

Appendix N

Creating a New Database Using Oracle 12c

Preview

Although the general database creation format tends to be generic, its execution tends to be DBMS-specific. The leading RDBMS vendors offer the DBA the option to create databases manually, using SQL commands or using a GUI-based process. Which option is selected depends on the DBA's sense of control and style.

Using the Oracle Database Configuration Assistant, it is simple to create a database. The DBA uses a wizard interface to answer a series of questions to establish the parameters for the database to be created. Other than the name of the database and passwords for administrator accounts, in most cases, the wizard will suggest common options and default values that produce a well-configured database. Figures N.1 through N.17 show you how to create a database with the help of the Oracle Database Configuration Assistant.

Data Files and Available Formats

MS Access **Oracle** **MS SQL** **My SQL**

MS Access **Oracle** **MS SQL** **My SQL**

There are no data files for this appendix.

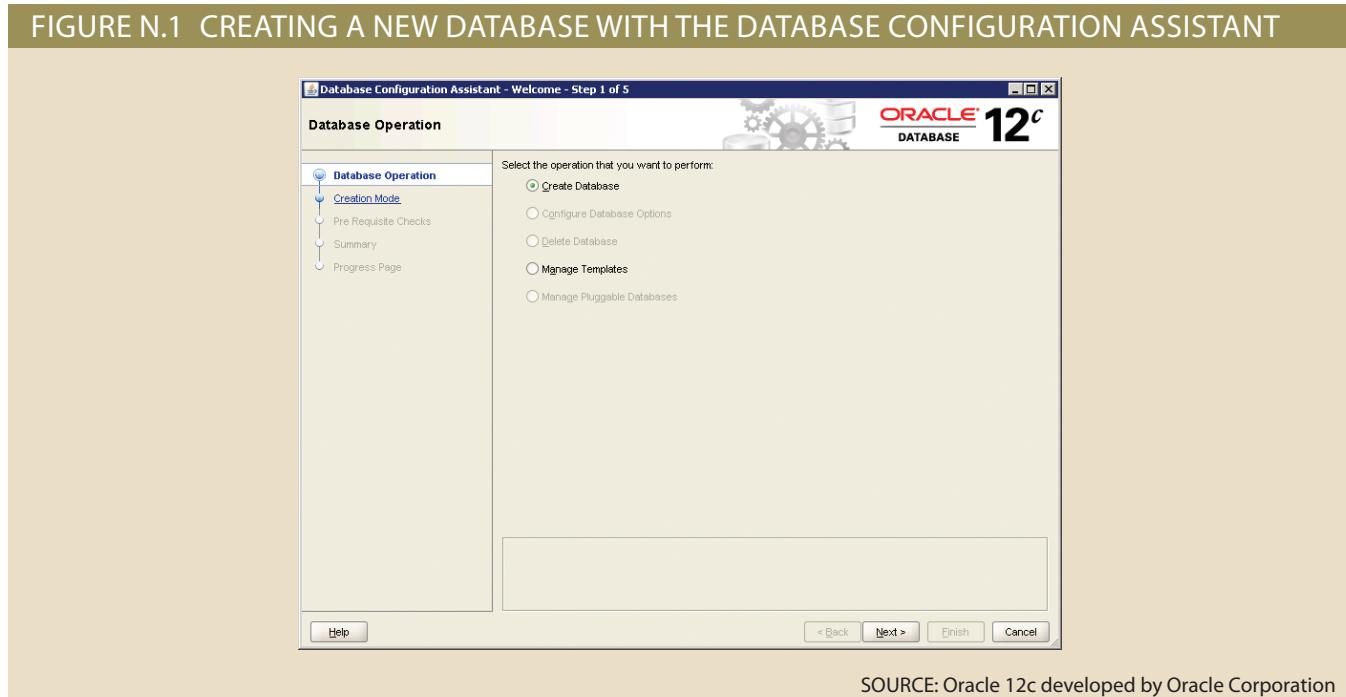
Data Files Available on cengagebrain.com

N-1 Creating a Standalone Database

Creating a single instance database in Oracle 12c is the equivalent of creating a database in Oracle 11g. It creates a standalone database that uses its own set of memory structures and background processes. Oracle has simplified the creation of a standalone database by improving default value selections so that the database can be created by specifying just the global database name and providing appropriate passwords.

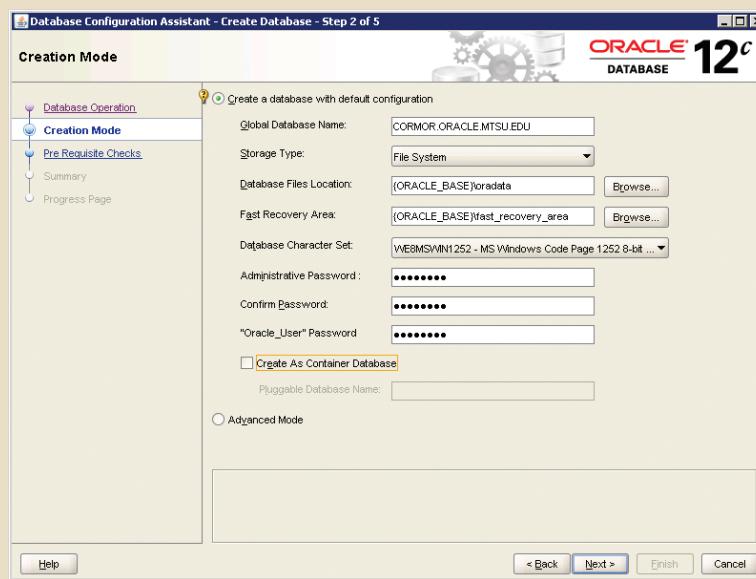
After confirming the database options selected in Figures N.1 through N.3, the database creation process starts as shown in Figure N.4. This process creates the database structure, including the necessary data dictionary tables, the administrator user accounts, and other supporting processes required by the DBMS to manage the database. Figure N.5 illustrates the completion of the database creation process. Depending on the capabilities of the computing platform on which the database is being installed, the configuration options chosen, and whether or not sample schemas are to be loaded into the database, the creation process can be lengthy.

FIGURE N.1 CREATING A NEW DATABASE WITH THE DATABASE CONFIGURATION ASSISTANT



SOURCE: Oracle 12c developed by Oracle Corporation

FIGURE N.2 NAMING THE DATABASE AND SETTING PASSWORDS



SOURCE: Oracle 12c developed by Oracle Corporation

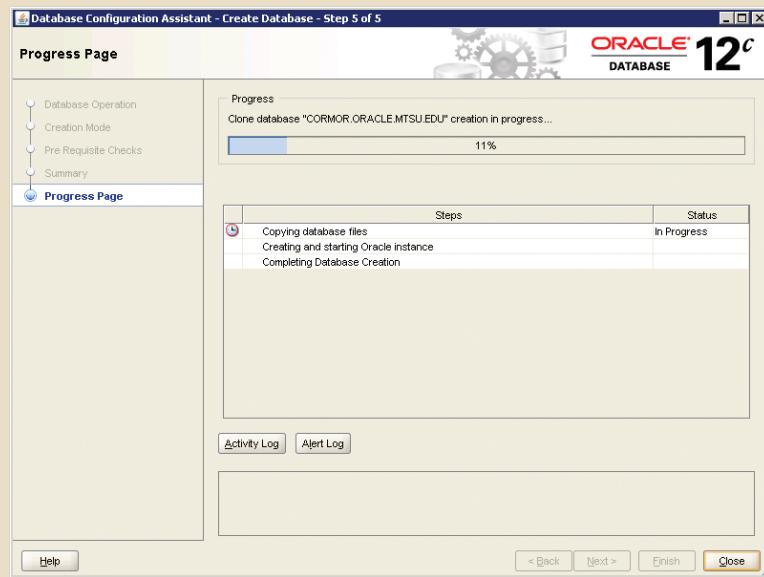
FIGURE N.3 REVIEW THE CREATION SUMMARY



SOURCE: Oracle 12c developed by Oracle Corporation

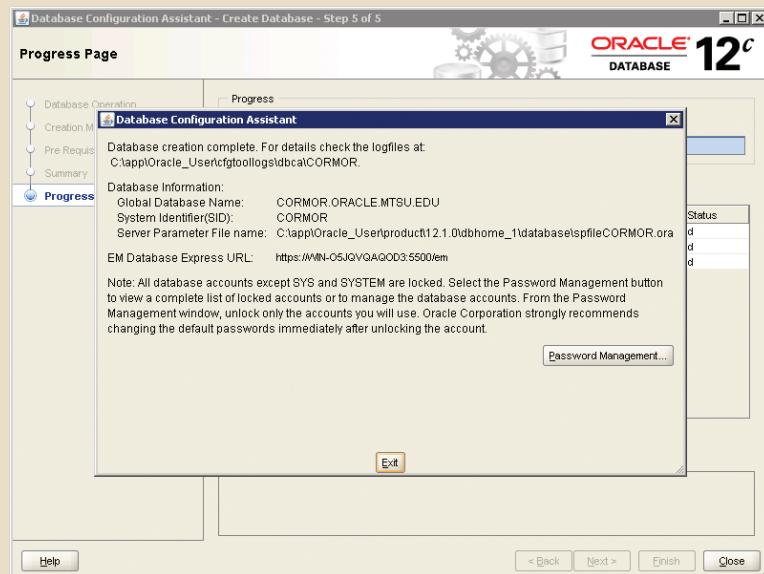
N-4 Appendix N

FIGURE N.4 DATABASE CREATION IN PROGRESS



SOURCE: Oracle 12c developed by Oracle Corporation

FIGURE N.5 DATABASE CREATION COMPLETE



SOURCE: Oracle 12c developed by Oracle Corporation

N-2 Creating a Multitenant Database

Oracle 12c introduced the concept of multitenant databases. Simplified, a multitenant database is a database that contains other databases. In a **multitenant database** environment, a container database is created that contains one or more pluggable databases. A **pluggable database** is simply a database that is contained in a database container. It looks and acts just like a standalone database to applications and users. These databases are called pluggable because they can be unplugged from one container database and easily plugged into another container database with relative ease. This makes it easy for a DBA to move a database from one server to another. Imagine, for example, a database used to support income tax preparation for an accounting firm. During most of the year, the database may get little use. As the income tax due date approaches, the database may become extremely busy. As a pluggable database, the DBA could keep the database plugged into a container on a low-cost, less powerful server most of the year, and then move it to a powerful server during its peak usage times. All of these changes in location would be transparent to the users and applications, and would require little effort for the DBA.

Another advantage of multitenant databases is that they consolidate database infrastructure. All of the pluggable databases in a container share the same memory structures and background processes. If a company has 50 standalone databases running on a single server, then each database must be managed independently in terms of creating backups, applying patches, monitoring performance, and so on. Further, each standalone database would be running its own set of processes to manage the database. With the multitenant database, all pluggable databases in the container can be managed simultaneously through the container database. Patches, backups, and performance statistics are all performed and aggregated through the container. Applying a patch to the container database applies the patch to all of the pluggable databases in that container. Thus, the DBA's task is greatly simplified and server resources are more efficiently utilized.

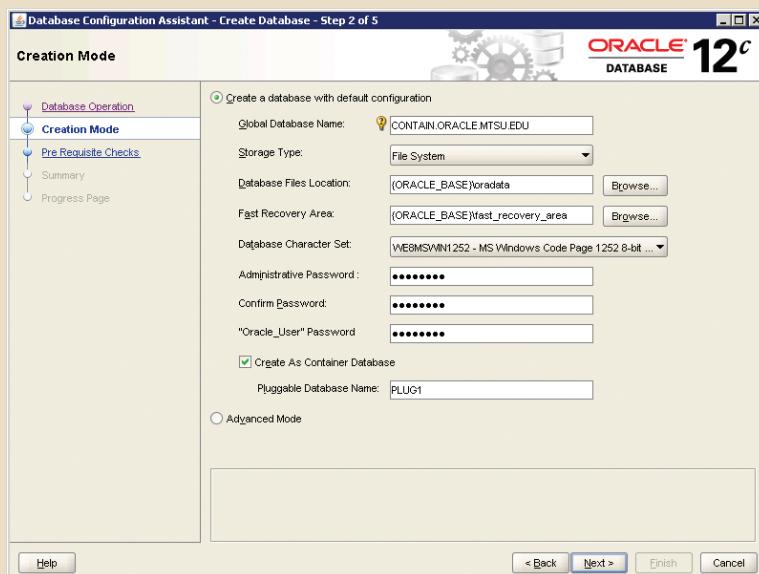
multitenant database

A database environment in which a container database can hold other databases.

pluggable database

In a multitenant database environment, a database that can be contained within a container database.

FIGURE N.6 CREATING A CONTAINER DATABASE FOR MULTITENANT DATABASES



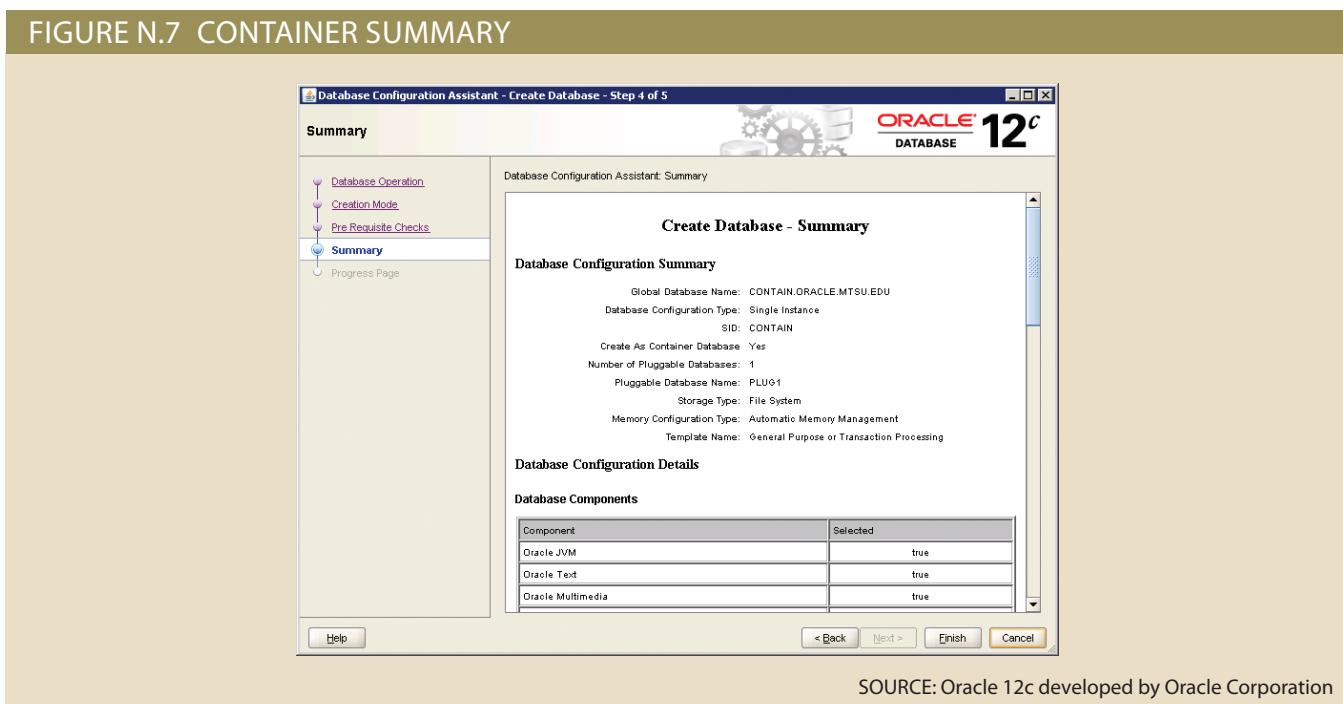
SOURCE: Oracle 12c developed by Oracle Corporation

Notice that in Figure N.6 the container is created with one pluggable database, PLUG1, already installed.

N-6 Appendix N

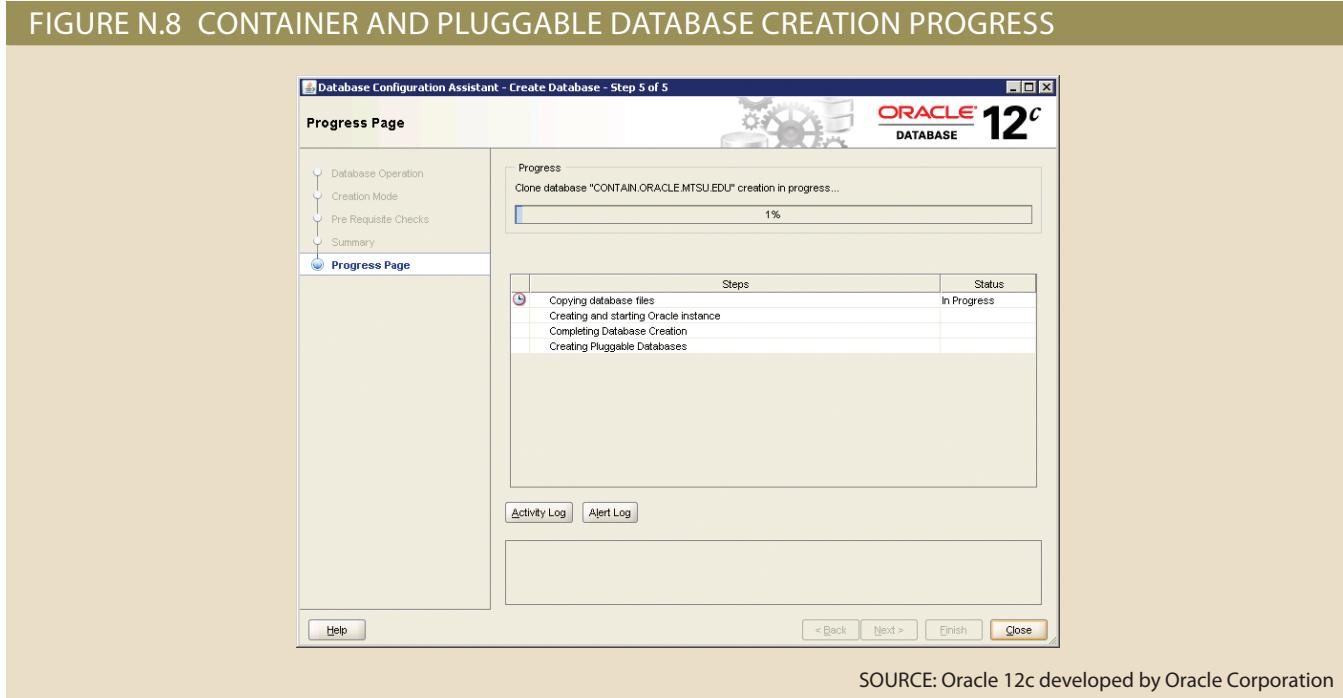
Figures N.7 and N.8 show the creation of the container database and the first pluggable database. Figure N.9 shows the successful creation of both databases—the container database, CONTAIN, and the pluggable database, PLUG1.

FIGURE N.7 CONTAINER SUMMARY



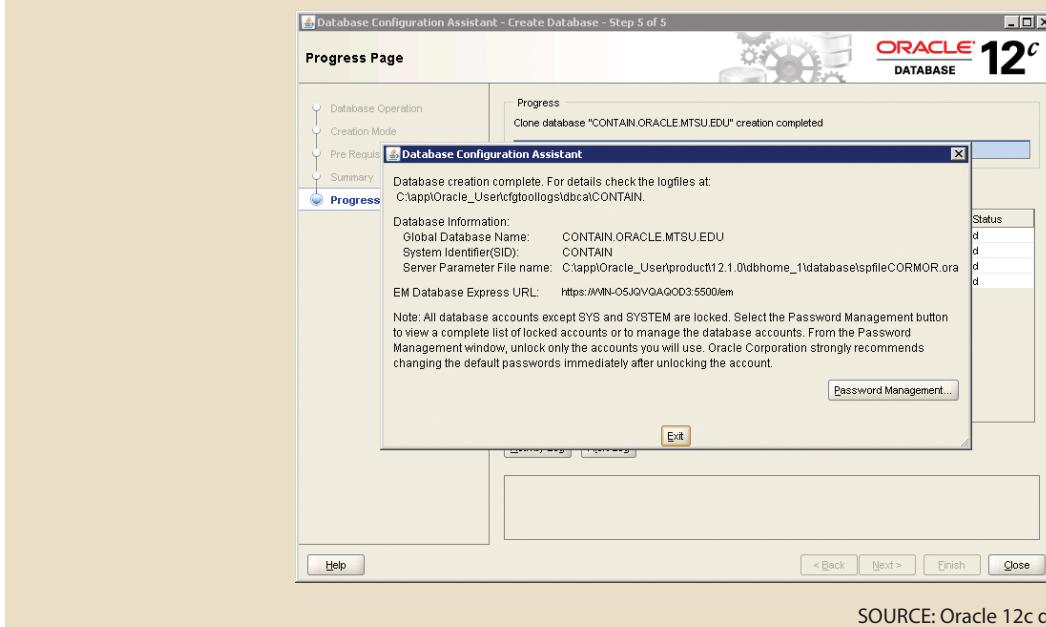
SOURCE: Oracle 12c developed by Oracle Corporation

FIGURE N.8 CONTAINER AND PLUGGABLE DATABASE CREATION PROGRESS



SOURCE: Oracle 12c developed by Oracle Corporation

FIGURE N.9 CONTAINER AND PLUGGABLE DATABASES CREATION COMPLETE



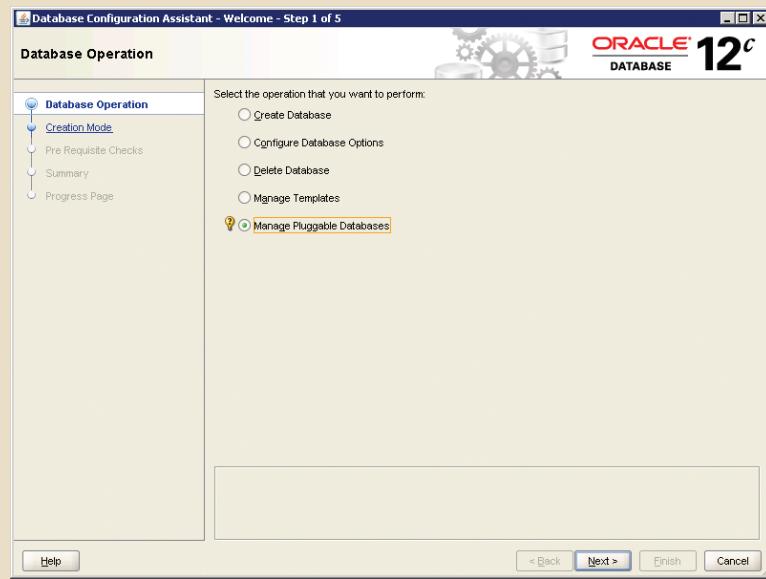
SOURCE: Oracle 12c developed by Oracle Corporation

N-3 Managing Pluggable Databases

Pluggable databases can be managed through the Database Configuration Assistant (DBCA). Using the DBCA, new pluggable databases can be created or deleted, and existing pluggable databases can be unplugged from a container or plugged into a container. Figures N.10 through N.17 illustrate the creation of a new pluggable database, PLUG2, in the CONTAIN container database created previously.

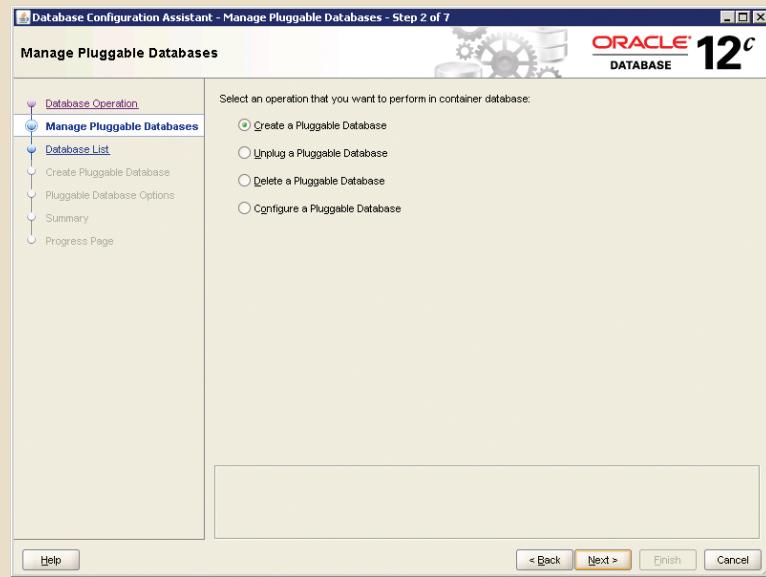
N-8 Appendix N

FIGURE N.10 MANAGING PLUGGABLE DATABASES



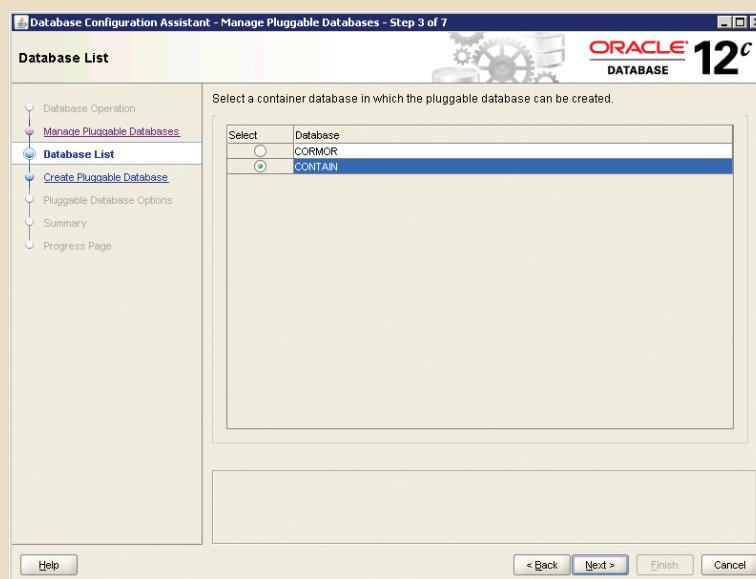
SOURCE: Oracle 12c developed by Oracle Corporation

FIGURE N.11 CREATING A NEW PLUGGABLE DATABASE



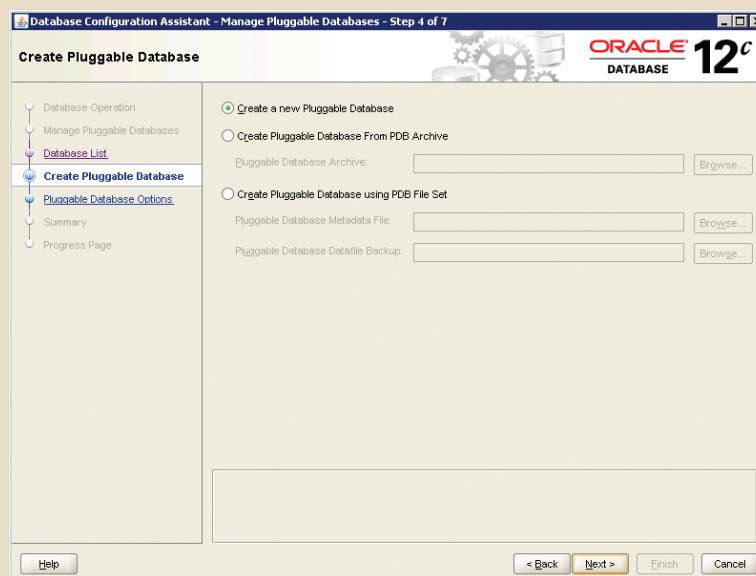
SOURCE: Oracle 12c developed by Oracle Corporation

FIGURE N.12 SPECIFYING A CONTAINER DATABASE



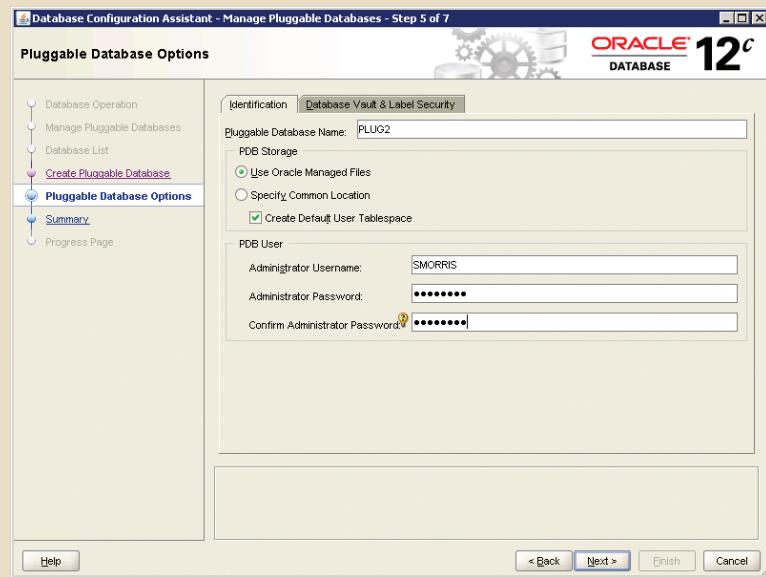
SOURCE: Oracle 12c developed by Oracle Corporation

FIGURE N.13 CHOOSING A CREATION SOURCE



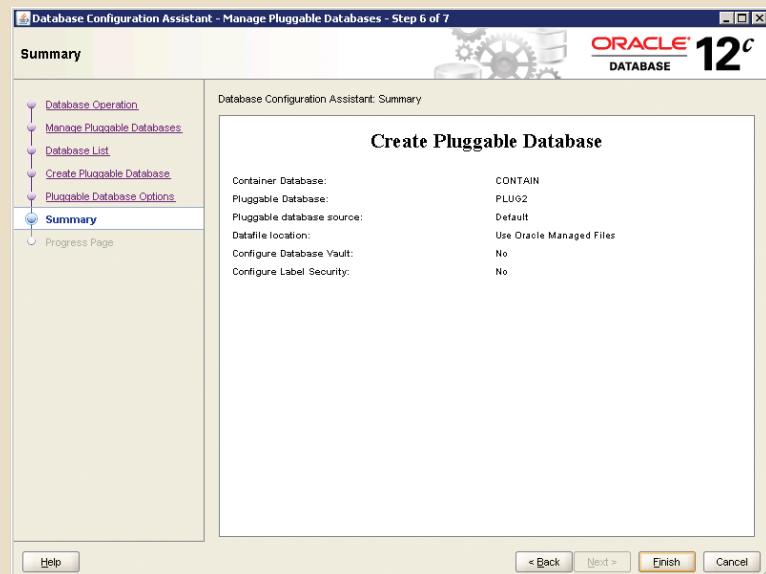
SOURCE: Oracle 12c developed by Oracle Corporation

FIGURE N.14 NAMING THE NEW DATABASE AND SPECIFYING AN ADMINISTRATOR



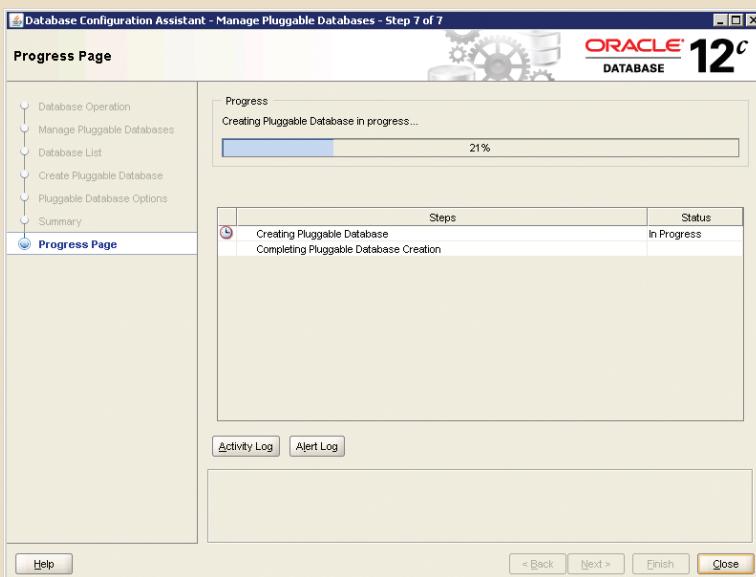
SOURCE: Oracle 12c developed by Oracle Corporation

FIGURE N.15 REVIEW THE PLUGGABLE DATABASE OPTIONS



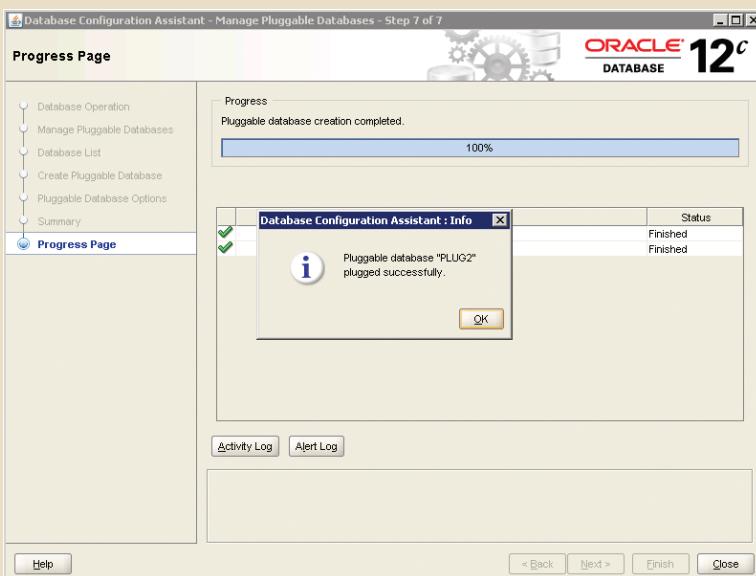
SOURCE: Oracle 12c developed by Oracle Corporation

FIGURE N.16 NEW PLUGGABLE DATABASE CREATION



SOURCE: Oracle 12c developed by Oracle Corporation

FIGURE N.17 NEW PLUGGABLE DATABASE COMPLETE



SOURCE: Oracle 12c developed by Oracle Corporation

Creation of the new pluggable database is much faster than the creation of a stand-alone or container database. The pluggable database uses the memory structures and background processes of the container database, so there are fewer components to create and few processes and services to start on the database server.

Key Terms

multitenant database, N-5

pluggable database, N-5