Using BOC Management Office® Products with Oracle® Databases

Prepare and create a database instance with Oracle 19c/21c

BOC Group
Design your Enterprise

Table of Contents

Introduction

This document is a screen-by-screen guide, which leads you through the full process of BOC Management Office® database creation with Oracle 19c or Oracle 21c.

Note: The guide uses standard settings for the database, which may not always fit your personal needs. The used settings are optional, unless noted otherwise.

Adapt Configuration of the Oracle Database Instance

Adapt the following parameters before creating a database for your BOC Management Office® product.

OPEN_CURSORS

The initialisation parameter OPEN_CURSORS specifies the maximum number of open cursors (handles to private SQL areas) a session can have at once. The default value is 50. This value must be increased to a minimum of 300 in the initialisation file of the database instance.

Settings on the Database Server

In this manual, the database name »adodb« is used as an example. The images are from Oracle 19c, but the process is the same for Oracle 21c (some of the field names used may differ slightly).

Part 1: Database Configuration Assistant

1. Open the Database Configuration Assistant (in the start menu of your operating system in the folder »Oracle«). Select »Create a Database«. Proceed with »Next«.

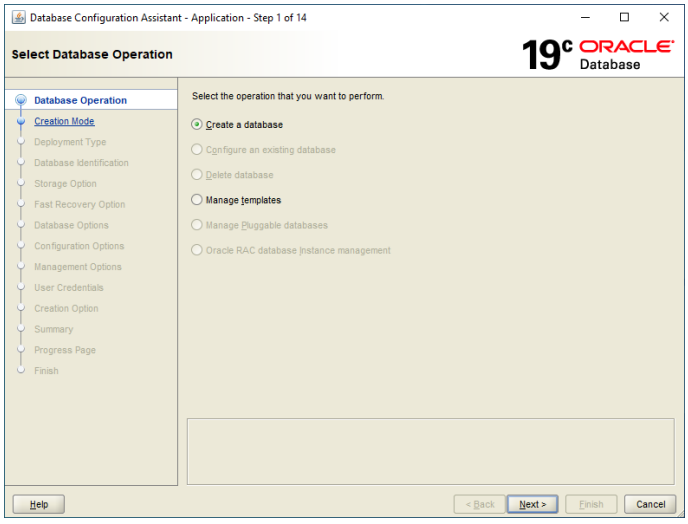

Fig. 1: Create database

2. Select »Advanced configuration«. Proceed with »Next«.

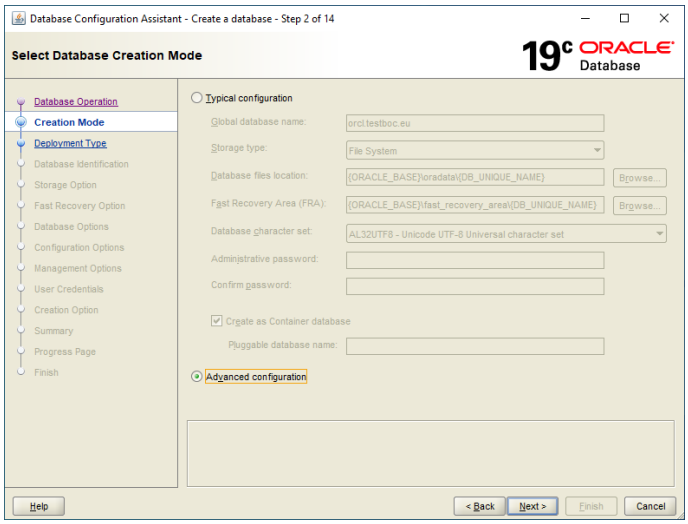

Fig. 2: Select advanced configuration

3. Select »Custom Database«. Proceed with »Next«.

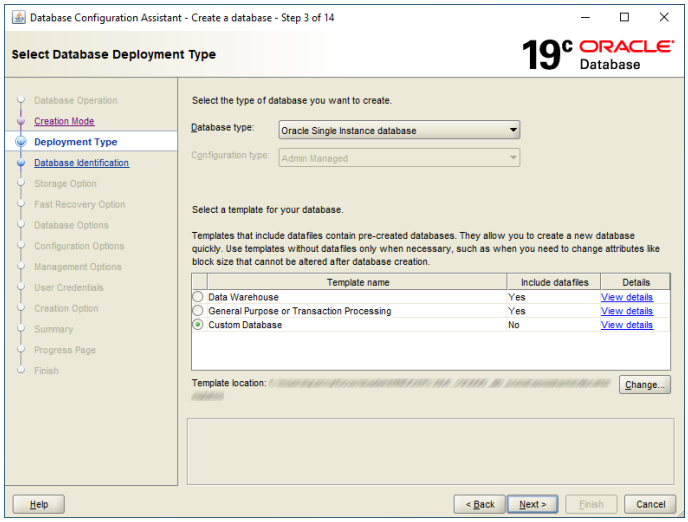

Fig. 3: Select custom database

4. Enter the database name in the »Global Database Name« and »SID« boxes.

   To create a **non-container database (non-CDB)**, clear the »Create as Container database« option. **Note:** This option is preselected and cannot be disabled in Oracle Database 21c and later, as the non-CDB architecture

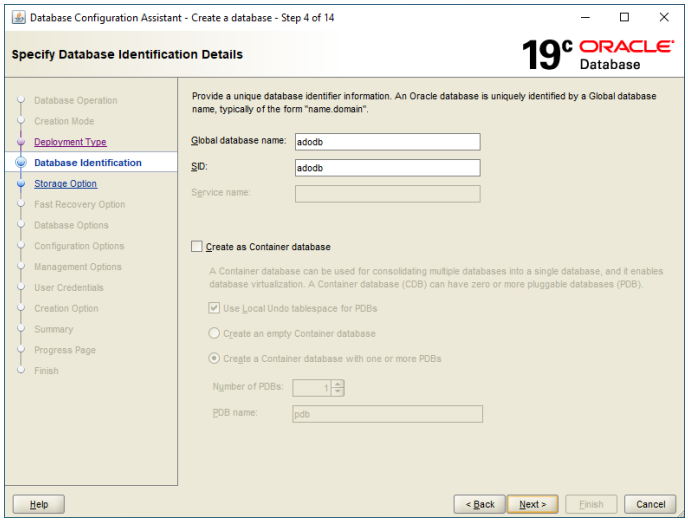

Fig. 4: Enter a global database name

is desupported.

To create a **container database (CDB)**, ensure »Create as Container database« is selected. Then, choose »Create a Container database with one or more PDBs«, set »Number of PDBs« to 1, and enter a name for the pluggable database (PDB) in the »PDB name« box.

Click »Next« to continue.

5. Enable »Use following for the database storage attributes«. Select »File System« as »Database files storage type«. The default »Database files location« is usually a good choice. Proceed with »Next«.

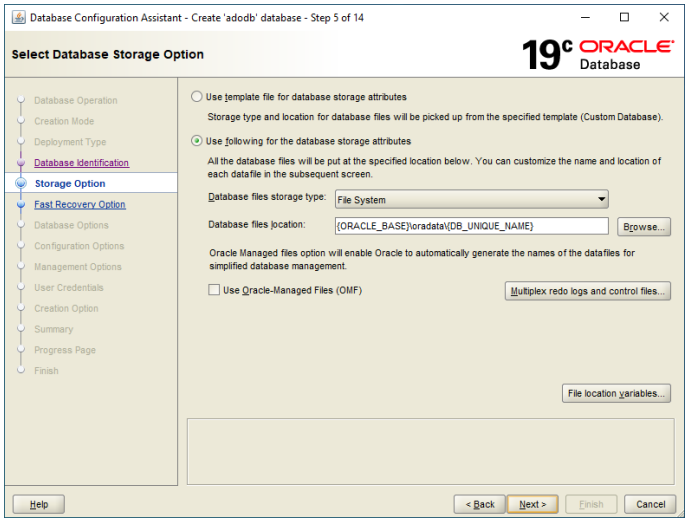

Fig. 5: Storage options

6. The default recovery options for the database are usually adequate. Proceed with »Next«.

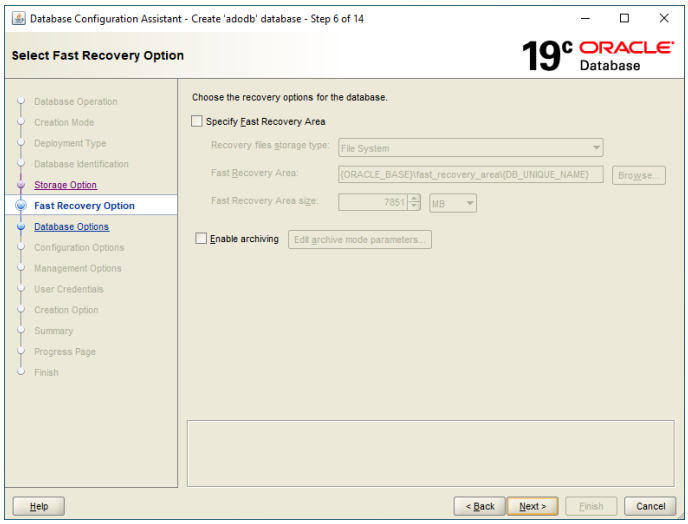

Fig. 6: Fast recovery options

7. Select an existing listener or create a new listener by specifying the listener name and port. Proceed with »Next«.

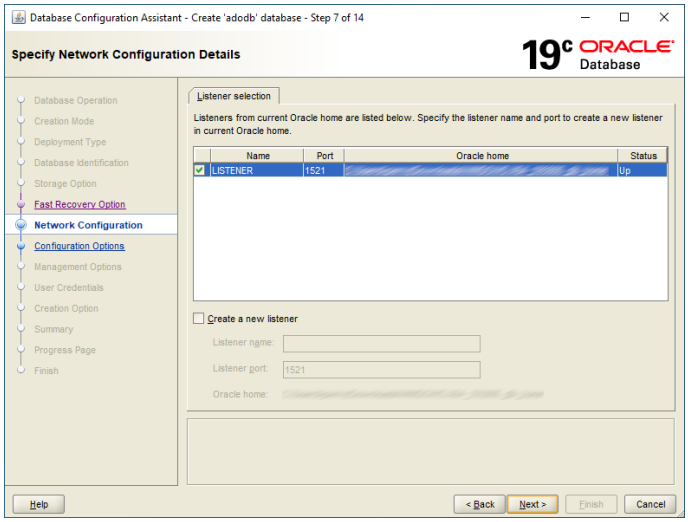

Fig. 7: Select or create listener

8. Clear all check boxes in the »Database components« tab. Proceed with »Next«.

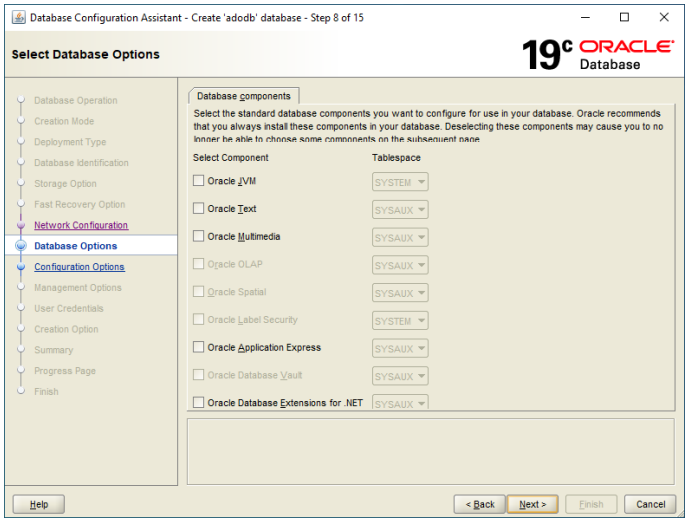

Fig. 8: Deactivate all database components

9. Activate »Use Automatic Shared Memory Management«. Enter a »SGA size« of 600 MB and a »PGA size« of 100 MB.

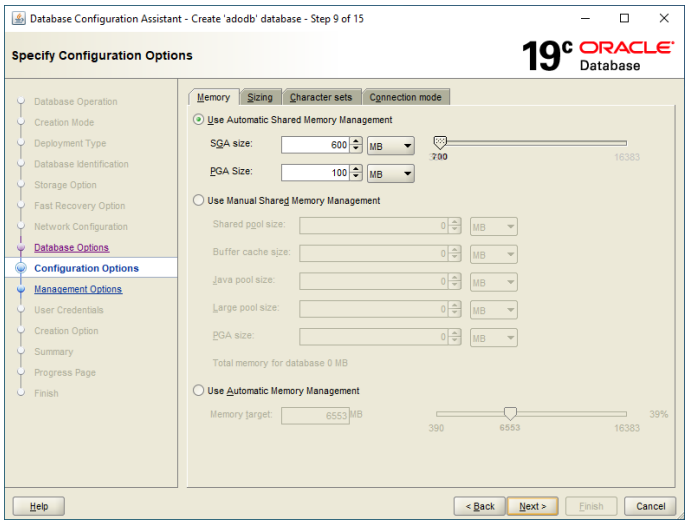

Fig. 9: Enter memory size (SGA and PGA)

10. Switch to the tab »Character Sets«. Select »Use Unicode (AL32UTF8)« and select »UTF8« from the »National Character Set« list.

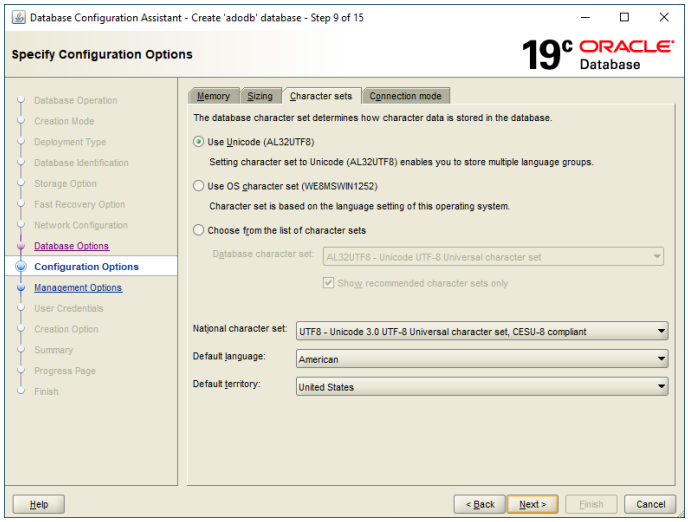

Fig. 10: Select character set »Unicode«

11. Switch to the tab »Connection Mode« and enable »Dedicated Server Mode«. Proceed with »Next«.

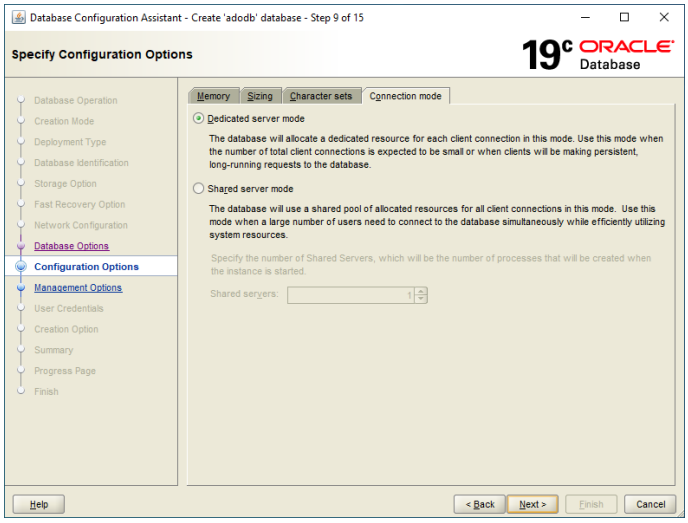

Fig. 11: Dedicated server mode

12. Select “Configure Enterprise Manager (EM) database express“. Proceed with »Next«.

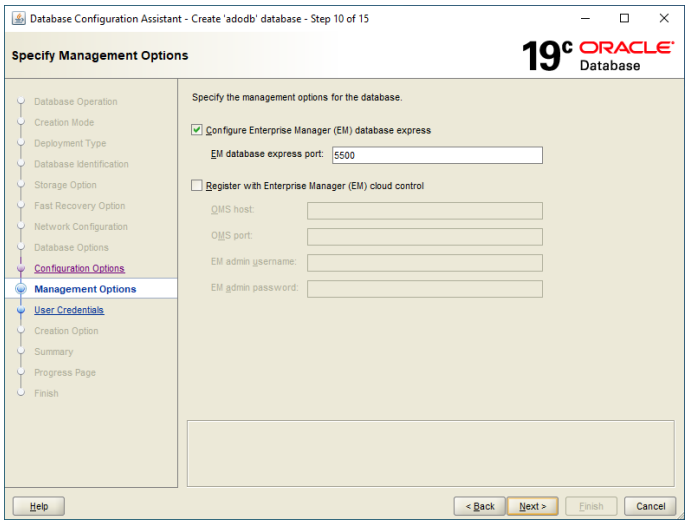

Fig. 12: Configure enterprise manager

13. Enter a password for the database users. Proceed with »Next«.

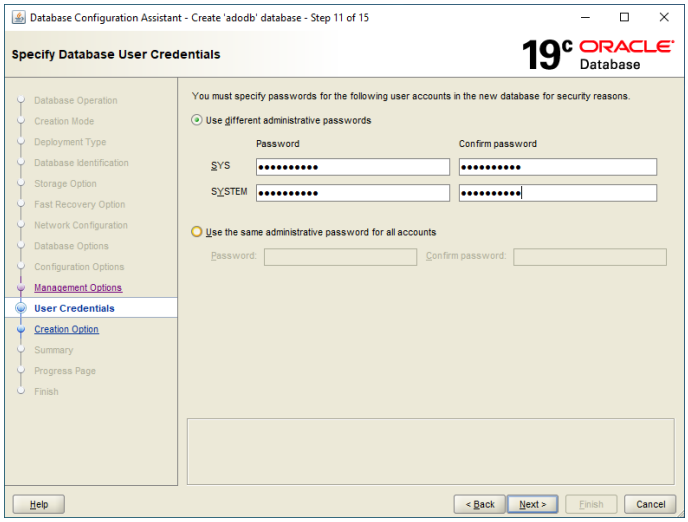

Fig. 13: Enter administrative passwords

14. Select »Create Database«. Proceed with »Next«.

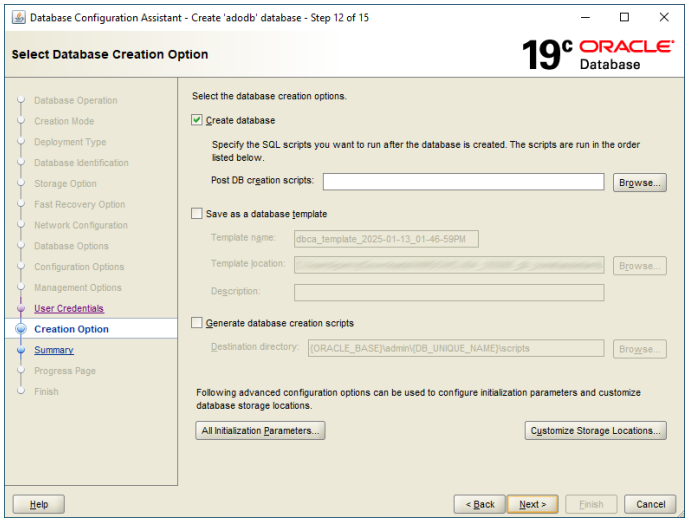

Fig. 14: Create database

15. A summary showing the details of the database configuration is displayed. Click »Finish« to start the database creation.

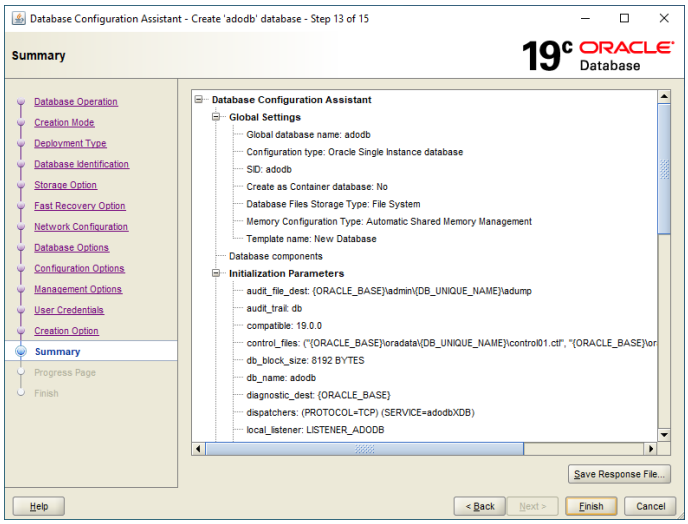

Fig. 15: Summary

The first part of the database creation is finished now.

Part 2: Script » aprporcl.sql «

Now it is time to adapt the script »`aprporcl.sql`« to your Oracle environment. You can find the script in the installation package of your BOC Management Office® product in the folder »`dbinfo`«:

```
/*
…
*/

REM **************TABLESPACE FOR USER********************
CREATE TABLESPACE ADOUSR_DATA DATAFILE
'<ORACLE_BASE>\oradata\<SID>\adousr1.dbf'
SIZE 500M AUTOEXTEND ON NEXT 250M MAXSIZE 5G
DEFAULT STORAGE ( INITIAL 500K NEXT 250K MINEXTENTS 1
MAXEXTENTS UNLIMITED PCTINCREASE 10);

CREATE PROFILE BOC_PROFILE LIMIT FAILED_LOGIN_ATTEMPTS
UNLIMITED PASSWORD_LIFE_TIME UNLIMITED;
CREATE USER ADOxx IDENTIFIED BY "r0KaQIFA]cPd2Ave" PROFILE
BOC_PROFILE;
ALTER USER ADOxx DEFAULT TABLESPACE ADOUSR_DATA QUOTA
UNLIMITED ON ADOUSR_DATA;
GRANT CONNECT TO ADOxx;

CREATE USER ADOXX_BOOT IDENTIFIED BY "iCfCK!lHP8S1L]Ry"
PROFILE BOC_PROFILE;
GRANT CONNECT TO ADOXX_BOOT;
CREATE SYNONYM ADOXX_BOOT.dbinfo FOR ADOxx.dbinfo;
GRANT SELECT ON ADOxx.dbinfo TO ADOXX_BOOT;
```

16. Replace the placeholder <ORACLE_BASE> with the path of your Oracle installation directory and <SID> with your database name and save the script. In this example `C:\Oracle` and `adodb` is used.
    Note: If you are running a container database (CDB), the placeholder <SID> must point to the pluggable database (PDB) subdirectory.
    Note: To find out how to replace the default database user »ADOxx« with an individual user, see **Create Database with an Individual DB User**.

```
REM **************TABLESPACE FOR USER********************
```

```
CREATE TABLESPACE ADOUSR_DATA DATAFILE
'c:\oracle\oradata\adodb\adousr1.dbf'
SIZE 500M AUTOEXTEND ON NEXT 250M MAXSIZE 5G
DEFAULT STORAGE ( INITIAL 500K NEXT 250K MINEXTENTS 1
MAXEXTENTS UNLIMITED PCTINCREASE 10);

CREATE PROFILE BOC_PROFILE LIMIT FAILED_LOGIN_ATTEMPTS
UNLIMITED PASSWORD_LIFE_TIME UNLIMITED;
CREATE USER ADOxx IDENTIFIED BY "r0KaQIFA]cPd2Ave" PROFILE
BOC_PROFILE;
ALTER USER ADOxx DEFAULT TABLESPACE ADOUSR_DATA QUOTA
UNLIMITED ON ADOUSR_DATA;
GRANT CONNECT TO ADOxx;

CREATE USER ADOXX_BOOT IDENTIFIED BY "iCfCK!lHP8S1L]Ry"
PROFILE BOC_PROFILE;
GRANT CONNECT TO ADOXX_BOOT;
CREATE SYNONYM ADOXX_BOOT.dbinfo FOR ADOxx.dbinfo;
GRANT SELECT ON ADOxx.dbinfo TO ADOXX_BOOT;
```

Part 3: SQL Plus

In this part the previously adapted script will be executed. For this step the program »SQL Plus« will be used.

Note: If you are running a CDB, make sure that the PDB is in read write mode before you execute the script (see **Modify the Open Mode of a PDB**).

17. Open a command prompt, and then enter the command »`sqlplus.exe system@<DB_name>`«. Replace `<DB_name>` with the name of your new database or the PDB name (if you are running a CDB). When prompted, enter the password previously defined for the database user »`system`« in step **13**.

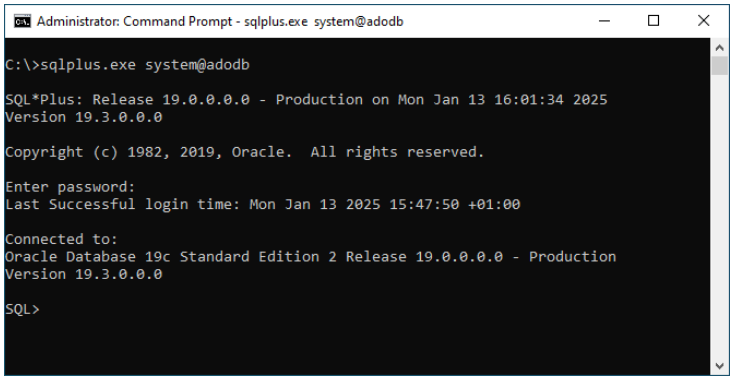

Fig. 16: Login to SQL Plus

18. Run the adapted script with the command »`start <pathname>\aprporcl.sql`«. `<pathname>` is the path where the script is located.

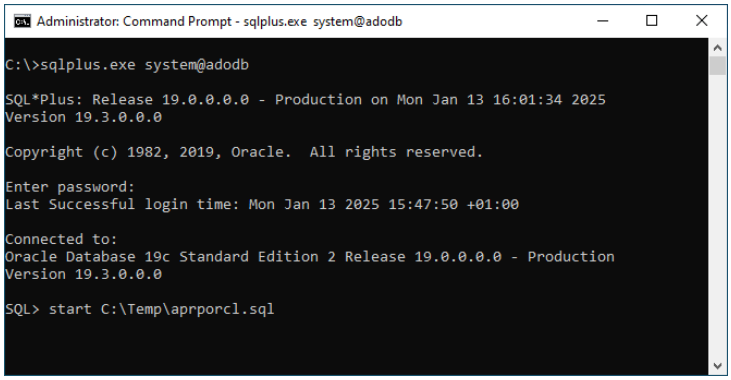

Fig. 17: Execute script

19. If the last step was successful, close SQL Plus with »`exit`«.

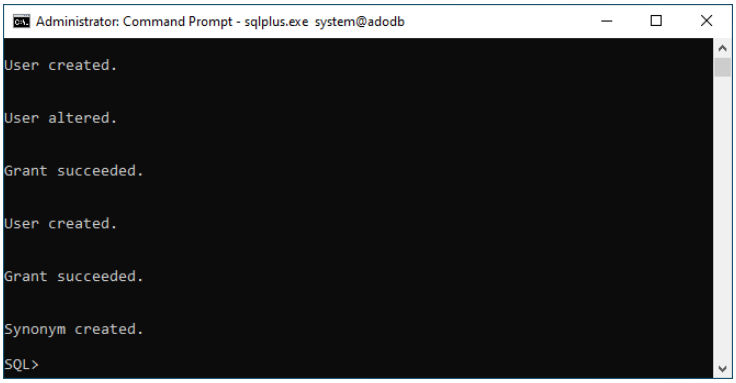

Fig. 18: Successfully created tablespace

Note: If any errors appear, you have to correct the script. Then you have to delete and newly create the database with the Database Configuration Assistant as described in the first chapter.

Settings on the Client

Part 4: Install 64-Bit Oracle Client

Install a 64-bit Oracle client on the machine where the application server is installed (for details see the Oracle documentation).

Part 5: Create a Net Service Name

In this part the creation of the Net Service Name (client connection) for the previously created database is described. This part has to be executed on the machine where the application server is installed.

20. Open the Oracle Net Configuration Assistant (in the start menu of your operating system in the folder »Oracle«). Select »Local Net Service Name configuration«. Proceed with »Next«.

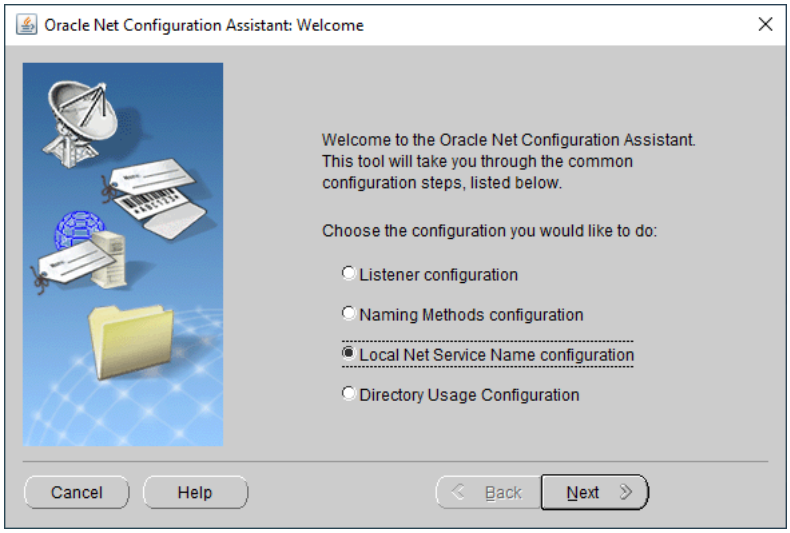

Fig. 19: Local Net Service Name configuration

21. Activate »Add«. Proceed with »Next«.

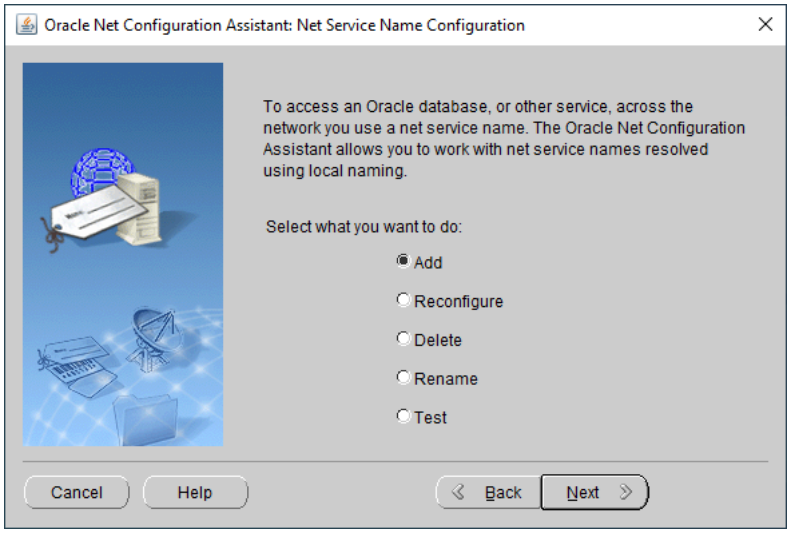

Fig. 20: Add net service name

22. In the »Service Name« box, enter the name of the previously created database (here e.g. »`adodb`«) or the PDB name (if you are running a CDB). Proceed with »Next«.

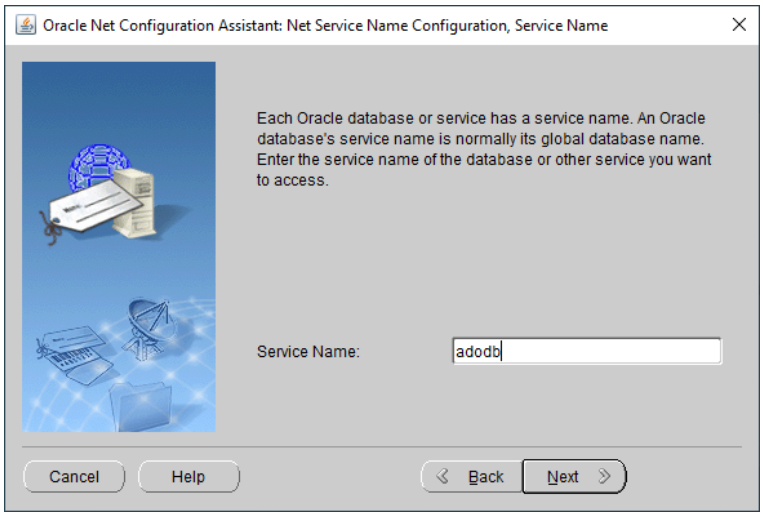

Fig. 21: Enter service name

23. Select »TCP« as the network protocol. Proceed with »Next«.

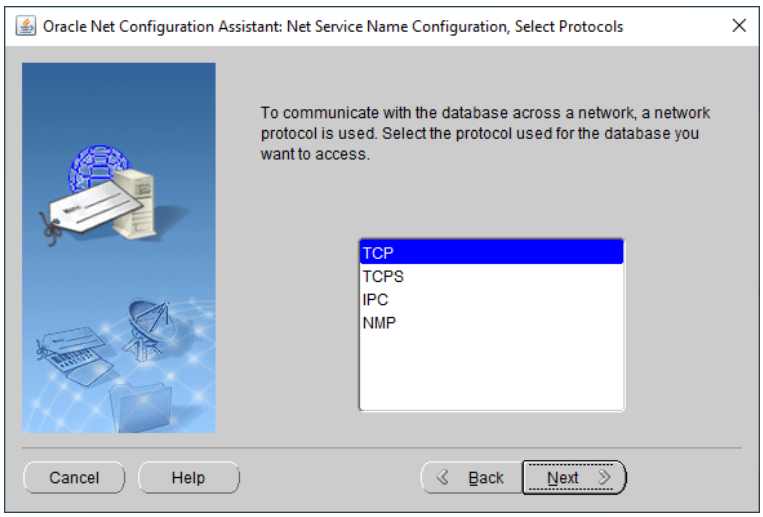

Fig. 22: Select network protocol

24. Enter the host name of your database computer (here »`dbserver`«) and the listener port specified in step 7 (here standard port »1521«). Proceed with »Next«.

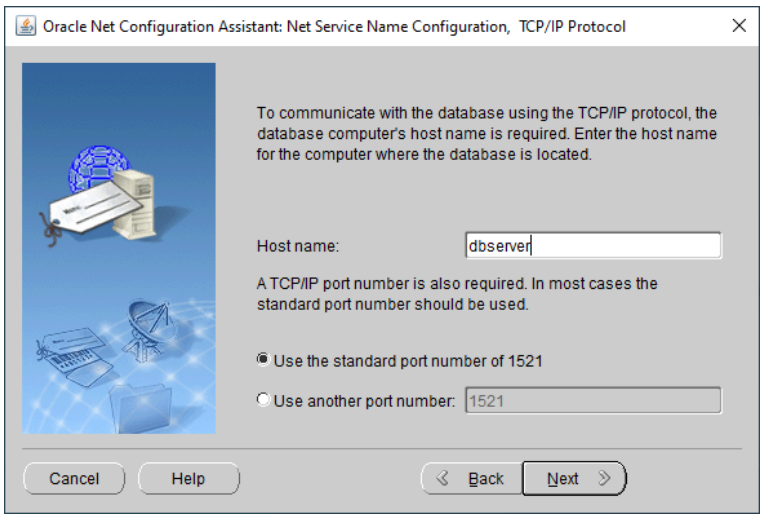

Fig. 23: Host name and port number

25. Activate »Yes, perform a test« and click »Next«. The database test will start.

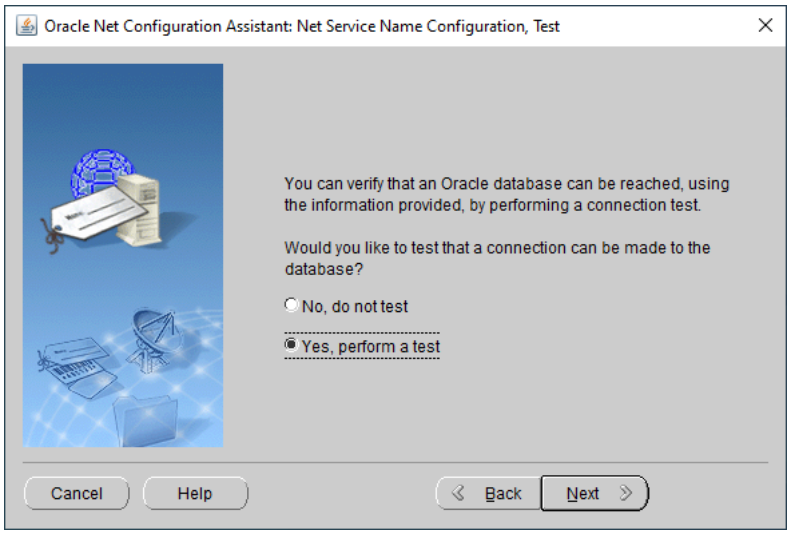

Fig. 24: Test connection

26. If the test was successful, click »Next«. If not, click »Change Login«.

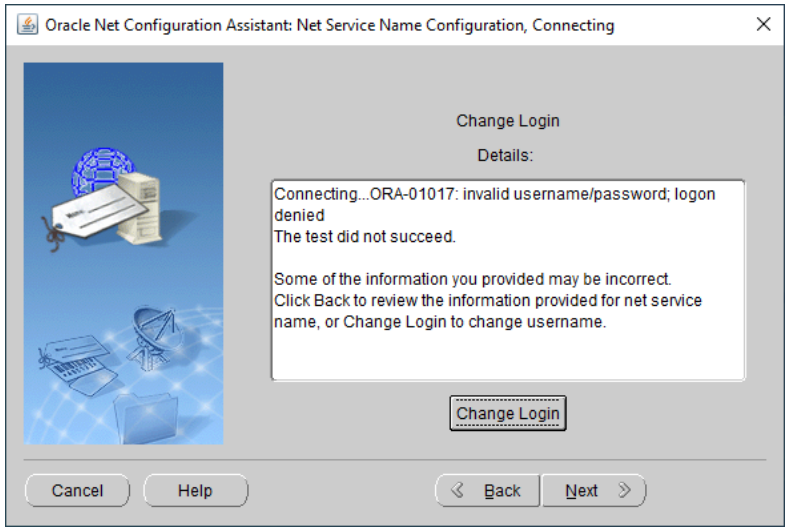

Fig. 25: Test result

27. Upon clicking »Change Login«, the dialogue »Change Login« opens. Enter the username »`system`« and the previously set password. Click »OK« to continue.

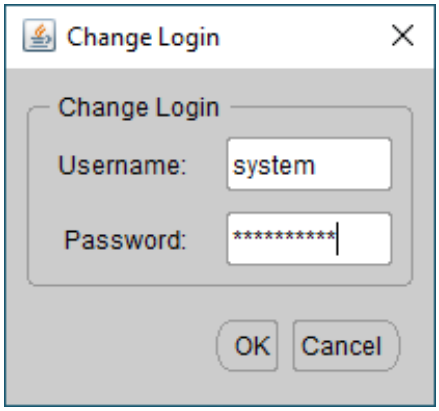

Fig. 26: Change Login

28. The test will be performed again. If the test was successful proceed with »Next«.

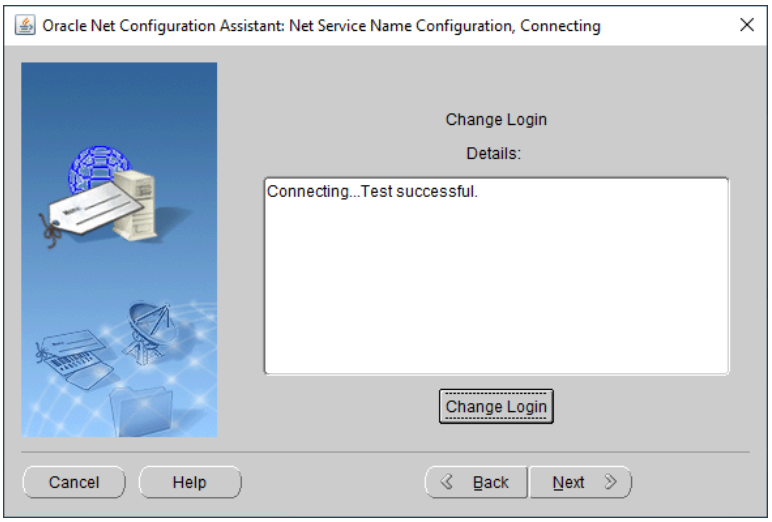

Fig. 27: Successful test result

29. Enter a name for this net service name. By default choose the same name as the database name (e.g. »`adodb`«). Proceed with »Next«.

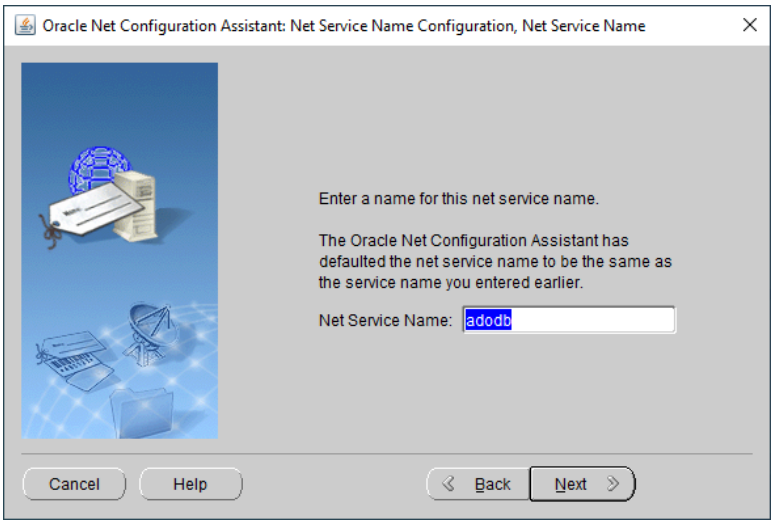

Fig. 28: Enter net service name

30. Select »No« when asked if you want to configure another net service name. Proceed with »Next«.

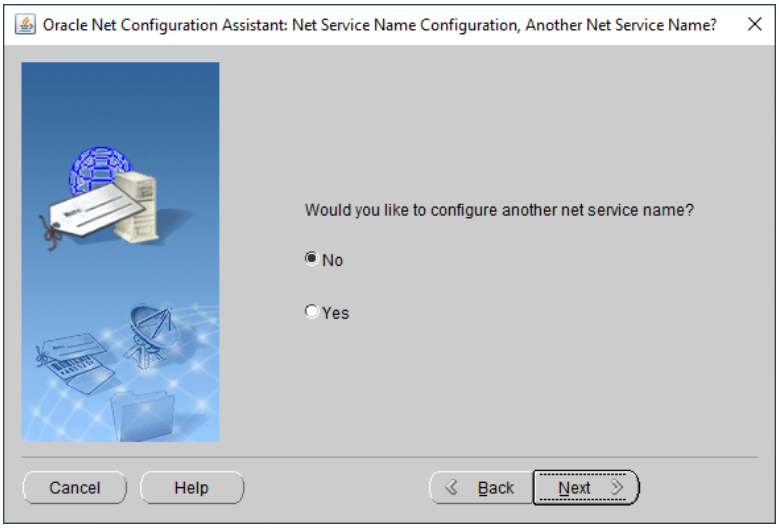

Fig. 29: Configure another Net Service Name

31. A confirmation dialogue appears. Click »Next« to return to the start screen. Click »Cancel«.

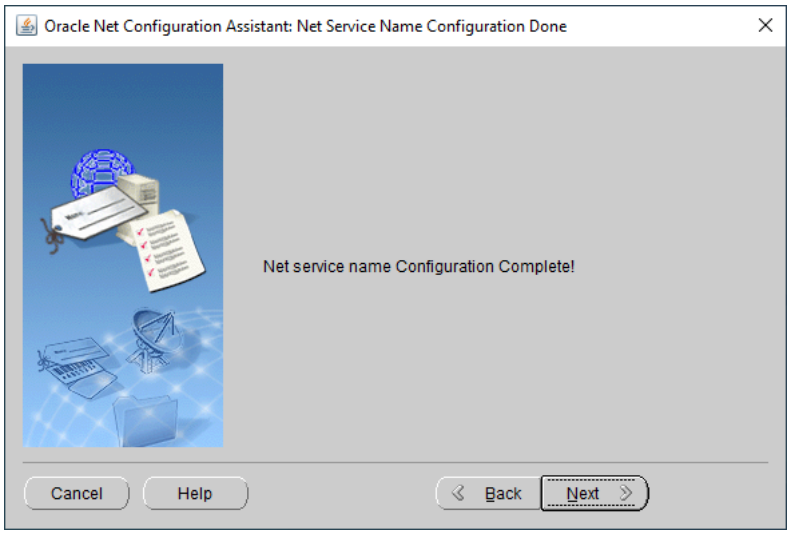

Fig. 30: Finish

The configuration is complete. The Oracle Net Configuration Assistant closes.

Initialise Database

Part 6: Create Tables and Initialise Database

Finally, you need to create tables and initialise the database. This part has to be executed on the machine where the application server is installed.

32. Open the Command Prompt as administrator and navigate to the application server installation directory.

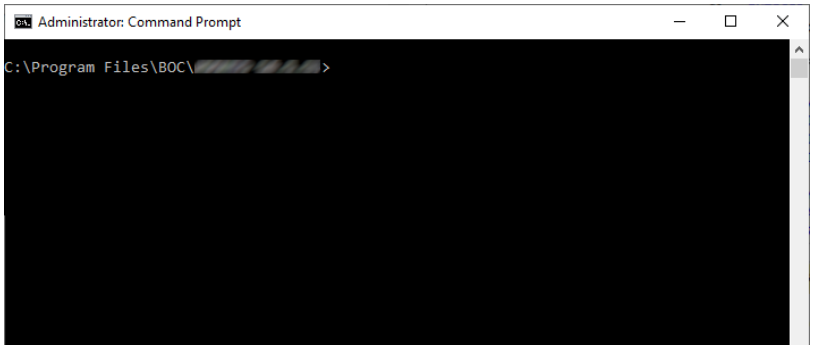

Fig. 31: Open the Command Prompt

33. Adapt the following command with your actual values and execute it:
`.\amain.exe -mode install -dbimode tablesonly -db <my-net-service-name> -dbadmin <my-database-admin-name> -dbadminpw <my-database-admin-password> -dt Oracle -lib "<pathname>\<library file>.axl" -lic "<pathname>\<licence file>.xxl"`

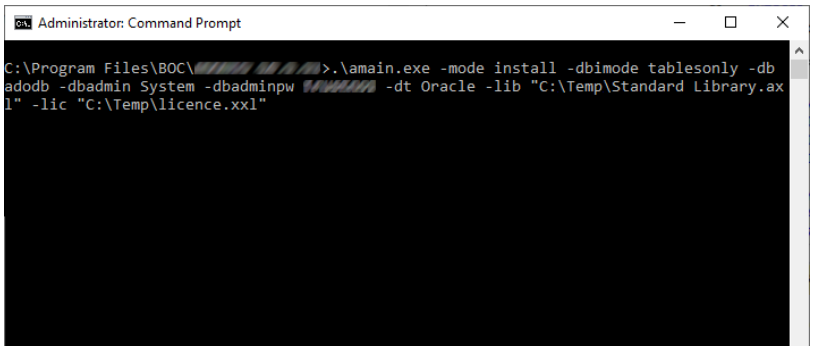

Fig. 32: Create Tables and Initialise Database

The configuration process is completed. You can use the database now.

Placeholders

Here is an explanation of the placeholders in the command:

- **<my-net-service-name>**: The name of the Net Service Name (client connection) to the new database.
- **<my-database-admin-name>**: The username of your database administrator.
- **<my-database-admin-password>**: The password of your database administrator.
- **<pathname>\<library file>.axl**: The full path and filename of the library file. Wrap in quotes.
- <**pathname>\<licence file>.xxl**: The full path and filename of the licence file. Wrap in quotes.

Example:

```
.\amain.exe -mode install -dbimode tablesonly -db adodb -dbadmin System -dbadminpw secret -dt Oracle -lib "C:\Temp\Standard Library.axl" -lic "C:\Temp\licence.xxl"
```

Part 7: Change Password of User »ADOxx« (Optional)

The database user (database login) »ADOxx« is required for login on a BOC Management Office® database. For security reasons, the default password »r0KaQIFA]cPd2Ave«of the user »ADOxx« should be changed after the initialisation of the database. This part has to be executed on the machine where the application server is installed.

34. Open the Command Prompt as administrator and navigate to the application server installation directory.

Fig. 33: Open the Command Prompt

35. Adapt the following command with your actual values and execute it:
`.\amain.exe -mode install -db <my-net-service-name> -dbadmin ADOxx -dbadminpw <old-database-user-password> -dt Oracle -changeownerpw -newpw <new-database-user-password>`

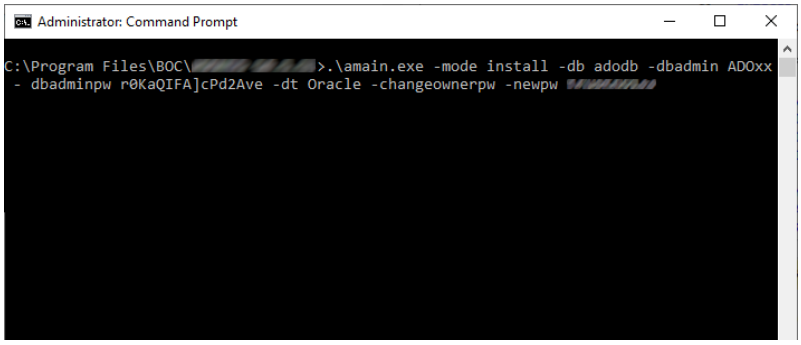

Fig. 34: Change Password of User »ADOxx«

The password of the user »ADOxx« is changed.

Placeholders

Here is an explanation of the placeholders in the command:

- **<my-net-service-name>**: The name of the Net Service Name (client connection) to the new database.

- **<old-database-user-password>**: The current password of the database user »ADOxx«.
- **<new-database-user-password>**: The new password that you want to assign to the database user »ADOxx«.

Example:

```
.\amain.exe -mode install -db adodb -dbadmin ADOxx -dbadminpw r0KaQIFA]cPd2Ave -dt Oracle -changeownerpw -newpw secret
```

Modify the Open Mode of a PDB

PDBs can be in various **open modes**. When a CDB is restarted, the PDB will be in **mounted mode** by default, and no objects can be written to it. Make sure that the PDB is in **read write mode** for table creation (see **Part 3: SQL Plus**) and for running the BOC Management Office® product.

To check and alter the open mode of a PDB, the following SQL statements can be issued in »SQL Plus«.

1) **Check PDB State**

To see what modes the PDBs in a CDB are in, you can use the following command:

```
SQL > show pdbs
```

The current mode of each PDB will be shown in the OPEN MODE column.

2) **Change Open mode of PDB to Read Write**

If necessary, you can change the open mode of a PDB to read write by using the ALTER PLUGGABLE DATABASE command:

```
SQL > alter pluggable database <PDP_NAME> open read write;
```

3) **Save PDB State**

Make sure to preserve the open mode of the PDB so that next time when the CDB restarts it will be in read write mode automatically:

```
SQL > alter pluggable database <PDP_NAME> save state;
```

Database Communication

The necessary communication between the BOC Management Office® product and the database is processed via two database users:

1) **ADOXX_BOOT**

An auxiliary user. During login, »ADOXX_BOOT« connects to the database (CONNECT rights) and retrieves the encrypted user name/password combination of the actual DB user »ADOxx« (SELECT rights on one single configuration table). »ADOXX_BOOT« has no further rights and cannot decrypt the password.

2) **ADOxx**

The database user in whose context the DB session is established. The entire database communication of the application is processed in the context of this user. For

this, »ADOxx« has sufficient rights on the tables of the BOC Management Office® product. This user can be replaced by an individual database user.

Variants

Create Database with an Individual DB User

When you create a database manually, you can replace the default database user »ADOxx« by any other user name/password combination. In order to do so, follow the same steps as described above, with the following exceptions:

Part 2: Script » aprporcl.sql «

- Adapt the script »`aprporcl.sql`«:
  - Replace the user »ADOxx« with your user (e.g. »ADOuser«).
  - Replace the password »r0KaQIFA]cPd2Ave« with your password.
  - Create a synonym »ADOXX_BOOT.dbinfo« for the table <individual DB User>.dbinfo. For example, replace »ADOxx.dbinfo« with »ADOuser.dbinfo«.

Part 6: Create Tables and Initialise Database

- You need to append the command line call with the name and password of the database user. Replace the placeholders with your actual values and execute the following command in the Command Prompt:

```
.\amain.exe -mode install -dbimode tablesonly -db <my-net-service-name> -dbadmin System -dbadminpw <my-database-admin-password> -dt Oracle -lib "<pathname>\<library file>.axl" -lic "<pathname>\<licence file>.xxl" -customschema -schemaowner <my-custom-user-name> -schemaownerpw <my-custom-user-password>
```

Part 7: Change Password of User »ADOxx« (Optional)

- Skip this step. Since you are using an individual user instead of the default database user »ADOxx«, and this individual user was just created with a new password, it is not necessary to change the password again immediately.

Drop Tables: Preparing a Database for Reuse

Instead of creating a new database, you can reuse an existing BOC Management Office® database for a new product version. This is especially useful when you only have access to one database (e.g. because of specific organisational restrictions).

To reuse a database, you need to drop all product specific tables, and then initialise the database with the new product version. Settings such as the created database users (see **Database Communication**), schemas and permissions are retained. All table data (models and objects and their relations, the user accounts and the defined user rights) are permanently deleted.

The tables can be dropped with the script »`oracle_dropadoxxtables.sql`«. You can find the script in the installation package of your BOC Management Office® product in the folder »`02 Application Server\dbinfo\oracle`«.

Prerequisites
Before you begin:

- We strongly recommend creating a backup of the database.
- Export the library-specific component settings and a migration package as described in the **Administration Help.** This step is necessary because the data will be re-imported into the database later.

Prepare Database for Reuse
In order to prepare an Oracle database for reuse:

1. Execute the script »`oracle_dropadoxxtables.sql`«.
    a. Open a command prompt, and then enter the command »`sqlplus.exe system@<DB name>`«. Replace `<DB name>` with the name of your new database. When prompted, enter the password previously defined for the database user »`system`« in step **13**.
    b. Run the script with the command »`start <pathname>\oracle_dropadoxxtables.sql`«. `<pathname>` is the path where the script is located.
    c. If the last step was successful, close SQL Plus with »`exit`«.
2. Create new tables and initialize the database for the BOC Management Office® as described in **Part 6: Create Tables and Initialise Database**.

The database is ready for use. Now you can perform the migration from the earlier version of the BOC Management Office® product to the new version as described in the **Installation Manual**.

Note: If you are using an individual DB User (see **Create Database with an Individual DB User**), you have to adapt the script »`oracle_dropadoxxtables.sql`« before using it. Replace all occurrences of »ADOxx« with your user (e.g. »ADOuser«).

Appendix

Required Database Permissions

This section describes the database user permissions that are required by BOC Management Office® products to run properly.

1) ADOXX_BOOT

The user »ADOXX_BOOT« must have the following permissions:

- CONNECT
- SELECT ON ADOxx.dbinfo

2) ADOxx

The user »ADOxx« must have the following permissions:

- CONNECT

Since all objects (tables, indexes) are created in the ADOxx user's schema, no explicit privileges need to be granted and they also cannot be revoked.

3) Database Administrator (e.g. ‘system’)

The database administrator user must have the following permissions.

To create the database:

- SYSDBA system privilege

To execute the script 'aprporcl.sql':

- CONNECT WITH ADMIN OPTION;
- CREATE TABLESPACE, CREATE PROFILE, CREATE USER, ALTER USER;
- CREATE ANY SYNONYM;

To create the ADOxx database schema:

- CREATE ANY TABLE, ALTER ANY TABLE, CREATE ANY INDEX, CREATE ANY TRIGGER;
- GRANT ANY OBJECT PRIVILEGE;