

Practice

Managing Initialization Parameters, SPFILE, and PFILE

Practice Target

In this practice you will perform the tasks involved in managing an Oracle database initialization parameters, the SPFILE, and the PFILE.

In high level, in this practice, you will perform the following tasks:

- Examine the SPFILE and the initialization parameters
- Change the initialization parameter file for the database

Practice Assumption

The practice assumes that `srv1` and its non-CDB database are up and running.



Ahmed Baraka
Oracle Database Administrator

Examining SPFILE and the Initialization Parameters

In the following steps, you will examine the initialization parameters in the SPFILE.

1. Open a Putty session to `srv1` as `oracle`.
2. Invoke SQL*Plus and connect to the database as `system`.

```
sqlplus system/ABcd##1234
```

3. Retrieve the SPFILE used by the instance when it started up.

The SPFILE name is `spfileoradb.ora`. This is the default SPFILE name and it comes from the format `spfile<ORACLE_SID>.ora`. It is located in the default location, which is `$ORACLE_HOME/dbs`. Every time the instance is started up, it automatically reads from this SPFILE, unless the `PFILE` option is used with the `STARTUP` command.

```
show parameter SPFILE
```

4. Display the contents of the SPFILE

The `cat` command does display the contents of the file. However, the file is actually a binary file and must not be edited with any editor. The only way to add, change, or remove a parameter from the SPFILE is via the `ALTER SYSTEM SET` statement.

Observe that most the parameters prefixed with asterisk symbol. This symbol does not make any difference in non-RAC databases like the one we have in `srv1`. It has an impact on a RAC databases.

The parameters that are prefixed with `oradb_`, they are mainly used to set the initial memory area sizes when the instance starts up. We will cover them in details later in the course. At this stage, you just need to know that those parameter values are not set by us. They were added by the instance.

The parameter file contains only a set of parameters. Their values are assigned from our answers to the `dbca` when we created the database. What about the other database parameters? How the instance derives their values?

```
host cat /u01/app/oracle/product/19.0.0/db_1/dbs/spfileoradb.ora
```

5. Retrieve the value of the `SGA_TARGET` parameter.

Observe that the parameter names in the view `V$PARAMETER` are in lower case.

```
show parameter SGA_TARGET
SELECT VALUE/1024/1024 MB FROM V$PARAMETER WHERE NAME='sga_target';
```

6. Retrieve the value of `SGA_TARGET` in the SPFILE.

Its values are the same in the memory and in the SPFILE. In my case there is a 1M difference, I don't know it cause. But this is negligible.

```
SELECT VALUE/1024/1024 MB FROM V$SPPARAMETER WHERE NAME='sga_target';
```

7. Change the value of `SGA_TARGET` to 2000M in the instance memory only.

```
ALTER SYSTEM SET SGA_TARGET=2000M SCOPE=MEMORY;
```

8. Verify that the parameter was changed in the memory and not in the SPFILE.

```
SELECT VALUE/1024/1024 MB FROM V$PARAMETER WHERE NAME='sga_target';  
SELECT VALUE/1024/1024 MB FROM V$SPPARAMETER WHERE NAME='sga_target';
```

9. Retrieve the value of the parameter `SGA_MAX_SIZE`.

This parameter specifies the maximum size that we can allocate for the SGA memory (regardless of the free memory available in the system).

```
show parameter sga_max_size
```

10. Try modifying the value of `SGA_MAX_SIZE` without setting the `SCOPE`.

By default the `SCOPE` equals to `BOTH`. The statement should return the following error:

```
ORA-02095: specified initialization parameter cannot be modified
```

This parameter cannot be modified in the memory because it is a static parameter. It can be modified only in the SPFILE. As we learnt, the changes made in the SPFILE take effect only after the instance restarts.

```
ALTER SYSTEM SET SGA_MAX_SIZE=1000M;
```

11. Display the current value of the `NLS_DATE_FORMAT` parameter

The parameter is not set to any value. Therefore, its default value will take effect.

This parameter controls the format of the dates displayed in the current session.

```
show parameter NLS_DATE_FORMAT
```

12. Display the current date and time.

The current date and time are displayed in the format `DD-MON-YY`

```
SELECT SYSDATE FROM DUAL;
```

13. Change the value of `NLS_DATE_FORMAT` for the current session. Then display the current date and time.

The current date and time are displayed as per the provided format.

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MM-YY HH24:MI:SS';
```

```
SELECT SYSDATE FROM DUAL;
```

14. Issue the following command to connect again to the database as `SYSTEM`. This command closes the current session and opens a new session.

```
conn system/ABcd##1234
```

15. Display the current date and time.

The current date and time are displayed in the default format. The change on the parameter was only for the scope of the session.

```
SELECT SYSDATE FROM DUAL;
```



Changing the Initialization Parameter File for the Database

In the following steps, you will develop an issue and resolve it by changing the database initialization parameter file.

16. Set the value of the `SGA_TARGET` to 50M in the SPFILE. This value is improper value because it is too small for the database to startup. This is a simulation of a scenario where a parameter is set an non-working value in the SPFILE.

```
ALTER SYSTEM SET SGA_TARGET=50M SCOPE=SPFILE;
```

17. Shutdown the database then try starting it up.

The `STARTUP` command fails because the `SGA_TARGET` is too small. It returns the following errors:

```
ORA-00821: Specified value of sga_target 52M is too small, needs to be at least 272M
ORA-01078: failure in processing system parameters
```

The parameter value is saved in the SPFILE and we cannot edit the SPFILE with an editor because it is a binary file. We cannot change its value using `ALTER SYSTEM` statement either because the instance is down.

```
conn / as sysdba
shutdown immediate
startup
```

The solution that you will implement in the following steps is to create PFILE from the SPFILE, fix the value of the `SGA_TARGET` in it, then rebuild the SPFILE from the PFILE.

18. Issue the following command to create a PFILE from the SPFILE.

```
CREATE PFILE='/home/oracle/PFILEtemp.ora' from SPFILE;
```

19. Open the `PFILEtemp.ora` file with the `vi` editor. Fix the value of the `SGA_TARGET` in it to 2516582400 (equivalent to $2400 \times 1024 \times 1024 = 2400M$).

```
host vi /home/oracle/PFILEtemp.ora
```

20. Startup the database with the PFILE.

This is just a temporary startup to make sure that the change is acceptable.

In real life scenario, consider stopping the Listener to make sure no user would connect to the database after we temporarily start up the database.

```
startup PFILE=/home/oracle/PFILEtemp.ora
```

21. Issue the following command to rebuild the SPFILE from the PFILE.

The statement creates the SPFILE in its default location and name. It overwrites the existing SPFILE.

```
CREATE SPFILE FROM PFILE='/home/oracle/PFILEtemp.ora';
```

- 22.** Shutdown and startup the database.

The `STARTUP` command reads from SPFILE this time.

```
shutdown immediate  
startup
```

- 23.** As a cleanup, delete the PFILE.

```
host rm /home/oracle/PFILEtemp.ora
```

- 24.** Exit from SQL*Plus.

```
quit
```



Summary

In this practice, you learnt the procedures to retrieve the initialization parameter values, change their values at the system and session levels, and manage the SPFILE and the PFILE.

