed dynamically to maintain the optimal Replicat throughput. This automation reduces the administrative overhead of manually dividing the work between Replicat processes. Oracle recommends to first configure a single Replicat process and monitor apply lag and performance. Adding more Replicat processes should only be carried out when performance of a single Replicat causes unacceptable apply latency.

4.8.1. Configure the replicat parameter file

```
GGSCI (rob01db01.robdomain) 5> edit param rpdb4
```

Enter the following parameter settings:

```
REPLICAT rpdb4
USERIDALIAS c##gg_admin_pdb5
-- set the environment
SETENV (ORACLE_SID='orcl')
SOURCECATALOG pdb5
SETENV (NLS_LANG=AMERICAN_AMERICA.AL32UTF8)
HANDLECOLLISIONS
DBOPTIONS SUPPRESSTRIGGERS
ASSUMETARGETDEFS
DISCARDFILE ./dirrpt/pdb5.dsc, purge
DDL INCLUDE MAPPED
DDLOPTIONS REPORT
DBOPTIONS INTEGRATEDPARAMS(parallelism 4)
-- To speed up replication add "batchsql"
-- See the GG Reference Guide for more info, options and possible restrictions.
-- BATCHSQL
-- list how many records are processed every half an hour
REPORTCOUNT EVERY 30 MINUTES, RATE
MAP pdb4.rob1a.*, TARGET pdb5.rob1a.*;
```

Remark: the Replicat process is not allowed to log in to the root container, so here we use the alias to log in to the target pdb5.

4.8.2. Create and start the replicat

```
GGSCI (rob01db01.robdomain as c##gg_admin@orcl/CDB$ROOT) 3> register replicat rpdb4 database
2020-09-20 12:17:30 INFO OGG-02528 REPLICAT RPDB4 successfully registered with database as inbound
server OGG$RPDB4.

GGSCI (rob01db01.robdomain as c##gg_admin@orcl/CDB$ROOT) 4> add replicat rpdb4, integrated, exttrail
./dirdat/pdb5/px
REPLICAT (Integrated) added.

GGSCI (rob01db01.robdomain as c##gg_admin@orcl/CDB$ROOT) 5> start replicat rpdb4, aftercsn 5007688
Sending START request to MANAGER ...
REPLICAT RPDB4 starting
```

4.8.3. Check the replicat process

Check the replicat process itself

```
GGSCI (rob01db01.robdomain as c##gg_admin@orcl/PDB5) 6> info replicat rpdb4
REPLICAT RPDB4 Last Started 2020-09-20 12:24 Status RUNNING
INTEGRATED
Checkpoint Lag 00:00:00 (updated 00:00:00 ago)
Process ID 24269
Log Read Checkpoint File /u01/app/oracle/product/ogg/19.1.0/dirdat/pdb5/px000000000
2020-09-20 12:24:26.001018 RBA 212661
```

Check the replicat report file

```
. . . .
. . . .
2020-09-20 12:24:13 INFO OGG-06604 Database PDB5.ROBDOMAIN CPU info: CPU Count 1, CPU Core Count
1, CPU Socket Count 1.
2020-09-20 12:24:13 WARNING OGG-05673 CSN-based duplicate suppression is disabled because there is no
checkpoint table for this Replicat.
2020-09-20 12:24:13 INFO OGG-02545 Parameter GROUPTRANSOPS is ignored by Integrated Replicat when
parallelism is greater than 1.
2020-09-20 12:24:13 INFO OGG-02527 Integrated Replicat does not populate a trace table.
2020-09-20 12:24:13 INFO OGG-06627 Replicat sets the key columns for each table replicated by
inbound server OGG$RPDB4.
2020-09-20 12:24:15 INFO OGG-02530 Integrated replicat successfully attached to inbound server
OGG$RPDB4.
*****
** Run Time Messages **
*****
2020-09-20 12:24:16 INFO OGG-02243 Opened trail file
/u01/app/oracle/product/ogg/19.1.0/dirdat/pdb5/px000000000 at 2020-09-20 12:24:16.184983.
```

=== Hands On Lab : Oracle GoldenGate 19c & Multitenancy ===

```
2020-09-20 12:24:16 WARNING OGG-02760 ASSUMETARGETDEFS is ignored because trail file
/u01/app/oracle/product/ogg/19.1.0/dirdat/pdb5/px000000000 contains table definitions.

2020-09-20 12:24:16 INFO OGG-03506 The source database character set, as determined from the trail
file, is UTF-8.

2020-09-20 12:24:19 INFO OGG-01373 User requested start after CSN 5007688.

2020-09-20 12:24:19 INFO OGG-01020 Processed extract process RESTART_ABEND record at seq 0, rba
34001 (aborted 0 records).

2020-09-20 12:24:19 INFO OGG-01374 Transaction delivery commencing at position Seqno 0, RBA
200188, Transaction ID 1.7.1408, CSN 5008297, 808 transaction(s) skipped.

2020-09-20 12:24:19 INFO OGG-06506 Wildcard MAP resolved (entry pdb4.robld.*): MAP
"PDB4"."ROBLA"."RL_HEARTBEAT", TARGET pdb5.robld."RL_HEARTBEAT".

2020-09-20 12:24:43 INFO OGG-02756 The definition for table PDB4.ROBLA.RL_HEARTBEAT is obtained
from the trail file.

2020-09-20 12:24:43 INFO OGG-06511 Using following columns in default map by name: ID, COLOUR,
INSERT_DATE_SOURCE, INSERT_DATE_TARGET, UPDATE_DATE_SOURCE, UPDATE_DATE_TARGET.

2020-09-20 12:24:43 INFO OGG-06510 Using the following key columns for target table
PDB5.ROBLA.RL_HEARTBEAT: ID.
```

Check the RL_HEARTBEAT_TABLE in the target database pdb5

```
SQL> alter session set nls_date_format='DD-MON-YYYY HH24:MI:SS';
SQL> show con_name

CON_NAME
-----
PDB5

SQL> select sysdate from dual;

SYSDATE
-----
20-SEP-2020 12:33:39

SQL> select * from (
select id, colour, insert_date_source from robld.rl_heartbeat order by insert_date_source desc)
where rownum < 11;

      ID COLOUR      INSERT_DATE_SOURCE
-----
2910 GREEN      20-SEP-2020 12:33:25
2909 GREEN      20-SEP-2020 12:32:25
2908 GREEN      20-SEP-2020 12:31:25
2907 GREEN      20-SEP-2020 12:30:26
2906 GREEN      20-SEP-2020 12:29:25
2905 GREEN      20-SEP-2020 12:28:25
2904 GREEN      20-SEP-2020 12:27:25
2903 GREEN      20-SEP-2020 12:26:25
2902 GREEN      20-SEP-2020 12:25:25
2901 GREEN      20-SEP-2020 12:24:26

10 rows selected.
```

So, we can see that the table robld.rl_heartbeat contains the latest records and that we are connected to the target database. (My heartbeat table contains some more records which I use to test DML and timing lag between source and target, but that is outside the scope of this lab)

5. Encountered errors during the setup

Error

OGG-06206 The database connection must be to the root level database for user c##gg_admin.

Analysis

Connection in the extract parameter file was made to pdb4, but should be made to cdb\$root

Solution:

use the alias c##gg_admin_root in the extract parameter file in stead of alias c##gg_admin_pdb4

Error

OGG-02050 Not enough database memory to honor requested MAX_SGA_SIZE of 1024.

Analysis

A bit strange. The actual sga_max_size is set higher than reported in the error message:

```
SQL> show parameter sga_max_size
```

NAME	TYPE	VALUE
sga_max_size	big integer	1184M

However, I did set the streams_pool_size to 300M and that get's eaten away from the SGA. So if I limit the streams_pool_size to 100M, that could resolve it. (Alternatively, I could increase both SGA_MAX_SIZE and SGA_TARGET, but I do not have too much memory on my host machine).

Solution:

Decrease the streams_pool_size to 100M

```
SQL> alter system set streams_pool_size=100M;
```

System altered.

=== Hands On Lab : Oracle GoldenGate 19c & Multitenancy ===

Error:

OGG-01044 The trail './dirdat/prd4/ex' is not assigned to extract 'XPDB4'. Assign the trail to the extract with the command "ADD EXTTRAIL/RMTTRAIL ./dirdat/prd4/ex, EXTRACT XPDB4".

Analysis:

I issued the command before. But I am not sure if the order was right in my command. I read a slightly different order on website <https://oracledb101.wordpress.com/2014/03/26/replicating-pdbs-with-goldengate-12c/>

Solution:

```
Ggsci> dblogin useridentialias c##gg_admin_root
GGSCI (rob01db01.robdomain as c##gg_admin@orcl/CDB$ROOT) 2> ADD EXTTRAIL ./dirdat/prd4/ex, EXTRACT
XPDB4 megabytes 10
EXTTRAIL added.
GGSCI (rob01db01.robdomain as c##gg_admin@orcl/CDB$ROOT) 3> start extract xpdb4
Sending START request to MANAGER ...
EXTRACT XPDB4 starting
```