Practice 19

Cross-Platform Data Transport Using Image Copies

Practice Target

In this practice you will perform tablespace and database cross-platform transportation (from Linux to Windows) using image copies.

Practice Overview

In this practice, you will implement the following cross-platform transport tasks:

- Perform cross-platform tablespace transport from `srv1` to `winsrv2` using image copies with conversion on the destination host.
- Perform cross-platform database transport from `srv1` to `winsrv2` using image copies with conversion on the source host.

Assumptions

This practice assumes the following are true:

- `srv1` appliance is up and running and its database `ORADB` is running in `OPEN` state.
- `winsrv2` appliance is up and running and its database `ORAWIN` is running in `OPEN` state.
- You downloaded the script `create_soe.sql` from the lecture downloadable resources.

About Opening a Linux file in Windows

In some practice steps, you will need to edit a file generated in a Linux platform in a Windows platform. In such cases, if you open the file in the standard Windows Notepad, the new lines do not appear.

One method to deal with this issue is to open the file in Wordpad, save it, then close it. After you do so, the next time you open the same file with Notepad, you can will see the new lines in it.

To open a file in Wordpad, you can **right-click** on it -> select “**Open With**” option -> **Choose another app** -> click on “**More Apps**” link -> select **Wordpad** and **unmark the checkbox** that reads “Always use tis app to open the file.”

Note:

Do not take snapshots of the appliances now. You will take them later.

A. Practice Preparation Steps

Perform the following steps to prepare the environment for this practice.

1. Start Putty and connect to srv1 as oracle

2. Invoke SQL*Plus and connect as sysdba to the local database.

```
sqlplus / as sysdba
```

3. Create a directory object that points to the backup directory in the shared folder.

```
CREATE DIRECTORY SF_DIR AS '/media/sf_extdisk/backup';
```

4. Grant read/write privilege on the directory object to system and soe user.

```
GRANT READ, WRITE ON DIRECTORY SF_DIR TO SYSTEM;
GRANT READ, WRITE ON DIRECTORY SF_DIR TO SOE;
```

5. In winsrv2, create a staging folder. In the practice document examples, the staging folder is D:\temp.

6. In winsrv2, open a command-prompt window, invoke SQL*Plus and connect as sysdba to the local database ORAWIN.

   In Windows, to copy text from a command-prompt window, just highlight with the mouse pointer the text that you want to copy then press [Enter]. To paste, right-click anywhere on the command-prompt window.

```
sqlplus / as sysdba
```

7. Create a directory object that points to the staging folder.

```
CREATE DIRECTORY TEMP_DIR AS 'D:\temp';
```

8. Grant read/write privilege on the directory object to system.

```
GRANT READ, WRITE ON DIRECTORY TEMP_DIR TO SYSTEM;
```

9. Execute the code in the script create_soe.sql to create the user soe in ORAWIN database and grant the required privileges to it.

   The script is available for download in the lecture downloadable resources.

10. Exit from SQL*Plus

11. Take snapshot of srv1. Give it the name “**Practice 19 Start**”.

12. Take snapshot of winsrv2. Give it the name “**Practice 19 Start**”.

B. Performing Cross-platform Tablespace Transport with Image Copies

In this section of the practice, you will transfer the tablespace soetbs from a Linux x86 64-bit platform (srv1) to a Windows 64-bit platform (winsrv2). You will perform the datafile conversion at the destination host.

Preparing the Files on the Source Host

The following steps are to be done in srv1.

13. In the SQL*Plus session that is connected to ORADB, make sure that the source and destination platforms are supported for cross-platform transportation.

```
SELECT PLATFORM_NAME FROM V$TRANSPORTABLE_PLATFORM
WHERE UPPER(PLATFORM_NAME) LIKE '%LINUX%X86%64-BIT%'
   OR UPPER(PLATFORM_NAME) LIKE '% WINDOWS%X86%64-BIT%';
```

14. Obtain the endian of each platform.

Although in our case the endian of the source platform is the same as the endian of the destination platform, you will still perform the conversion command for the sake of education.

```
SELECT PLATFORM_NAME, ENDIAN_FORMAT FROM V$TRANSPORTABLE_PLATFORM
WHERE UPPER(PLATFORM_NAME) LIKE '%LINUX%X86%64-BIT%'
   OR UPPER(PLATFORM_NAME) LIKE '% WINDOWS%X86%64-BIT%';
```

15. Make sure the tablespace soetbs is self-contained

```
exec DBMS_TTS.TRANSPORT_SET_CHECK('SOETBS', TRUE,TRUE);


# after executing the procedure above, the following query should return no row:
SELECT * FROM TRANSPORT_SET_VIOLATIONS;
```

16. Place the tablespace soetbs in read-only mode.

```
ALTER TABLESPACE soetbs READ ONLY;
```

17. Copy the current tablespace datafiles to the shared folder.

a. Obtain the fullname of soetbs datafile. Take a note of it.

```
SELECT NAME FROM V$DATAFILE WHERE TS#=(SELECT T.TS# FROM V$TABLESPACE T WHERE
NAME='SOETBS');
```

b. Copy the file to the shared folder

```
host cp /u01/app/oracle/oradata/ORADB/datafile/<soetbs_dfile>
/media/sf_extdisk/backup/
```

18. Create the export dump file on srv1

```
expdp system/oracle dumpfile=soetbs.dmp directory=SF_DIR transport_tablespaces=soetbs
logfile=soetbs.log
```

19. On the hosting PC, move the produced files (tablespace datafile and the dump file) from the shared folder accessed by the Linux appliance (srv1) to the shared folder accessed by Windows appliance (winsrv2).

In my environment, I moved the files in my PC from D:\staging\Linux\backup to D:\staging\Windows\backup

Note: tablespaces cannot be plugged into destination databases, if the owners of their contents are not there in the database. If this is case in a real life scenario, produce script in the source database to re-create the users that own the tablespace objects in the destination database.

Plugging in soetbs Tablespace into winsrv2

Following are the steps to be done in winsrv2 to plug in the soetbs tablespace in ORAWIN database.

20. In the VirtualBox window of winsrv2, open a command-prompt window.

21. Invoke SQL*Plus and login to ORAWIN as sysdba.

```
sqlplus / as sysdba
```

22. Retrieve the location where ORAWIN datafiles are saved. Take a note of this location.

In our environment, it is D:\oracle\oradata\ORAWIN\DATAFILE

```
SELECT NAME FROM V$DATAFILE;
```

23. In winsrv2, copy the soetbs datafile and the dump file from the shared folder to the staging folder (D:\temp). Take a note of the full data file name after copying it. You will use it with the next commands.

Oracle does not support converting a datafile that is located in a network mapped drive in Windows, like the shared folder F: drive. That is why we need to copy it to a local drive.

```
host copy F:\backup\o1_*.dbf D:\temp
host copy F:\backup\*.dmp D:\temp
```

24. Invoke rman with connecting to the local instance then issue the CONVERT DATAFILE command on the datafile with saving the produced datafile in the directory noted earlier.

Substitute the <db file name> with the file full name of the data file copy in the D:\temp directory.

```
rman target /

CONVERT DATAFILE 'D:\temp\<db file name>'
 FROM PLATFORM 'Linux x86 64-bit'
FORMAT 'D:\oracle\oradata\ORAWIN\DATAFILE\%U';
```

25. Verify the produced file.

Observe the format of the auto-generated name of the new datafile.

```
host "dir D:\oracle\oradata\ORAWIN\DATAFILE\*SOETBS*";
```

26. Plug the tablespace into the destination database using Data Pump import utility.

You can obtain the full name of the created datafile from the output of the previous step.

```
impdp system/oracle dumpfile=soetbs.dmp directory=TEMP_DIR transport_datafiles='<db file
name>' logfile=soetbs.log
```

27. Set the default tablespace of soe to soetbs and grant the schema unlimited space on the tablespace.

```
sqlplus / as sysdba

ALTER USER SOE DEFAULT TABLESPACE SOETBS;
ALTER USER SOE QUOTA UNLIMITED ON SOETBS;
```

28. Select a sample data to verify that the transportation was successful.

```
SELECT COUNT(*) FROM SOE.ORDERS;
```

Clean Up

29. Shutdown both srv1 and winsrv2.

30. Restore srv1 to the snapshot “**Practice 19 Start**”. Start srv1.

31. Restore winsrv2 to the snapshot “**Practice 19 Start**”. Start winsrv2.

32. Delete the files from the shared folder.

C. Performing Cross-platform Database Transport with Image Copies

In this section of the practice you will transfer the database ORADB from Linux x86 64-bit platform (srv1) and create it as ORADB2 in the Windows 64-bit platform (winsrv2). Converting the datafiles is performed at the source host.

Preparing for the Files on the Source Host

Following are the steps to be done in srv1.

33. Start Putty and connect to srv1 as oracle

34. Invoke SQL*Plus and connect as sysdba to the local database.

```
sqlplus / as sysdba
```

35. Re-start the database in read-only mode.

```
SHUTDOWN IMMEDIATE
STARTUP MOUNT
ALTER DATABASE OPEN READ ONLY;
```

36. Check whether the database can be transported to a target platform.

```
SET SERVEROUTPUT ON

DECLARE
 v_ready BOOLEAN;
BEGIN
 v_ready := DBMS_TDB.CHECK_DB('Microsoft Windows x86 64-bit',DBMS_TDB.SKIP_READONLY);
 IF v_ready THEN
  DBMS_OUTPUT.PUT_LINE('Transportable to the platform.');
 ELSE
  DBMS_OUTPUT.PUT_LINE('Not Transportable to the platform.');
 END IF;
END;
/
```

37. Identify external tables, directories, or BFILEs.

In real life scenario, you need to take a decision on copying those objects in the destination database. For your practice environment, we will ignore them.

```
SET SERVEROUTPUT ON
DECLARE
 v_ext BOOLEAN;
BEGIN
 v_ext := DBMS_TDB.CHECK_EXTERNAL;
END;
/
```

38. Invoke RMAN with connecting to the local instance then generate the output files using CONVERT DATABASE command. Save the output files in the shared folder.

If the command returns the error “ORA-19926: Database cannot be converted at this time”, restart the database in read only state.

The command first lists the external tables, directory objects, and the users in the password file. Those entities should be manually copied or re-created in the destination database.

```
rman target /
```

```
CONVERT DATABASE
 NEW DATABASE 'ORADB2'
 TRANSPORT SCRIPT '/media/sf_extdisk/backup/transportscript.sql'
 TO PLATFORM 'Microsoft Windows x86 64-bit'
 FORMAT '/media/sf_extdisk/backup/%U';
```

39. Start the ORADB in read/write mode

```
SHUTDOWN
STARTUP
```

40. On the hosting PC, move the produced files from the shared folder accessed by the Linux appliance (srv1) to the shared folder accessed by Windows appliance (winsrv2).

Creating the Transported Database in winsrv2

Following are the steps to be done in winsrv2 to create ORADB2 database.

41. In the VirtualBox window of winsrv2, rename the generated init file in the shared folder to pfileORADB2.ora

42. Open a command-prompt window and paste the following code in it to create the directories that will accommodate the database files.

```
mkdir D:\oracle\oradata\ORADB2\CONTROLFILE
mkdir D:\oracle\oradata\ORADB2\DATAFILE
mkdir D:\oracle\oradata\ORADB2\ONLINELOG
mkdir D:\oracle\fast_recovery_area\ORADB2
mkdir D:\oracle\audit\ORADB2
```

43. Copy the converted datafiles from shared folder to the DATAFILE directory that you created in the previous step. Copy the datafiles, do not move them.

```
copy F:\backup\data* D:\oracle\oradata\ORADB2\DATAFILE
```

44. Copy the generated pfile and transportscript script from the shared folder to the local staging folder D:\temp

```
copy F:\backup\transportscript.sql D:\temp
copy F:\backup\pfileORADB2.ora D:\temp
```

45. Remove ORAWIN database and its service

A server can host multiple databases. We remove ORAWIN database just to release the resources in the server.

a. Invoke dbca utility and remove ORAWIN database.

b. Start the Windows Services (click on [Windows] button, type “**Services**”, and click on Services icon)

c. Verify that **OracleServiceORAWIN** is still there.

d. Open a command-prompt window as an administrator.

Click on [Windows] button, type “**cmd**”, when you see the cmd icon, right-click on it and elect “**Run As Administrator**”.

e. In the displayed command line, run the following command (do **not** close the command-line window now).

```
oradim -delete -sid ORAWIN
```

f. Refresh the **Services** window to verify that the service has been removed (you can press [F5] to refresh Services window).

46. Add ORADB2 service

a. In the administrator command-line window, run the following command. When it prompts for password, enter: **Mypassword123**

```
oradim -NEW -SID ORADB2
```

Note: if you enter a wrong password, the service will be created but it will not startup. You need to delete it (using the command `oradim -delete -sid ORADB2`) and create it again. Sometimes, the deletion command does not delete the service but it marks it for deletion. In this case, reboot the VM and the service will automatically be removed.

b. Verify that the service has been added and started.

c. Modify the service startup type to **automatic** (right-click on the service -> select properties -> change the startup type to automatic -> click on OK button).

d. Close the administrator command-line window.

e. Close Services window

47. Edit pfileORADB2.ora and change the values of initialization parameters that comes under a section remarked as:

“# Please change the values of the following parameters”

Remember you need to open the file with Wordpad first, save it, close it, then open it with Notepad.

```
control_files='D:\oracle\oradata\ORADB2\CONTROLFILE\control1.ctl','D:\oracle\oradata\ORADB2\DATAFILE\control2.ctl'
db_create_file_dest='D:\oracle\oradata\ORADB2\DATAFILE'
db_recovery_file_dest='D:\oracle\fast_recovery_area\ORADB2'
db_recovery_file_dest_size= 42949672960
audit_file_dest='D:\oracle\audit\ORADB2'
db_name='ORADB2'
```

48. Edit the file transportscript and make the following changes on it.

a. Fix the pfile full name pointed by PFILE in the file. You will fix them in three locations. You will need to save the file from Wordpad before you can edit it in notepad.

```
STARTUP NOMOUNT PFILE='D:\temp\pfileORADB2.ora'
```

b. Change the datafile locations from /media/sf_extdisk/backup to D:\oracle\oradata\ORADB2\DATAFILE

c. Change the forward slash symbol to back slash symbol in the directory names.

d. Replace the command that enables the BCT with the following command:

```
ALTER DATABASE ENABLE BLOCK CHANGE TRACKING;
```

49. Execute the script file transportscript

a. In the command-line window, set the ORACLE_SID variable and invoke SQL*Plus with connecting as sysdba to the local instance.

In the SQL*Plus prompt, you should see the message “Connected to an idle instance”. If you see the message “Protocol Adapter Error”, most likely the service that you created earlier did not start. Create the service successfully before you proceed.

```
set ORACLE_SID=ORADB2
sqlplus / as sysdba
```

b. Run each command in the script file transportscript manually.

I personally prefer to manually run the commands instead of running the entire script at once because running them individually is easier to maintain should you face any issue.

The script utlrp.sql takes long time to finish. Just be patient please.

If running the script fails in the middle and you want to restart the step again, you have to: (1) shutdown the instance (2) copy the datafiles again from the shared folder to the datafile location.

c. Verify the data is available.

```
SELECT COUNT(*) FROM SOE.ORDERS;
```

Note: In real life scenario, you should consider performing further post-database-creation steps like the following:

- Create external tables
- Create directory objects
- Create SPFILE
- Set the environment variable ORACLE_SID to the new database
- Set the auto-start script (in Linux when Oracle Restart is not configured)
- Configure the local listener

Clean Up

50. Shutdown both srv1 and winsrv2.

51. Restore `srv1` to the snapshot "**Practice 19 Start**" and start `srv1`.

52. Restore `winsrv2` to the snapshot "**Practice 19 Start**" and start `winsrv2`.

53. Delete the snapshots "Practice 19 Start" for `srv1` and `winsrv2`.

54. Delete the files from the shared folder.

Summary

In this practice, you implemented the following cross-platform transport tasks:

- Perform cross-platform tablespace transport from `srv1` to `winsrv2` using image copies with conversion on the destination host.
- Perform cross-platform database transport from `srv1` to `winsrv2` using image copies with conversion on the source host.