Using Oracle GoldenGate in a Multitenant Container Database

Practice Overview

In this practice, you will implement two tasks at the same time:

- Configure an Oracle GoldenGate 19c replication between an Oracle 12c PDB and an Oracle 19c PDB, including tables and sequences.
- Use Oracle GoldenGate for database upgrade with nearly zero downtime.

Pre-requisites

To implement this practice, we need to have the following two virtual machines (vms):

- **Source database:** is an Oracle 12c CDB database on Linux x86-64. It has one user PDB.
- **Destination database:** is an Oracle 19c CDB database on Linux x86-64. It has one user PDB.

To have those machines, you have three options:

- Build the machines from scratch
- Use the existing course environment by installing Oracle 19c in the second vm (ggsrv2).
- Download the vms from my website. I made an environment available for download from my website. Visit the link https://ahmedbaraka.com/public/download/ and download the file available in the section labelled as "**Oracle GoldenGate 19c on Two Oracle Databases**".
  The file size is nearly **21GB**. Therefore, it will take some time to download it, if you have a slow connection to the Internet.

Regardless of the way you chose to build up the environment, ultimately you should have an environment with the following specifications:

VM	Specification	Value
ogg1 (source)	Hostname	ogg1
	OS	Linux 7
	Database release	12c R2
	GoldenGate release	19c
	Memory	4GB (6Gb recommended)
	Archivelog Mode	Enabled (*)
ogg2 (destination)	Hostname	ogg2
	OS	Linux 7
	Database release	19c
	GoldenGate release	19c
	Memory	4GB It is recommended to increase it to at least 6 GB
	Archivelog Mode	Enabled (*)

(*) In the downloadable environment, there is a `cron` job which runs every 2nd hour to clean up the archivelogs.

Preparing the Environment

Assuming that you have the environment as mentioned in the "**pre-requisites**" section, in this section of the practice, you will make changes to prepare the environment for this practice.

A. Update Network Settings

If you downloaded the environment from my website, you need to update the network settings of each appliance because the network adapter name is different from its name in your machine.

1. Make sure the `ogg1` and `ogg2` appliances are turned off. If not, shut them down.

2. In the VirtualBox window, for **each appliance**, perform the following:

Click on "**Settings**" button (`[Ctl]+[s]`), "**Network**", from the **Name** drop list, select the adapter that matches your environment, click on **OK** button.

B. Enabling "Shared Clipboard" settings

Note: If you are using VMs from my website, do not implement the steps in this section. They are already implemented in the VMs.

3. Make sure the `ogg1` and `ogg2` appliances are turned off. If not, shut them down.

4. In the VirtualBox window, for **each appliance**, perform the following:

Click on "**Settings**" button, "**General**", "**Advanced**" tab, and set the "**Shared Clipboard**" to "**Bidirectional**", click on **OK** button.

C. (optional) Creating and configuring a staging directory

Note: If you are using VMs from my website, do not implement the steps in this section. They are already implemented in the VMs.

In the following steps you will create a staging directory in the hosting PC. This directory will be accessed by the VirtualBox appliances ogg1 and ogg2. This task is not needed for this practice but it might be useful for future needs.

5. In your hosting machine, under a disk drive letter of your choice, create a staging directory.

6. In the **VirtualBox Manager**, for **each appliance**, open the "**Settings**", click on "**Shared Folders**" link in the right-hand pane. Add shared folder by pressing "**plus**" icon. Then select the path to directory created in the preceding step, and mark the "**Auto-mount**" box. Change the "**Folder Name**" to "**staging**"

7. Perform the following steps to add oracle user to the vboxsf group. This group has privilege to access the shared folder.

   a. Start the appliance `ogg1`

   b. Start Putty and login as `root` to `ogg1`

   c. Open a terminal window and execute the following command to make sure the shared folder is seen by the appliance:

   ```
   ls -ld /media/sf_staging/
   ```

   d. Add `oracle` to `vboxsf` group.

   ```
   usermod -a -G vboxsf oracle
   ```

   e. Repeat the same steps for `ogg2`

D. Updating the Network Configuration

This task might be needed if you are using the VMs downloaded from my website.

The VMs are configured with static IP addresses. In the following steps you will verify that the network configuration works fine in your environment and make the necessary changes if needed.

8. In the VirtualBox window of ogg1, click on **Application** > **System Tools** > **Settings** > **Network** > under the **Wired** section, click on the **gear button** > click on the **IPv4** address tab

9. Check the static IP address and make sure it is acceptable in your network and pingable by your hosting PC. Modify the IP address, if needed. If you change the IP address, you must perform the following:

   a. In the network setting, under the Wired section, enable and disable the network card so that the new IP address takes effect.

   b. Fix the IP address of the vm in the `/etc/hosts` file.

   c. Fix the IP address of the vm in the `/etc/hosts` file in the other vms.

10. Repeat this section steps on `ogg2`

E. Installing Swingbench

Swingbench is a free Java-based benchmark application. You will use it as the sample application for this practice.

In the following steps, you will install Swingbench on your hosting PC.

11. Download **Swingbench 2.6** from one of the following sources: Course **downloadable resources** section OR Dominic Giles portal

12. In your **hosting PC**, copy the software zip file to the disk drive where you want to install the software.

13. Extract the zip file into the selected drive letter.

    If you use the installation file attached to the course, the application files should be extracted into a folder named as 'swingbench2.6'. This folder will be referred to as $SWINGHOME folder.

14. In the **hosting PC**, open a command prompt window and change the directory to $SWINGHOME\winbin

```
cd D:\swingbench2.6\winbin
```

If you are using the VMs from my website, the Swingbench schema (soe) is already installed in PDB1. In this case, skip the following steps. If the soe is not installed in your environment, perform the following step to install it:

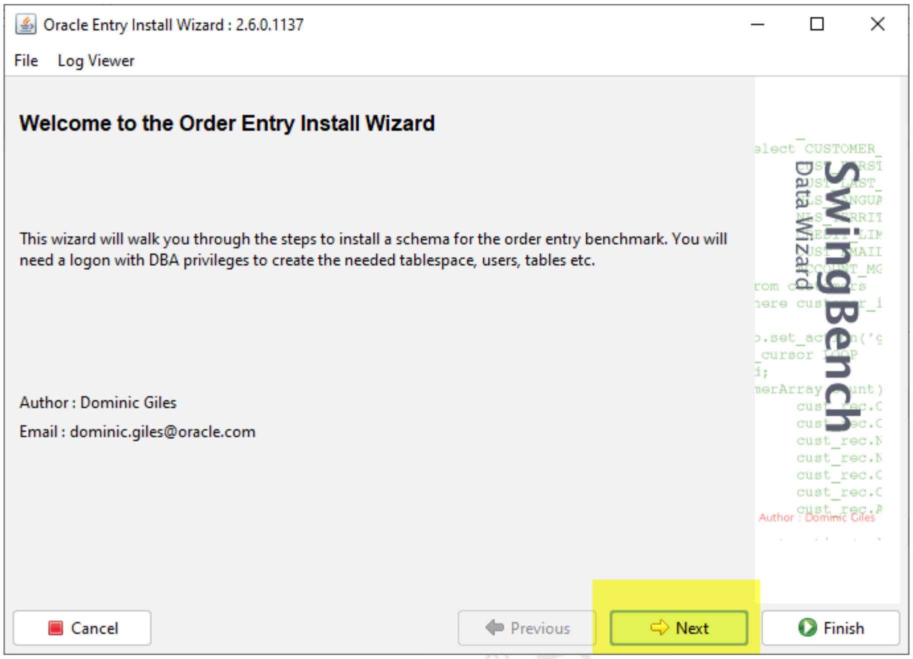
Note: Implement the following steps only if soe is not installed in the source database.

15. Start the "Order Entry Wizard"

```
oewizard.bat
```

Respond to the wizard windows as follows to create `soe` schema and load data into it.

After the wizard finishes its execution, you should see a message that informs all the objects are valid.

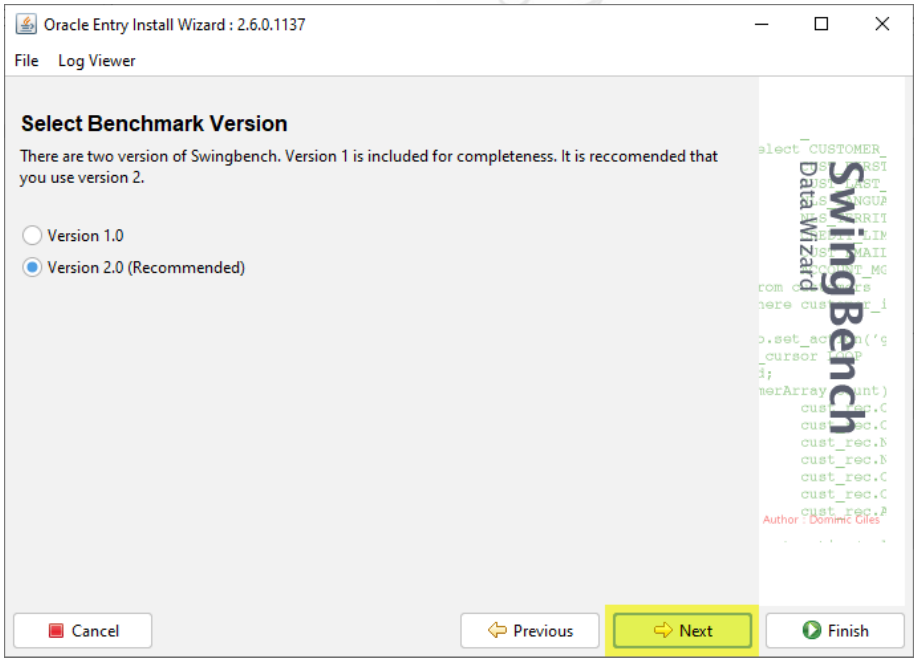


Welcome to the Order Entry Install Wizard

This wizard will walk you through the steps to install a schema for the order entry benchmark. You will need a logon with DBA privileges to create the needed tablespace, users, tables etc.

Author : Dominic Giles

Email : dominic.giles@oracle.com

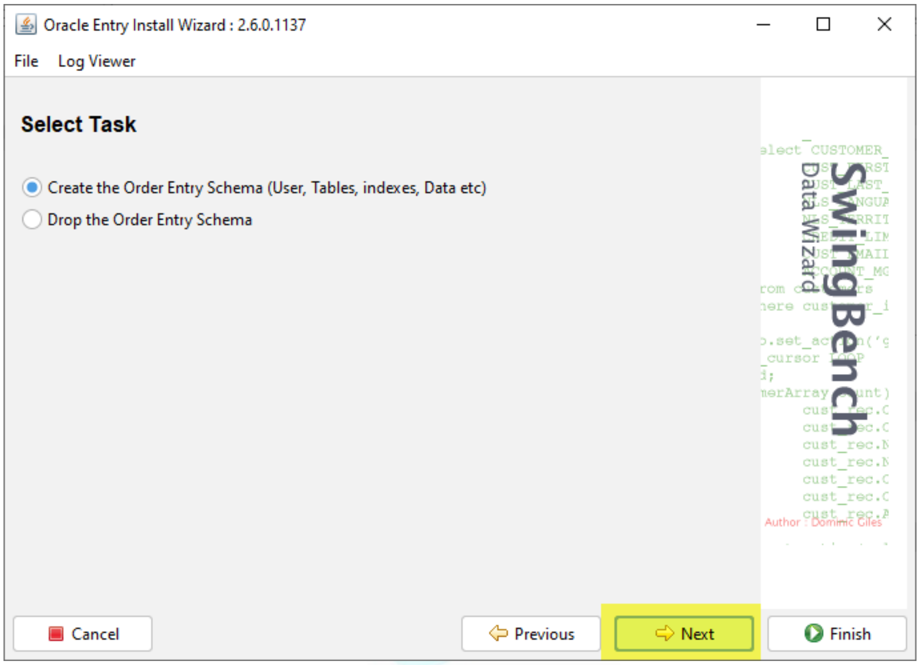


Select Benchmark Version

There are two version of Swingbench. Version 1 is included for completeness. It is reccomended that you use version 2.

○ Version 1.0

● Version 2.0 (Recommended)

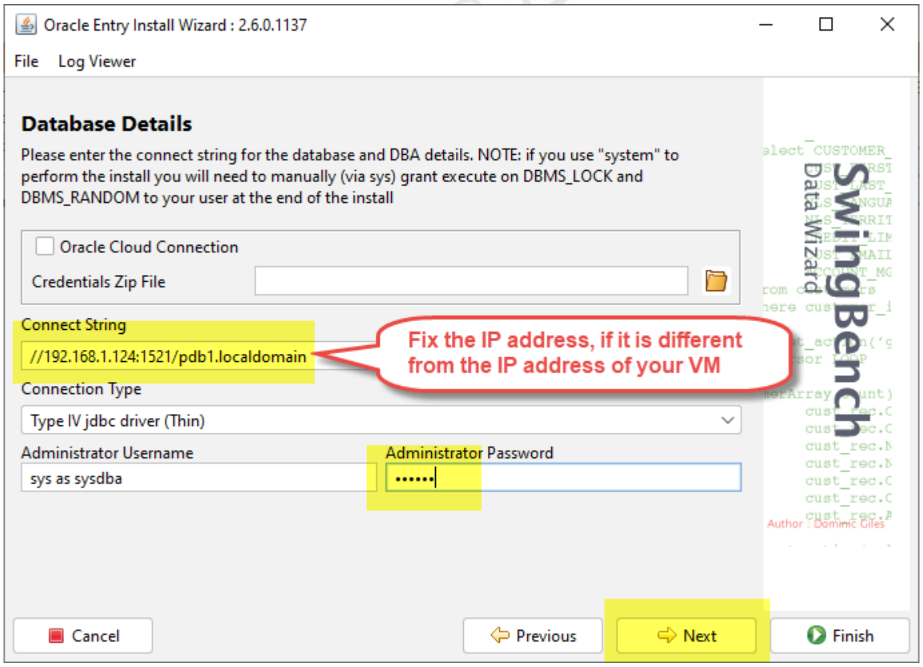


Oracle Entry Install Wizard : 2.6.0.1137

Select Task

● Create the Order Entry Schema (User, Tables, indexes, Data etc)

○ Drop the Order Entry Schema



Oracle Entry Install Wizard : 2.6.0.1137

Database Details

Please enter the connect string for the database and DBA details. NOTE: if you use "system" to perform the install you will need to manually (via sys) grant execute on DBMS_LOCK and DBMS_RANDOM to your user at the end of the install

□ Oracle Cloud Connection

Credentials Zip File:

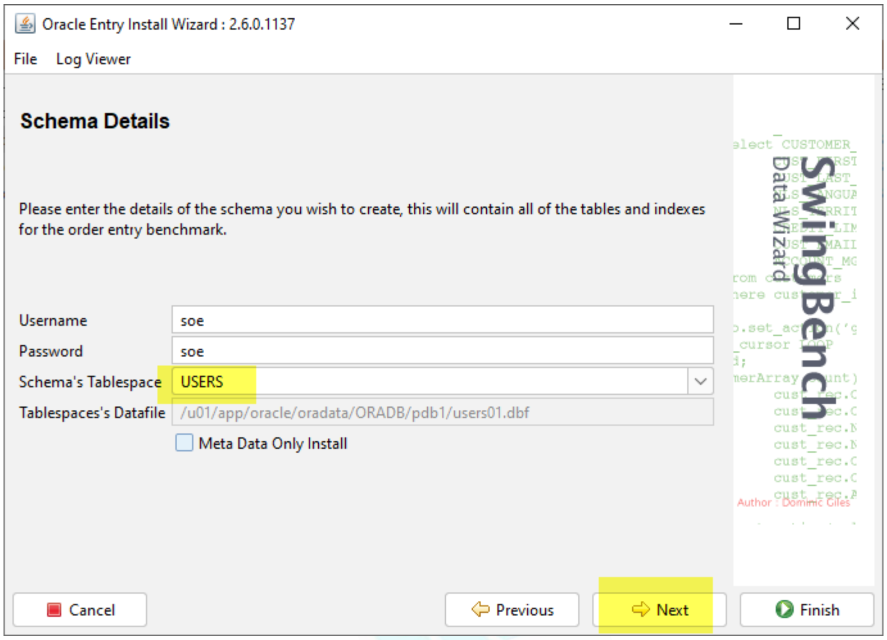
Connect String: //192.168.1.124:1521/pdb1.localdomain

Fix the IP address, if it is different from the IP address of your VM

Connection Type: Type IV jdbc driver (Thin)

Administrator Username: sys as sysdba

Administrator Password: ••••••



Oracle Entry Install Wizard : 2.6.0.1137

Schema Details

Please enter the details of the schema you wish to create, this will contain all of the tables and indexes for the order entry benchmark.

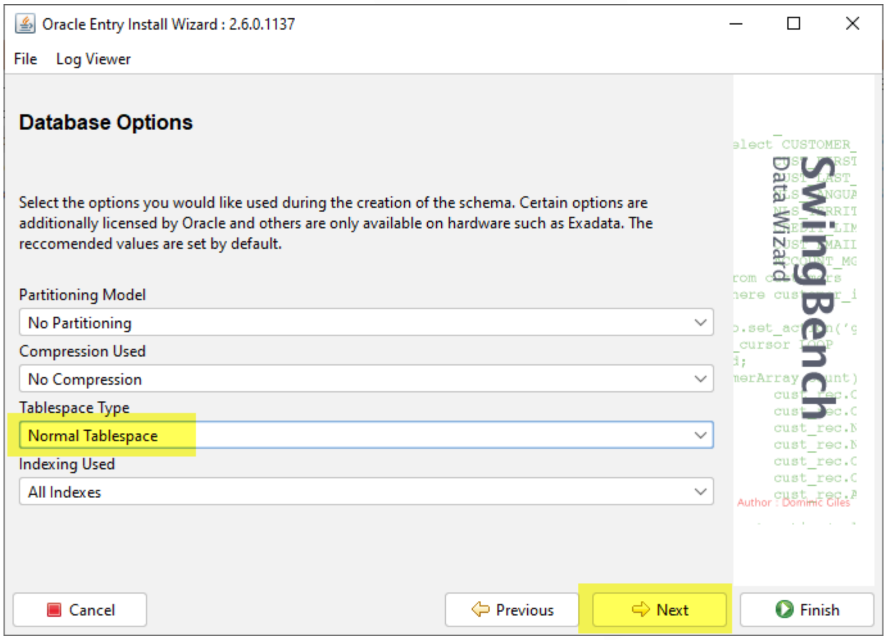
Username: soe

Password: soe

Schema's Tablespace: USERS

Tablespaces's Datafile: /u01/app/oracle/oradata/ORADB/pdb1/users01.dbf

□ Meta Data Only Install



Oracle Entry Install Wizard : 2.6.0.1137

Database Options

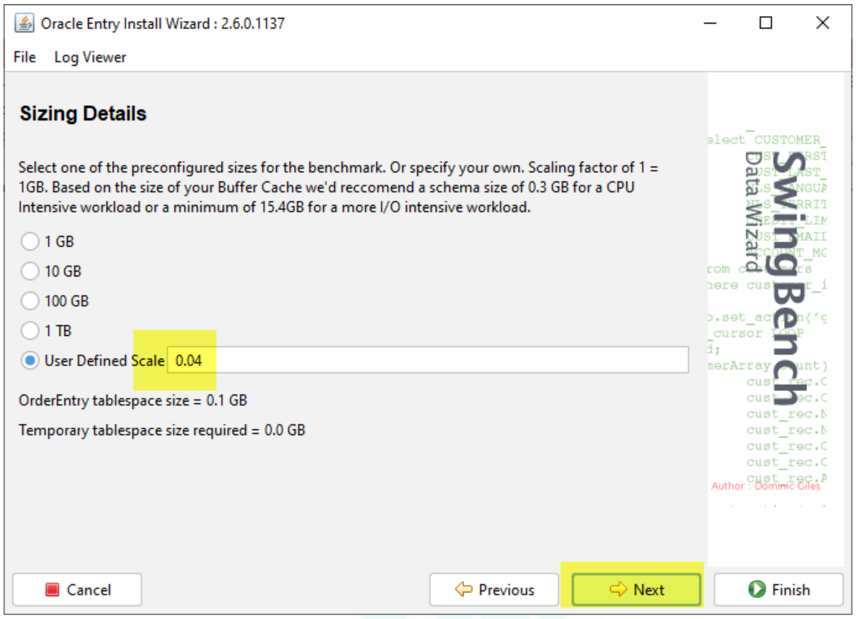
Select the options you would like used during the creation of the schema. Certain options are additionally licensed by Oracle and others are only available on hardware such as Exadata. The reccomended values are set by default.

Partitioning Model: No Partitioning

Compression Used: No Compression

Tablespace Type: Normal Tablespace

Indexing Used: All Indexes



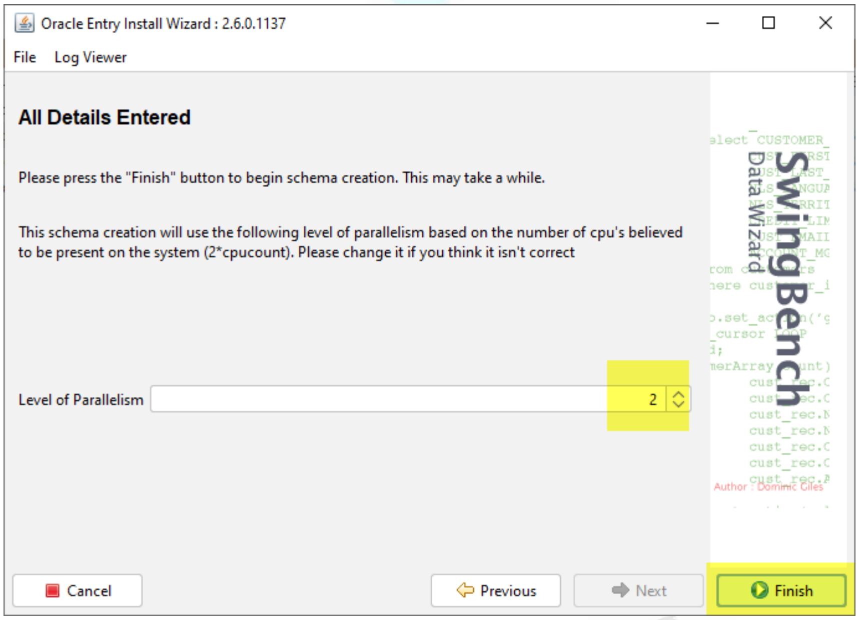
Sizing Details

Select one of the preconfigured sizes for the benchmark. Or specify your own. Scaling factor of 1 = 1GB. Based on the size of your Buffer Cache we'd reccomend a schema size of 0.3 GB for a CPU Intensive workload or a minimum of 15.4GB for a more I/O intensive workload.

○ 1 GB
○ 10 GB
○ 100 GB
○ 1 TB
● User Defined Scale 0.04

OrderEntry tablespace size = 0.1 GB

Temporary tablespace size required = 0.0 GB



All Details Entered

Please press the "Finish" button to begin schema creation. This may take a while.

This schema creation will use the following level of parallelism based on the number of cpu's believed to be present on the system (2*cpucount). Please change it if you think it isn't correct

Level of Parallelism 2

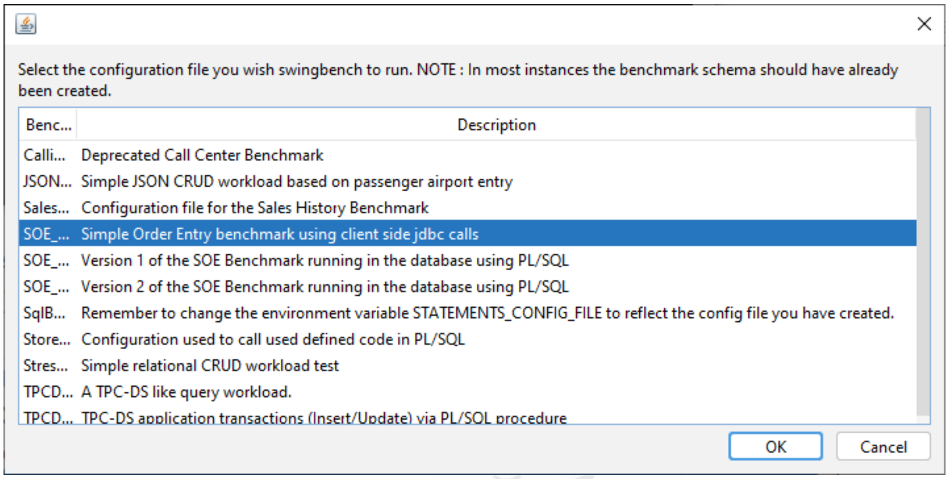
After installing the `soe` schema, we can invoke the Swingbench.

16. Start Swingbench by issuing the following commands:

The code below assumed that Java 8 for Windows **64-bit** is installed in the folder `C:\Program Files\Java\jre1.8.0_261`. If it is installed somewhere else in your PC, fix the code accordingly.

```
set PATH=C:\Program Files\Java\jre1.8.0_261\bin;%PATH%
swingbench.bat
```

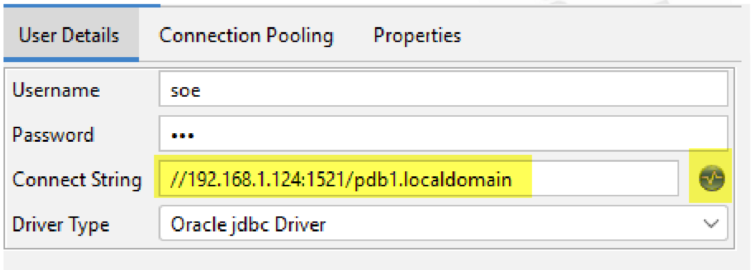
17. When you see the following window, select the option "**SOE_Client_Side**".



Select the configuration file you wish swingbench to run. NOTE : In most instances the benchmark schema should have already been created.

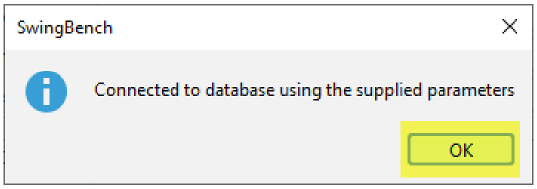
Benc...	Description
Calli...	Deprecated Call Center Benchmark
JSON...	Simple JSON CRUD workload based on passenger airport entry
Sales...	Configuration file for the Sales History Benchmark
SOE_...	Simple Order Entry benchmark using client side jdbc calls
SOE_...	Version 1 of the SOE Benchmark running in the database using PL/SQL
SOE_...	Version 2 of the SOE Benchmark running in the database using PL/SQL
SqlB...	Remember to change the environment variable STATEMENTS_CONFIG_FILE to reflect the config file you have created.
Store...	Configuration used to call used defined code in PL/SQL
Stres...	Simple relational CRUD workload test
TPCD...	A TPC-DS like query workload.
TPCD...	TPC-DS application transactions (Insert/Update) via PL/SQL procedure

18. Under the **User Details** tab, set the fields to the values as in the following screenshot. Click on **Test Connection** button.



Username	soe
Password	•••
Connect String	//192.168.1.124:1521/pdb1.localdomain
Driver Type	Oracle jdbc Driver

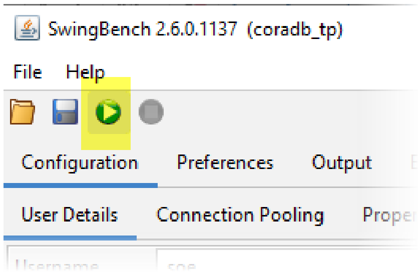
19. You should see the following successful message:



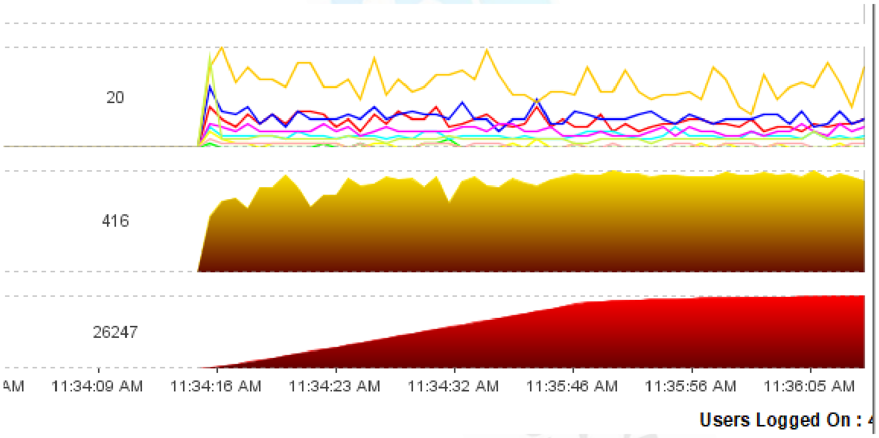
SwingBench

Connected to database using the supplied parameters

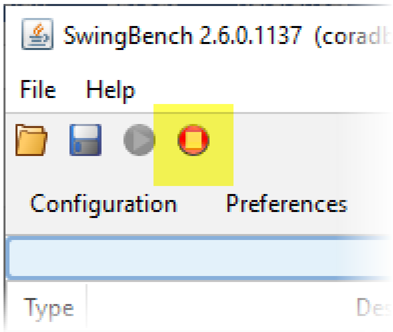
20. Click on **Start Benchmark Run** button:



21. Observe that the "**Transactions Per Minute**" chart is increasing by time and it eventually gets saturated.



22. Stop the Benchmark Run by clicking on its button.



23. Exit from Swingbench.

Configuring GoldenGate 19c Replication on PDB1

In this section of the practice, you will configure a uni-directional replication between the source and the destination systems.

F. Preparing the Environment for GoldenGate

24. In Oracle VirtualBox, take snapshots of the environment VMs.

Note: Please do not proceed without taking snapshots to your VMs. You will perform significant changes on the VMs. Snapshots will help you to quickly recover your VMs in case an issue is raised.

25. In `ogg1` and `ogg2`, invoke SQL*Plus from the GG home and login to the root container as sysdba.

```
cd $GG_HOME
sqlplus / as sysdba
```

26. Make sure that the parameter `STREAMS_POOL_SIZE` is configured.

If `SGA_TARGET` is set to a nonzero value and `STREAMS_POOL_SIZE` is not specified or is set to a null value, Automatic Shared Memory Management uses 0 (zero) bytes as a minimum for the `STREAMS_POOL_SIZE`. GoldenGate requires Steams Pool to be configured.

In real life scenario, use the following formula to calculate the value that you should set to `STREAMS_POOL_SIZE`:

(`MAX_SGA_SIZE` * # of integrated Extracts) + 25% head room

To set a value to this parameter, use the following command. You need to restart the database instance afterwards. In the pre-built environment, this parameter is set to 128 M.

```
ALTER SYSTEM SET STREAMS_POOL_SIZE=<value> SCOPE=SPFILE;
```

```
show parameter STREAMS_POOL_SIZE
```

27. Make sure `ENABLE_GOLDENGATE_REPLICATION` is enabled in each database. If it is not, enable it.

```
show parameter ENABLE_GOLDENGATE_REPLICATION

ALTER SYSTEM SET ENABLE_GOLDENGATE_REPLICATION=TRUE SCOPE=BOTH;
```

28. In `ogg1` and `ogg2`, create a database user that will be used by Oracle GoldenGate.

This is a common user that will have access to all the PDBs included in the replication in the source database. Sometimes it is referred to as the `c##` user.

```
CREATE USER c##ggadmin IDENTIFIED BY oracle DEFAULT TABLESPACE USERS TEMPORARY
TABLESPACE TEMP QUOTA UNLIMITED ON USERS;
```

29. In `ogg1`, grant the required privileges to the created user.

```
-- container is set to ALL, but it can be set to specific pdbs
BEGIN
 DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE ( GRANTEE => 'C##GGADMIN',
PRIVILEGE_TYPE => 'CAPTURE', GRANT_SELECT_PRIVILEGES => TRUE, DO_GRANTS => TRUE,
CONTAINER => 'ALL');
END;
/
```

30. In `ogg2`, grant the required privileges to the created user.

```
BEGIN
 DBMS_GOLDENGATE_AUTH.GRANT_ADMIN_PRIVILEGE ( GRANTEE => 'c##ggadmin',
PRIVILEGE_TYPE => 'APPLY', GRANT_SELECT_PRIVILEGES => TRUE, DO_GRANTS => TRUE,
CONTAINER => 'ALL');
END;
/
```

31. In `ogg1` and `ogg2`, grant the following further privileges.

```
GRANT CREATE SESSION TO c##ggadmin CONTAINER=all;
GRANT ALTER SYSTEM TO c##ggadmin CONTAINER=all;
GRANT RESOURCE TO c##ggadmin CONTAINER=all;
GRANT CONNECT TO c##ggadmin CONTAINER=all;
GRANT SELECT ANY DICTIONARY TO c##ggadmin CONTAINER=all;
GRANT EXECUTE on DBMS_FLASHBACK TO c##ggadmin CONTAINER=all;
GRANT LOCK ANY TABLE TO c##ggadmin CONTAINER=all;
GRANT ALTER USER TO c##ggadmin CONTAINER=all;
-- in real life, you might not really need this privilege:
GRANT UNLIMITED TABLESPACE TO c##ggadmin CONTAINER=all;
```

32. In `ogg1`, grant the following privilege to the common user:

```
GRANT SELECT ANY TRANSACTION TO c##ggadmin CONTAINER=all;
```

33. In `ogg2`, grant the following privilege to the common user:

```
GRANT INSERT ANY TABLE TO c##ggadmin CONTAINER=all;
GRANT UPDATE ANY TABLE TO c##ggadmin CONTAINER=all;
GRANT DELETE ANY TABLE TO c##ggadmin CONTAINER=all;
GRANT CREATE TABLE TO c##ggadmin CONTAINER=all;
```

34. In `ogg1`, create a user that will be used for replicating sequences and grant the required privileges to it.

Note: If the system you are replicating does not need synchronizing the sequences, do skip this step.

Note: Synchronizing sequences is not supported in bi-directional configuration.

```
conn / as sysdba
ALTER SESSION SET CONTAINER=pdb1;
CREATE USER GGATE IDENTIFIED BY oracle DEFAULT TABLESPACE USERS TEMPORARY
TABLESPACE TEMP QUOTA UNLIMITED ON USERS CONTAINER=CURRENT;
GRANT CONNECT, RESOURCE, DBA TO GGATE ;
@sequence
GRANT EXECUTE on ggate.updateSequence TO c##ggadmin;
```

35. In `ogg1`, issue the following statement. This step is needed only when replicating sequences is required.

```
conn / as sysdba

ALTER TABLE sys.seq$ ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY) COLUMNS;
```

36. In `ogg2`, create a user that will be used for replicating sequences and grant the required privileges to it.

Note: If the system you are replicating does not need synchronizing the sequences, do skip this step.

```
conn / as sysdba
ALTER SESSION SET CONTAINER=PDB1;
CREATE USER GGATE IDENTIFIED BY oracle DEFAULT TABLESPACE USERS TEMPORARY
TABLESPACE TEMP QUOTA UNLIMITED ON USERS CONTAINER=CURRENT;
GRANT CONNECT, RESOURCE, DBA TO GGATE;
@sequence
GRANT EXECUTE ON GGATE.REPLICATESEQUENCE TO C##GGADMIN;
```

37. In ogg2, grant the following privileges to the c## account.

Note: Granting these privileges are not part of the standard procedure for replicating sequences. However, without those privileges, the Replicat either hangs on replicating sequences or abends with ORA-01031 error.

```
conn / as sysdba
GRANT SELECT ANY SEQUENCE TO C##GGADMIN CONTAINER=ALL;
GRANT ALTER ANY SEQUENCE TO C##GGADMIN CONTAINER=ALL;
```

38. In ogg1, verify that all the tables in soe schema have primary key constraints.

Observe that the tables ORDERENTRY_METADATA and LOGON do not have a primary keys.

```
conn soe/soe@pdb1ogg1
SELECT TABLE_NAME FROM USER_TABLES T
WHERE NOT EXISTS (SELECT 1
                  FROM USER_CONSTRAINTS C
                  WHERE C.TABLE_NAME= T.TABLE_NAME AND C.CONSTRAINT_TYPE = 'P');
```

39. Create unique indexes on ORDERENTRY_METADATA and LOGON

```
CREATE UNIQUE INDEX ORDENTRY_UQ ON ORDERENTRY_METADATA(METADATA_KEY) TABLESPACE
USERS;
CREATE UNIQUE INDEX LOGON_UQ ON LOGON(LOGON_ID) TABLESPACE USERS;
```

40. In soe schema, make sure all the primary key constraints are enabled and validated:

Only validated primary key constraints are considered by GoldenGate capture. For some reason, Swingbench creates its primary key constraints invalidated.

```
col TABLE_NAME format a20
col CONSTRAINT_NAME format a35

SELECT TABLE_NAME, CONSTRAINT_NAME, STATUS, VALIDATED
FROM USER_CONSTRAINTS WHERE CONSTRAINT_TYPE='P';
```

41. Run the following code to enable and validate the primary keys:

```
BEGIN
 FOR R IN (SELECT TABLE_NAME, CONSTRAINT_NAME FROM USER_CONSTRAINTS WHERE
CONSTRAINT_TYPE='P') LOOP
  EXECUTE IMMEDIATE 'ALTER TABLE '|| R.TABLE_NAME || ' ENABLE VALIDATE
CONSTRAINT ' || R.CONSTRAINT_NAME;
 END LOOP;
END;
/

-- verify:
SELECT TABLE_NAME, CONSTRAINT_NAME, STATUS, VALIDATED
FROM USER_CONSTRAINTS WHERE CONSTRAINT_TYPE='P';
```

For security reasons, it is recommended to create a credential store to save the user passwords in it. This is much better than entering the user password in the parameter file.

42. In `ogg1`, run the following commands to create a credential store and save the GG admin user credential in it.

```
ggsci
add credentialstore
alter credentialstore add user c##ggadmin password oracle alias c##ggadminogg1
dblogin useridalias c##ggadminogg1
```

43. In `ogg2`, run the following commands to create a credential store. Create two aliases in it. One for the root container and one for the destination pdb (`PDB1`).

```
ggsci
add credentialstore
alter credentialstore add user c##ggadmin password oracle alias c##ggadminogg2
# this is the account that will be used by the replicat to pdb1
alter credentialstore add user c##ggadmin@pdb1ogg2 password oracle alias
c##ggadminpdb1

# to verify:
info credentialstore
```

44. In `ogg1`, enable schema-level supplemental logging for all `soe` tables.

Note: Do not ignore the displayed information. Read them and make sure they do not represent an issue. One of the common messages that you would look for is the following message:

"No unique key is defined for table <table_name>"

You will see this message for tables that do not have primary keys or unique indexes. You will also see it for the tables with invalided or disabled primary key or unique constraints.

If you build an index on a table to fix this message, you need first to drop the logging configuration using the following command and build it again:

`delete schematrandata pdb1.soe`

```
add schematrandata pdb1.soe
```

45. In `ogg1` and `ogg2`, create a `GLOBALS` file and set the `GGSCHEMA` parameter in it to the user used for synchronizing sequences. You must restart the `ggsci` after creating or modifying this file.

Note: The documentation sometimes refers to this schema as DDLuser.

Note: Observe the `GLOBALS` parameter file exists in the GG home, not in the `dirprm` subdirectory.

```
EDIT PARAMS ./GLOBALS
```

```
GGSCHEMA ggate
```

46. In `ogg1` and `ogg2`, exit from `ggsci` then invoke it again so that the `GLOBALS` file is read and make sure it does not have syntax error.

47. In `ogg2`, copy the script `recreate_soe.sql` to the staging folder. Run the script in SQL*Plus as sysdba to create the `soe` schema in the destination database.

```
sqlplus / as sysdba

@ /media/sf_staging/recreate_soe.sql
```

48. Create a testing table and sequence in the source database.

Those objects will be used to test the GoldenGate replication after it is set up.

```
sqlplus  soe/soe@pdb1ogg1
CREATE TABLE TEST ( PID NUMBER ) ;
CREATE SEQUENCE TEST_SEQ NOCACHE;
```

G. Configuring GoldenGate Processes

49. In `ogg1`, edit the manager process parameter file and add the code that follows to it.

```
edit param mgr
```

```
Port 7810
DynamicPortList 9500-9520
PurgeOldExtracts ./dirdat/*, UseCheckpoints, MinKeepDays 5
```

50. Start the Manager process.

```
start mgr

Info mgr
```

51. Repeat the same steps to create and start the Manager process in the target database system `ogg2`.

52. Create directories to save the trail files into them.

It is easier to manage the trail files when the trail files of each PDB are saved in a separate directory.

```
-- in ogg1:
sh mkdir ./dirdat/es

-- in ogg2:
sh mkdir ./dirdat/rtogg1
```

53. In `ogg1`, create the Extract parameter file and add the code that follows to it.

```
edit param eogg1
```

```
Extract eogg1
USERIDALIAS c##ggadminogg1
ReportCount Every 30 Minutes, Rate
Report at 01:00
ReportRollover at 01:15 on SUNDAY
-- if timezone cannot be obtained from pdb, assume it is the same as the root:
TRANLOGOPTIONS USE_ROOT_CONTAINER_TIMEZONE
-- include scheduling col values: pk, unq, and fk
LOGALLSUPCOLS
-- include before and after images in the same record
UPDATERECORDFORMAT COMPACT
ExtTrail ./dirdat/es/es
Sequence pdb1.soe.*;
Table pdb1.soe.*;
```

54. Add the Extract process and its trail.

```
DBLOGIN USERIDALIAS c##ggadminogg1
-- if you want to extract from more than one pdb1, add them between the brackets
--  separated by commans
REGISTER EXTRACT eogg1, DATABASE CONTAINER (pdb1)
ADD EXTRACT eogg1, INTEGRATED TRANLOG, BEGIN NOW
Add ExtTrail ./dirdat/es/es, Extract eogg1
```

55. Create the Data Pump parameter file and add the code that follows to it.

```
edit param pogg1
```

```
Extract pogg1
Passthru
RmtHost ogg2, MgrPort 7810
RmtTrail ./dirdat/rtogg1/ra
SEQUENCE pdb1.soe.*;
TABLE pdb1.soe.*;
```

56. Add the Data Pump Extract group

```
Add Extract pogg1, ExtTrailSource ./dirdat/es/es

-- you can create a trail for each pdb1
Add RmtTrail ./dirdat/rtogg1/ra, Extract pogg1
```

57. Start the Extract process and flush the sequences immediately after starting the Extract.

As always, the first tool to look into for troubleshooting any issue is the file `ggserr.log`

```
dblogin useridalias c##ggadminogg1
start extract eogg1
-- If you are not synchronizing the sequences, skip flushing the sequences
flush sequence pdb1.soe.*
info extract eogg1
```

58. Look in the trail directory and verify that local trail(s) are created.

```
sh ls -al dirdat/es
```

59. Start the Data Pump process group and make sure it is running fine.

```
start extract pogg1
info ER *
info all
```

60. In `ogg1`, check on the warning or error messages. Make sure no action is required to fix them.

Tip: In `vi` editor, to go to the end of the file, press on the keys `[Shift]+[g]`

```
sh vi ggserr.log
```

61. In `ogg2`, verify the trail files are created:

```
sh ls -al $GG_HOME/dirdat/rtogg1/
```

62. In `ogg2`, create the Replicat parameter file and add the parameters that follows to it.

```
edit param rpdb1
```

```
Replicat rpdb1
USERIDALIAS c##ggadminpdb1
ReportCount Every 30 Minutes, Rate
Report at 01:00
ReportRollover at 01:15 on SUNDAY
DBOPTIONS INTEGRATEDPARAMS(max_sga_size 200, COMMIT_SERIALIZATION FULL)
DiscardFile ./dirrpt/rogg2.dsc, purge
Map pdb1.soe.*, Target soe.*;
```

63. Register the Replicat.

Do not start the Replicat now. It will be started after the initial loading is finished.

```
DBLOGIN USERIDALIAS c##ggadminogg2
ADD REPLICAT rpdb1, INTEGRATED, EXTTRAIL ./dirdat/rtogg1/ra
```

H. Performing Initial Loading and Starting GoldenGate Processes

In this practice section, you will use Data Pump utilities to perform the initial loading.

64. In `ogg1` and `ogg2`, create a staging directory to save the dump files into it.

```
mkdir ~/staging
sqlplus / as sysdba
ALTER SESSION SET CONTAINER=PDB1;
CREATE DIRECTORY STAGING AS '/home/oracle/staging';
grant read,write on directory staging to soe;
grant read,write on directory staging to system;
```

65. In `ogg1`, obtain the current SCN value and take a note of it.

```
conn / as sysdba
SELECT CURRENT_SCN FROM V$DATABASE;
```

66. In `ogg1`, export `soe` schema up to the SCN value obtained in the preceding step.

It is always recommended to encrypt the generated dump files with a password.

```
expdp soe/soe@pdb1ogg1 schemas=soe directory=STAGING parallel=2
dumpfile=soe%u.dmp logfile=soe.log encryption_pwd_prompt=yes logtime=all
flashback_scn=<scn>
```

67. In `ogg2`, copy the produced dump files into the staging directory.

```
scp oracle@ogg1:/home/oracle/staging/soe*.dmp /home/oracle/staging
```

68. In `ogg1`, delete the dump files.

```
rm /home/oracle/staging/*
```

69. In `ogg2`, import the dump files into `pdb1`.

```
impdp system/oracle@pdb1ogg2 directory=staging schemas=soe logfile=soe.log
remap_tablespace=users:soetbs encryption_pwd_prompt=yes logtime=all
dumpfile=soe%u.dmp

# if the import succeeds, delete the dump files
rm /home/oracle/staging/*
```

70. Compile the package `ORDERENTRY`

```
sqlplus soe/soe@pdb1ogg2
ALTER PACKAGE "SOE"."ORDERENTRY" COMPILE;
```

71. In `ogg2`, start the Replicat after the obtained SCN value.

```
ggsci
start replicat rpdb1, aftercsn <scn>
info rpdb1
lag rpdb1
```

I. Testing the Replication

72. In `ogg1`, insert some sample data using the sequences into the testing table:

```
sqlplus soe/soe@pdb1ogg1
INSERT INTO TEST VALUES (TEST_SEQ.NEXTVAL);
INSERT INTO TEST VALUES (TEST_SEQ.NEXTVAL);
INSERT INTO TEST VALUES (TEST_SEQ.NEXTVAL);
COMMIT;
```

73. In `ogg2`, verify the change is replicated.

If the data change is not replicated, check out the errors in `ggserr.log` file in each host, resolve them, then try again.

```
sqlplus soe/soe@pdb1ogg2
SELECT * FROM TEST ;
```

Note: Consider taking a snapshot at this stage for the vms.

74. If the test succeeds, start Swingbench sessions.

75. Wait for a minute and make sure all the GoldenGate processes are working fine.

```
# in ogg1:
# run the following ggsci command several times and make sure the rba is
#  increamenting
Info *

# in ogg2:
# run the following ggsci command several times and make sure the rba is
#  increamenting
info rpdb1
```

Now you have the application sessions running as normal and the replication going on from the current production database (12c) to the new database release (19c). When we are ready for migrating to the new database, we start with the following steps. This where the downtime of this procedure starts.

76. Stop Swingbench sessions.

This is a simulation to stopping users from using the system.

77. In `ogg1`, use the lag command to make sure that the Extract captured all the pending transactions. Then stop the Extract.

```
lag eogg1
-- stop if EOF is reported in the lag command output. If not, try the lag
--  command again.
stop eogg1

lag pogg1
stop pogg1
```

78. In `ogg2`, make sure the Replicat processed all the pending transactions in the trial files.

```
lag rpdb1
stop rpdb1
```

79. Change the connection configuration in Swingbench so that it connects to `pdb1` in `ogg2`.

80. Start Swingbench sessions.

This step represents the end of the downtime period. If it succeeds, it means the upgrade is concluded.

81. Stop the Swingbench session.

```
sqlplus soe/soe@pdb1ogg1

DROP TABLE TEST ;
DROP SEQUENCE TEST_SEQ;

sqlplus soe/soe@pdb1ogg2

DROP TABLE TEST;
DROP SEQUENCE TEST_SEQ;
```

Summary

- Specific modifications should be made to configure GoldenGate in a multitenant environment.

- GoldenGate can be used to upgrade an application database with nearly zero downtime. We just need to make sure that the databases are supported by GoldenGate and the application data is not influenced by GoldenGate restrictions and constraints.