

Using Oracle Streams to Maintain Tablespace Data

Purpose

This module describes how to use Oracle Streams to maintain tablespace data.

Topics

This module will discuss the following topics

- [Overview](#)
- [Prerequisites](#)
- [Creating a Source and Destination Database using DBCA](#)
- [Creating the Tablespace to be Transported on the Source Database](#)
- [Creating the Users at the Source and Destination Databases](#)
- [Setting Up Streams on the Destination Database](#)
- [Setting Up Streams on the Source Database](#)
- [Maintaining Tablespaces](#)

 **Move your mouse over this icon to show all screenshots. You can also move your mouse over each individual icon to see only the screenshot associated with it.**

Overview

[Back to List](#)

Leveraging the Oracle Transportable Tablespaces and Oracle Streams, the Oracle database offers an efficient way for migrating applications to the Grid. With a single command, the database administrator can identify a set of tablespaces from one database, ship the tablespace set to another database even if the second database is on a different operating system or platform, and plug this set into the second database. During this time, both the source and destination databases are open and available for any user activity. Meanwhile, Oracle Streams has begun to capture any changes from the source database that occur during the tablespace copy to the replica database. After the tablespace set is available at the replica database, the replica database is synchronized by Oracle Streams with the changes from the source database. All of this is done with a single command with no downtime required.

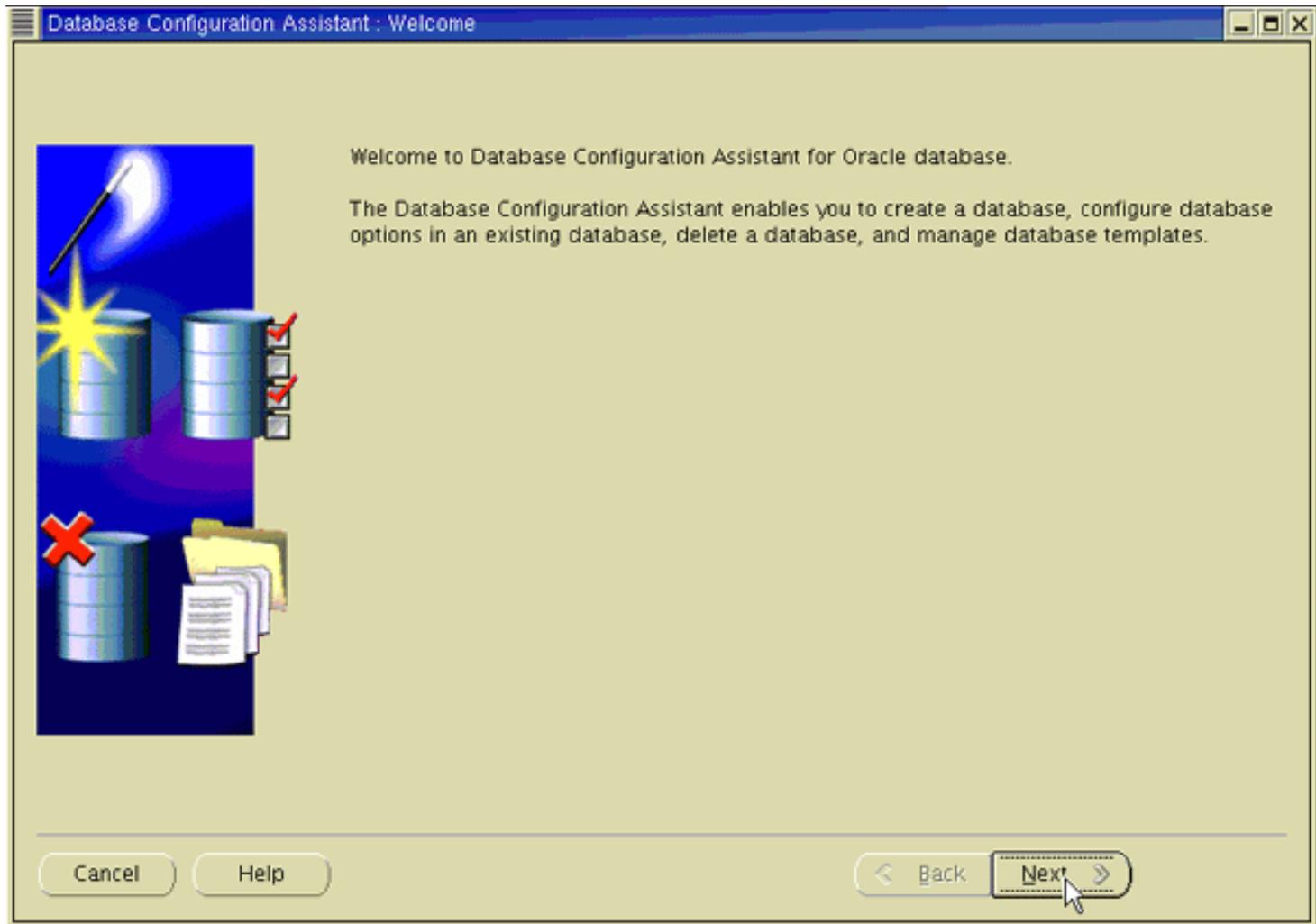
Creating a Source and Destination Database using DBCA

This lesson requires two databases: a source database (SITE1) and a destination database (SITE2). Perform the following steps:

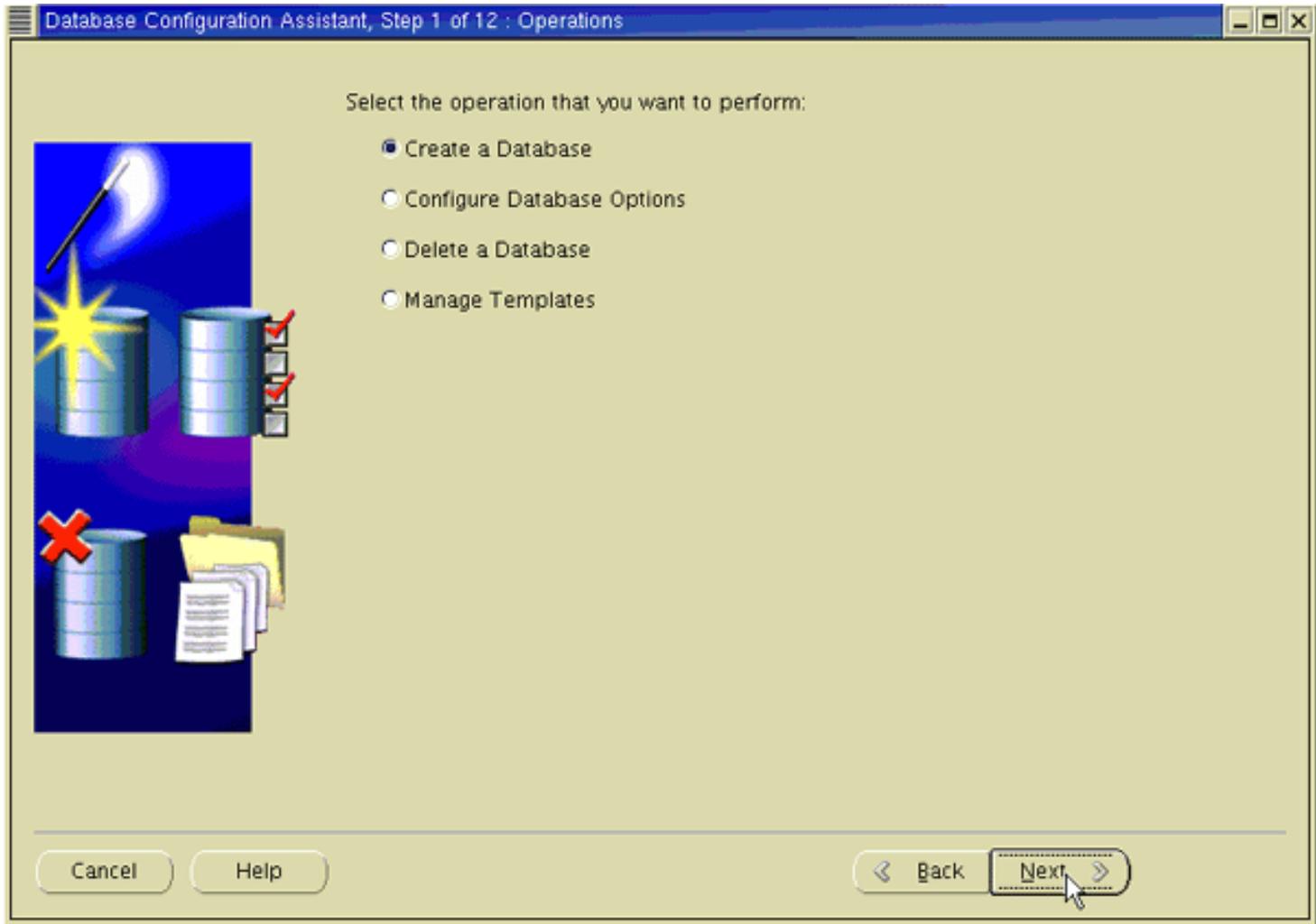
1. From a terminal window, execute the following command:

```
dbca
```

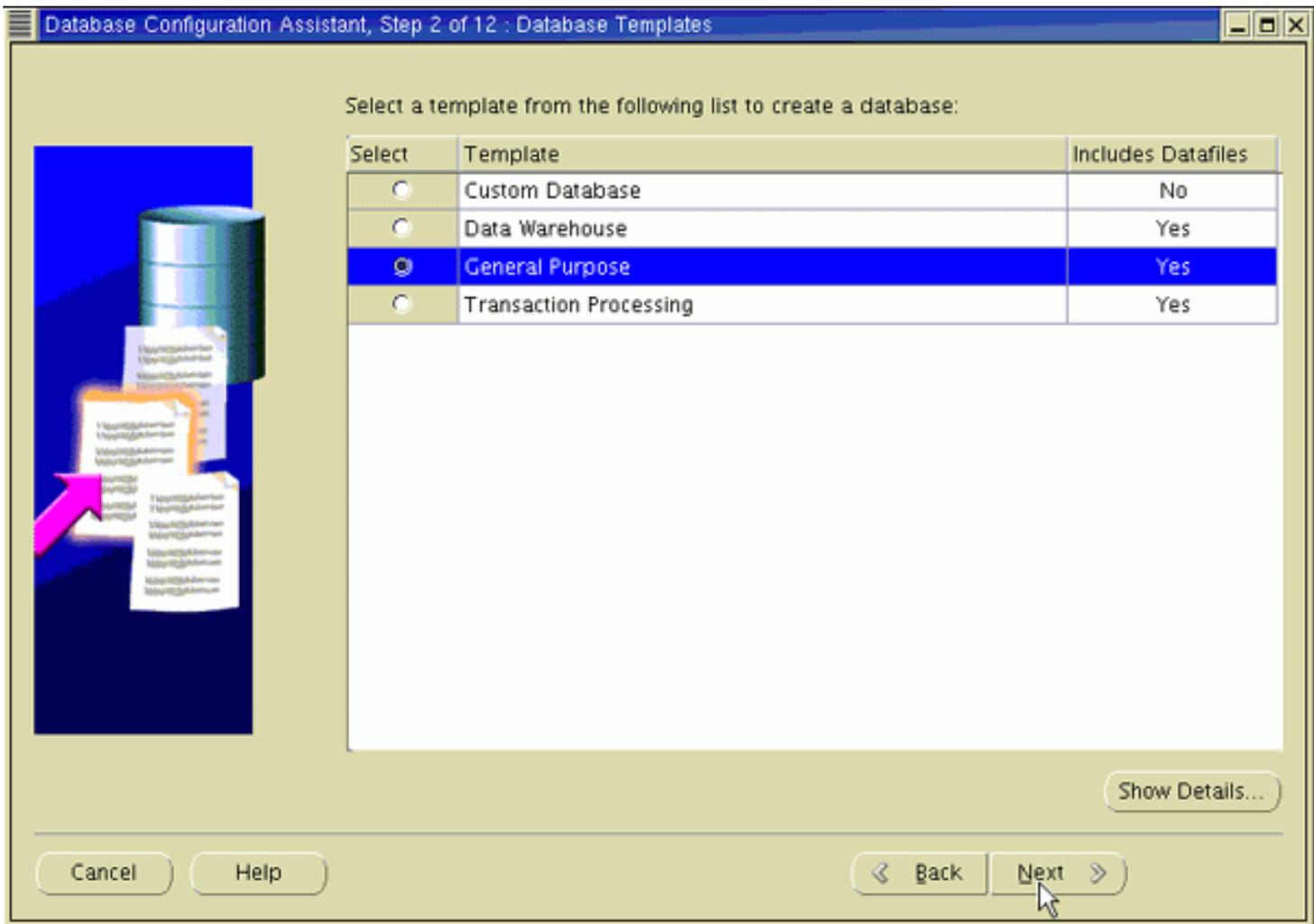
2. Click **Next** .



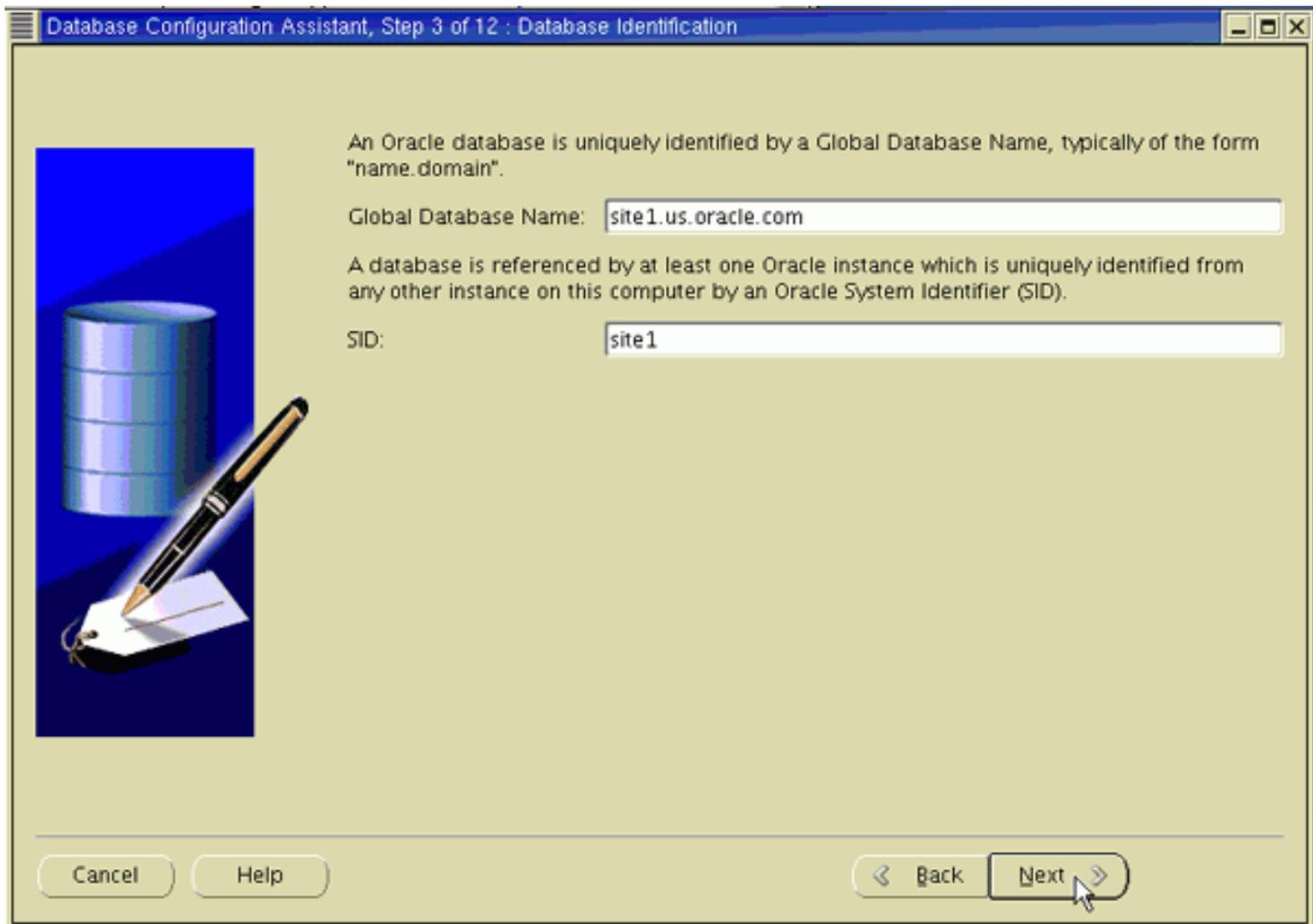
3. Make sure **Create a Database** is selected and click **Next** .



4. Select **General Purpose** and click **Next** .



5. Enter **site1.us.oracle.com** for the Global Database Name and **site1** for the SID and click **Next** .



6. For Management Options, you want to select **Use Database Control for Database Management** and click **Next** .

Database Configuration Assistant, Step 4 of 12 : Management Options

Each Oracle database may be managed centrally using the Oracle Enterprise Manager Grid Control or locally using the Oracle Enterprise Manager Database Control. Choose the management option that you would like to use to manage this database.

Configure the Database with Enterprise Manager

Use Grid Control for Database Management

Select the Management Service:

Use Database Control for Database Management

Enable Email Notifications

Outgoing Mail (SMTP) Server:

Email Address:

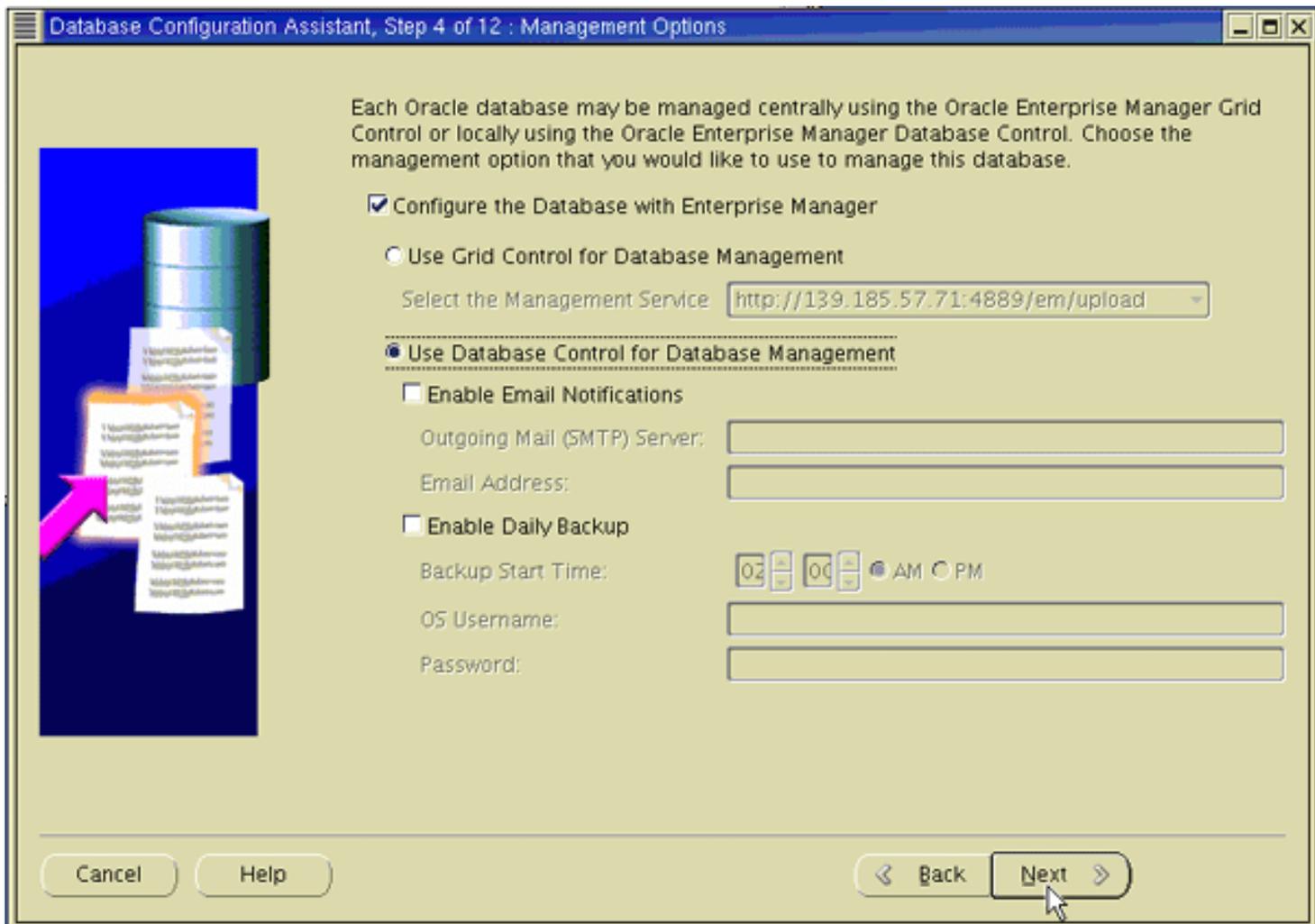
Enable Daily Backup

Backup Start Time: AM PM

OS Username:

Password:

Cancel Help < Back Next >



7. Enter **oracle** for password field for both the SYS and SYSTEM users and click **Next**.

Database Configuration Assistant, Step 5 of 12 : Database Credentials

For security reasons, you must specify passwords for the following user accounts in the new database.

Use the Same Password for All Accounts

Password:

Confirm Password:

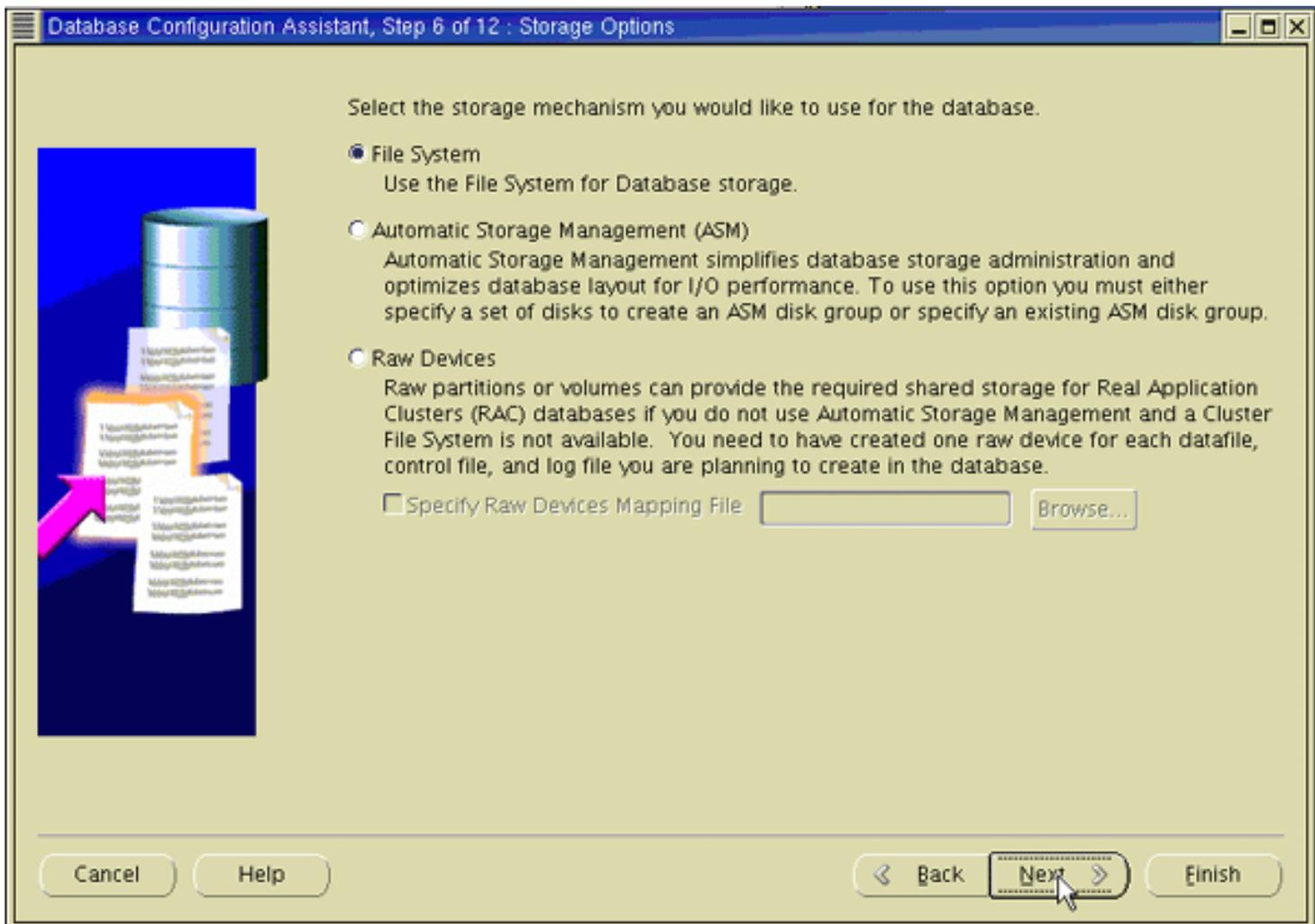
Use Different Passwords

User Name	Password	Confirm Password
SYS		
SYSTEM		
DBSNMP		
SYSMAN		

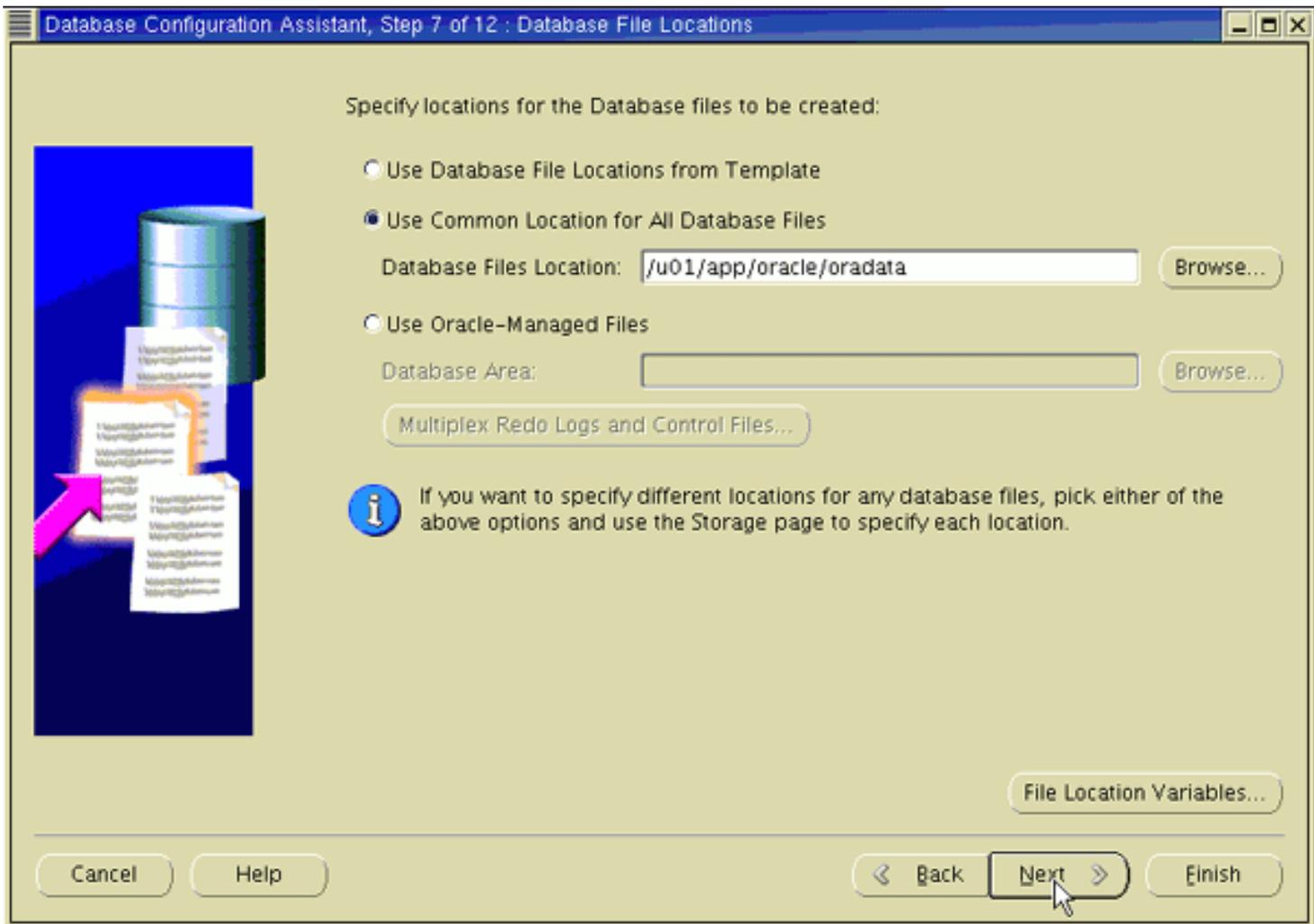
Cancel Help Back Next



8. Make sure **File system** is selected and click **Next** .

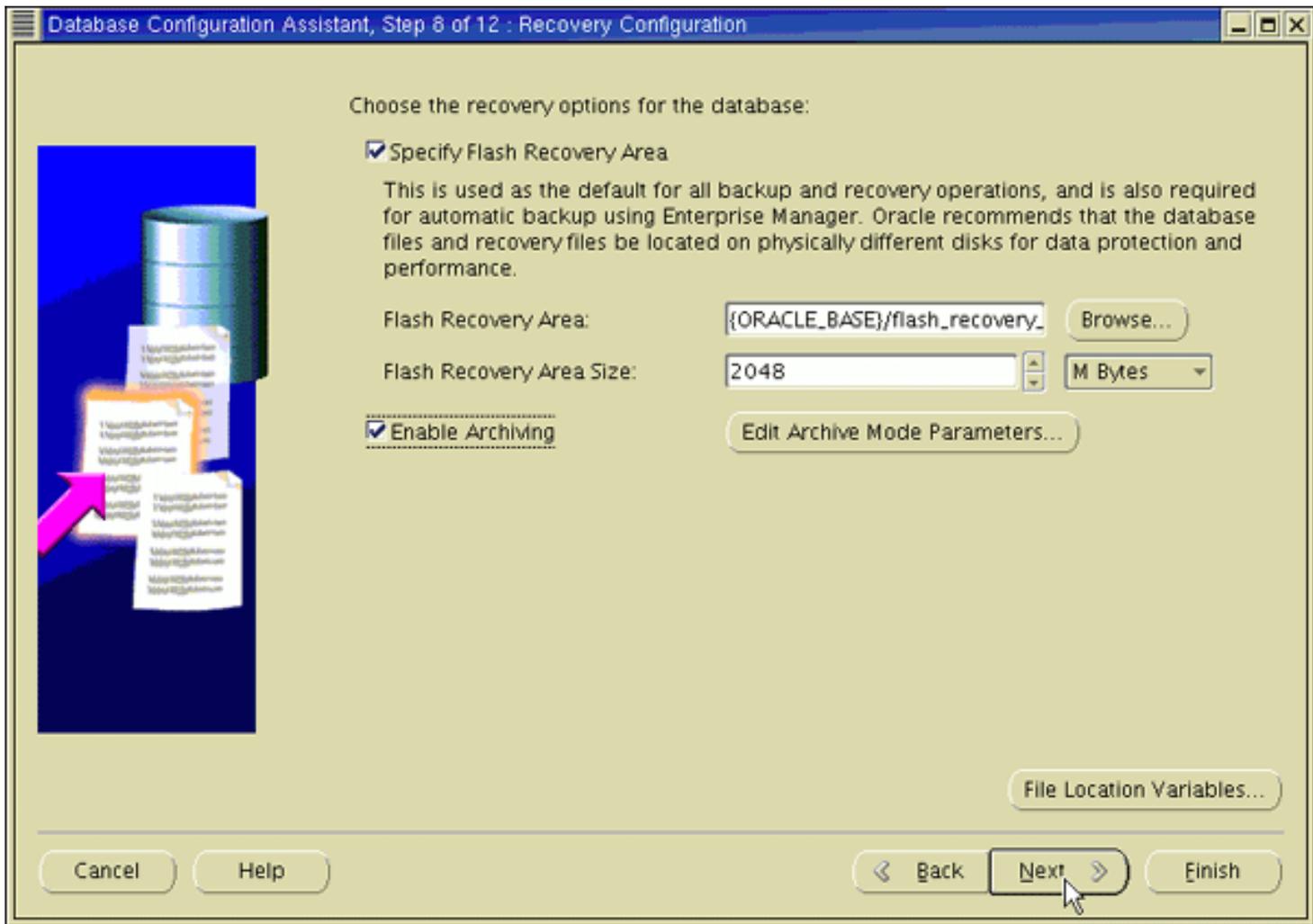


9. Click the **Use Common Location for All Database Files** and enter `/u01/app/oracle/oradata` in the Database Area field. Click **Next**.



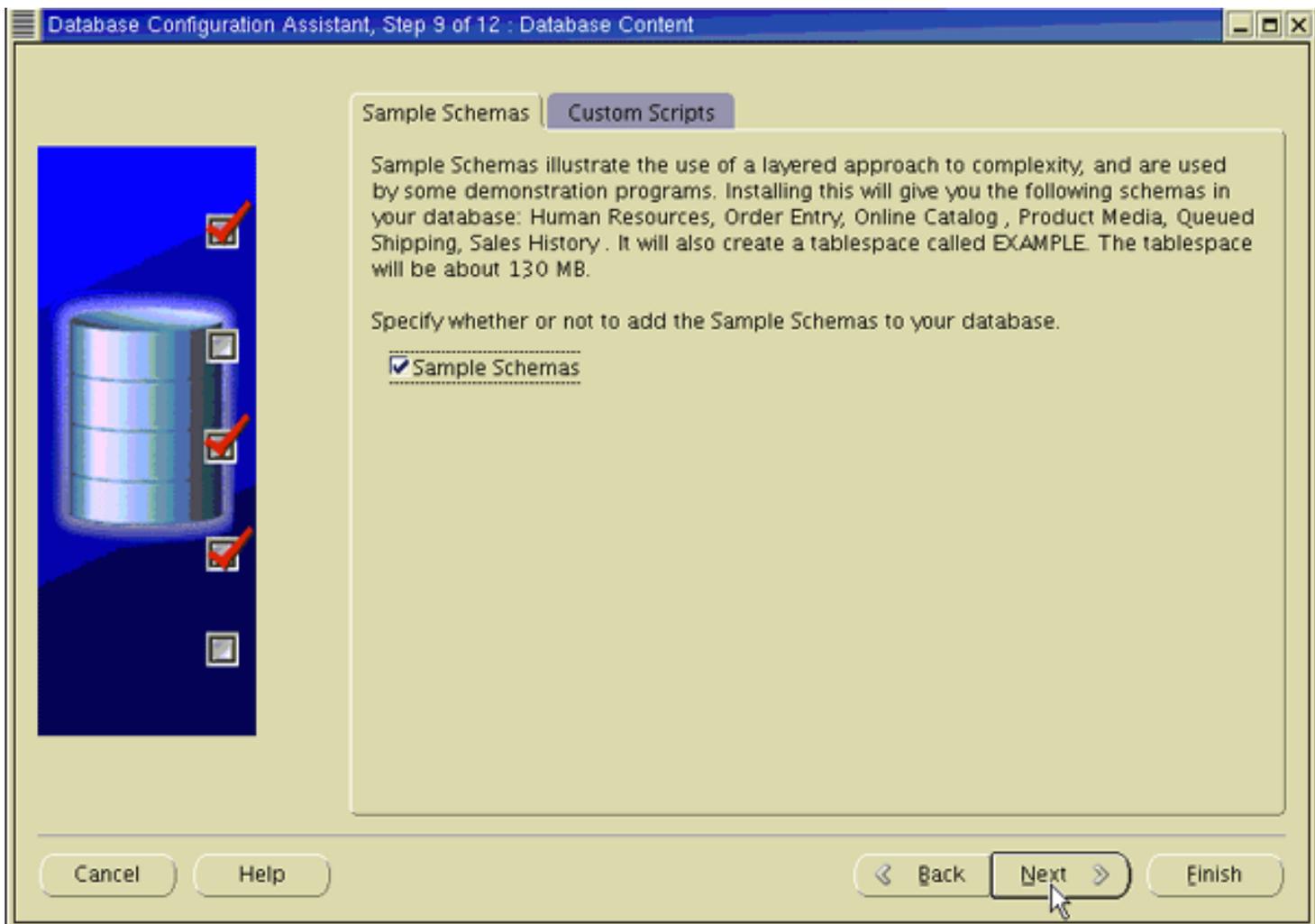
Click **Enable Archiving** and click **Next** .

10.



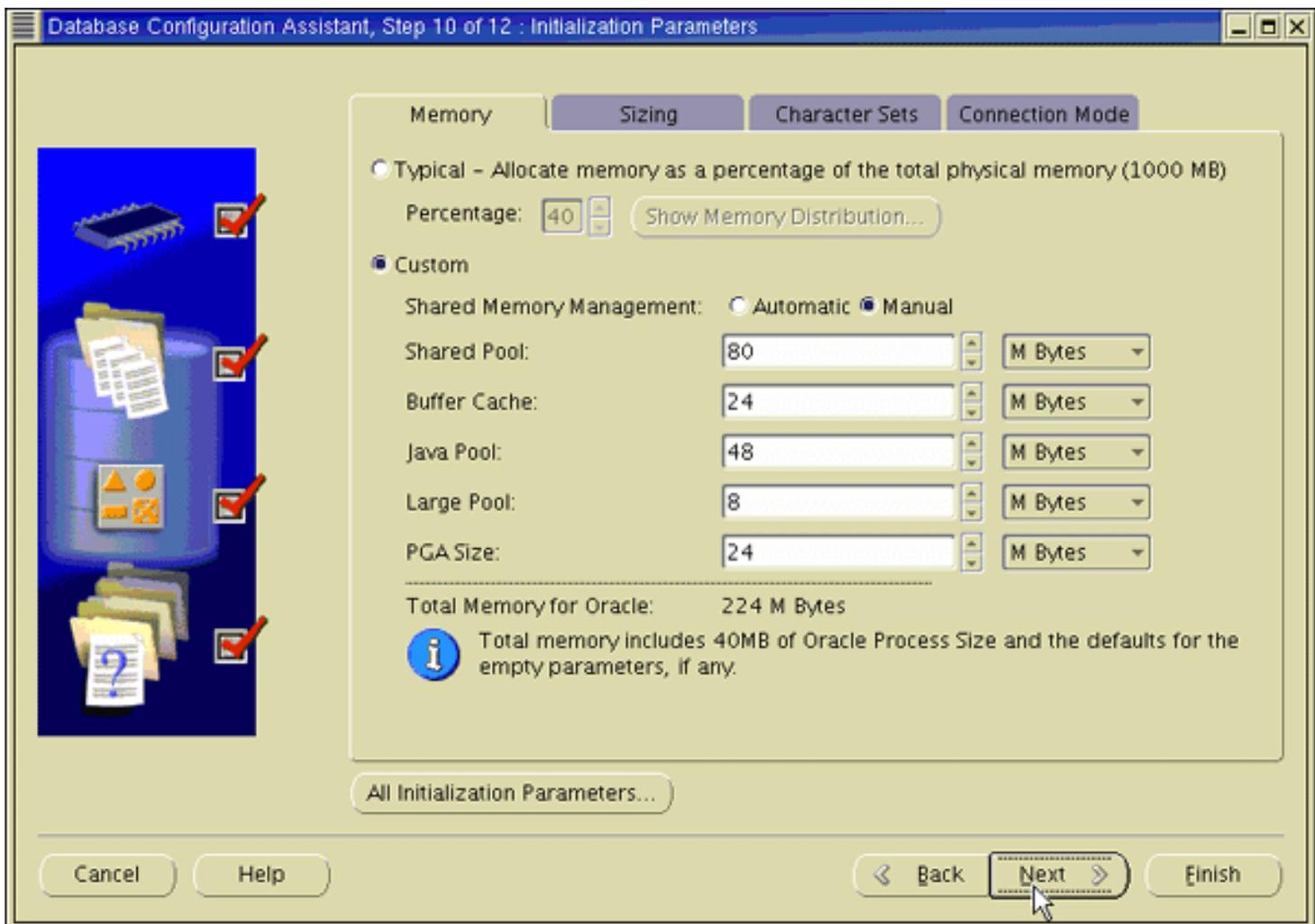
Select **Sample Schemas** and click **Next** .

11.



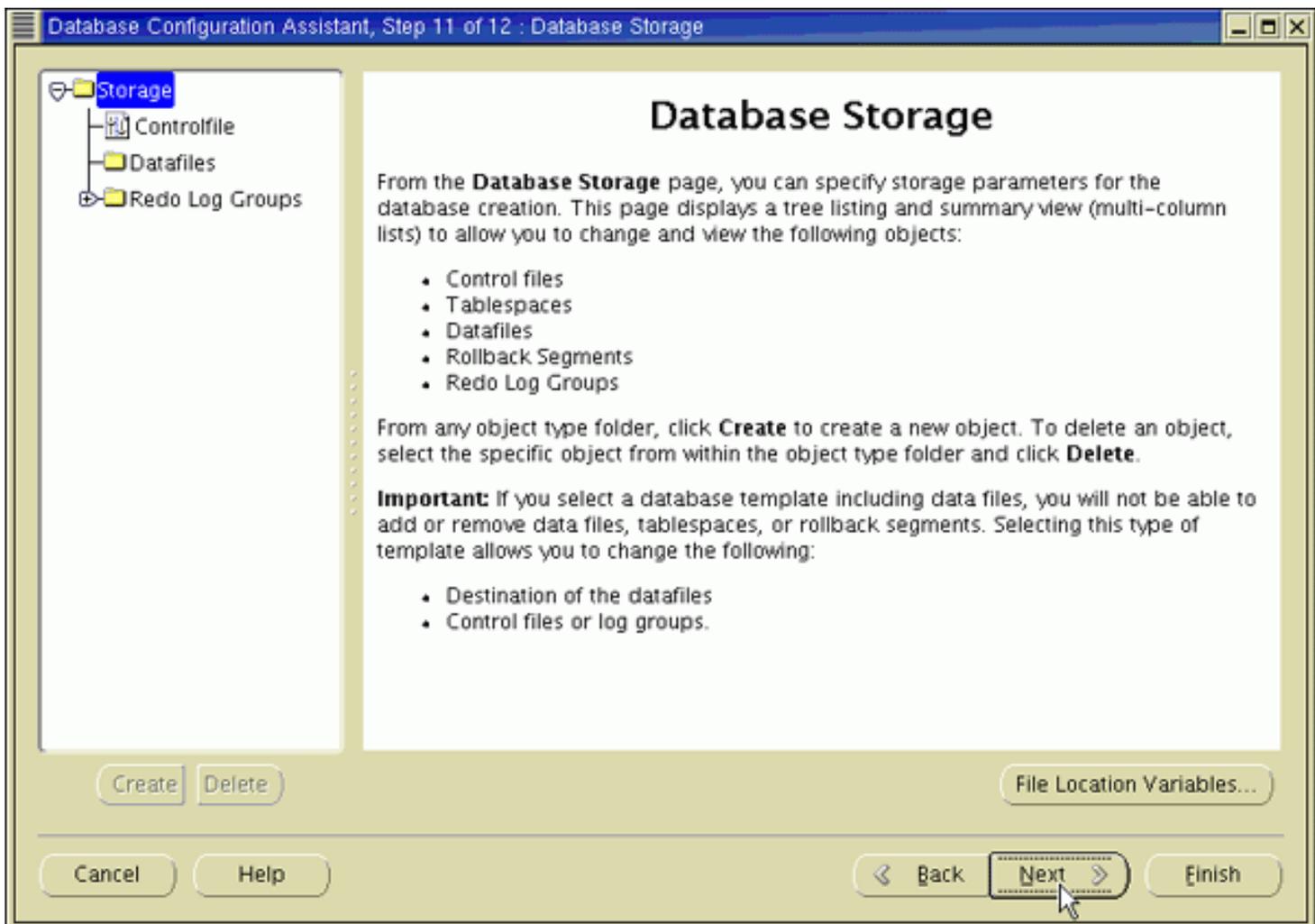
Accept the defaults for Initialization Parameters and click **Next** .

12.



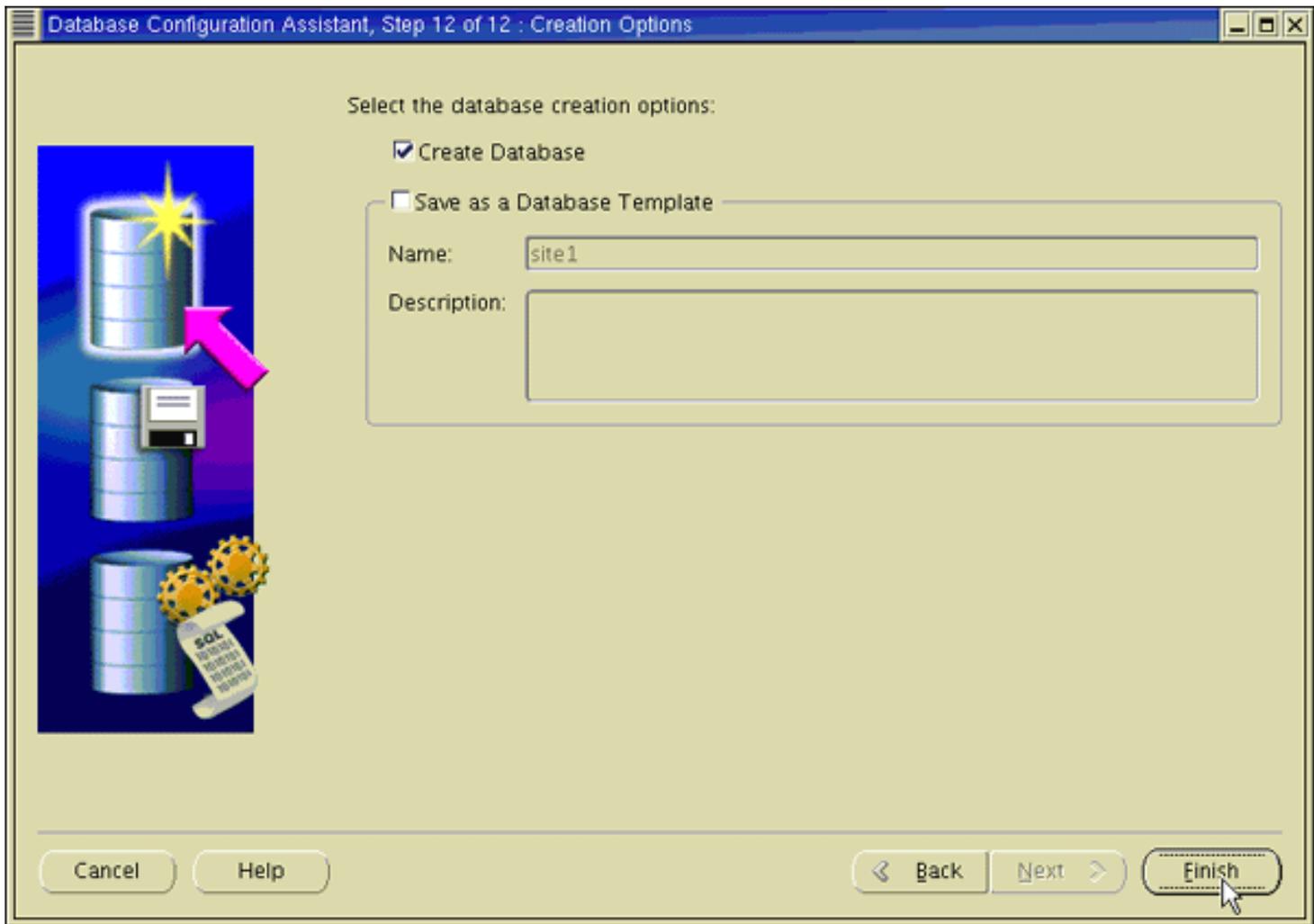
Accept the defaults for Database Storage and click **Next** .

13.



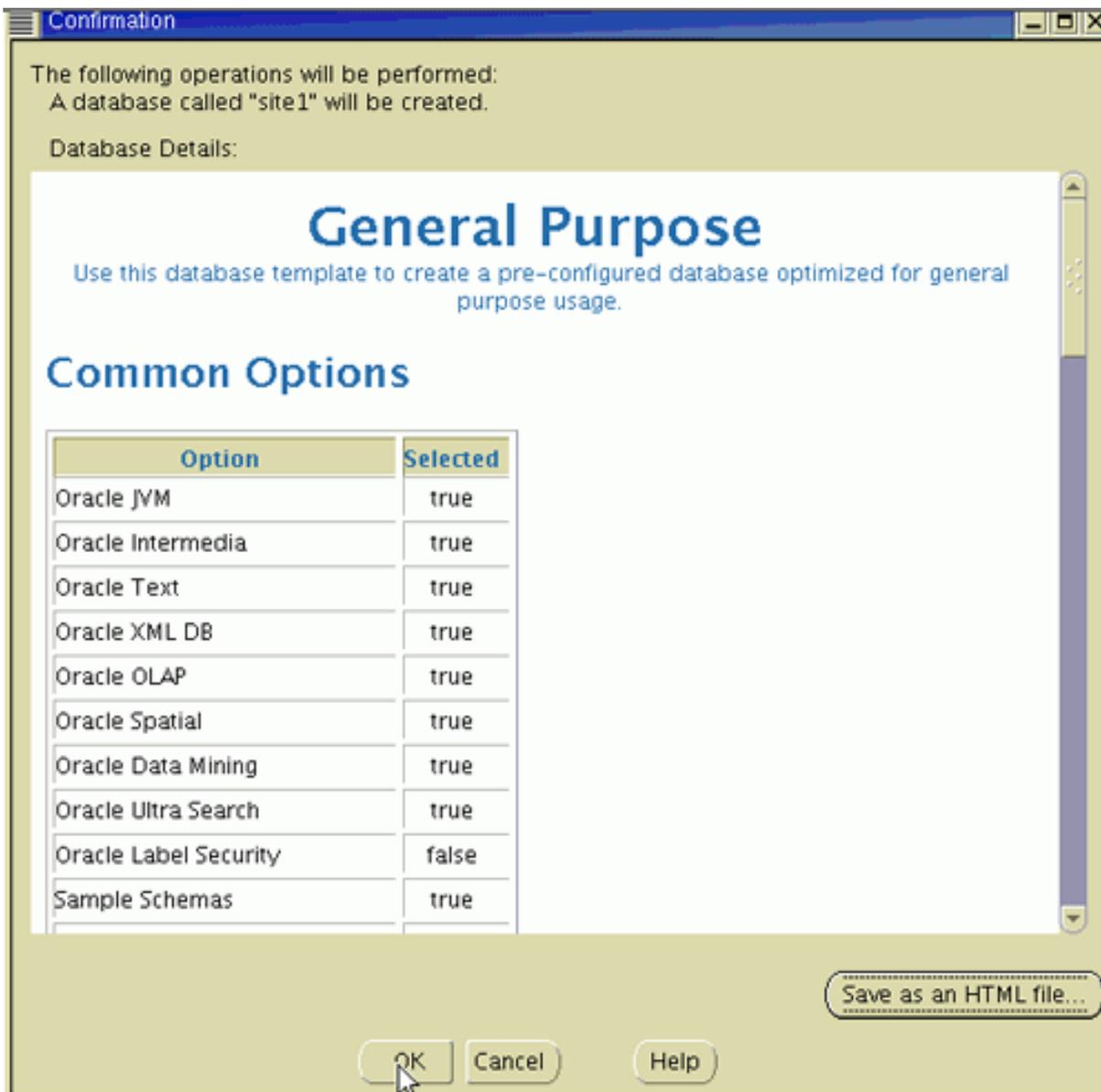
Make sure **Create Database** is checked and click **Next** .

14.



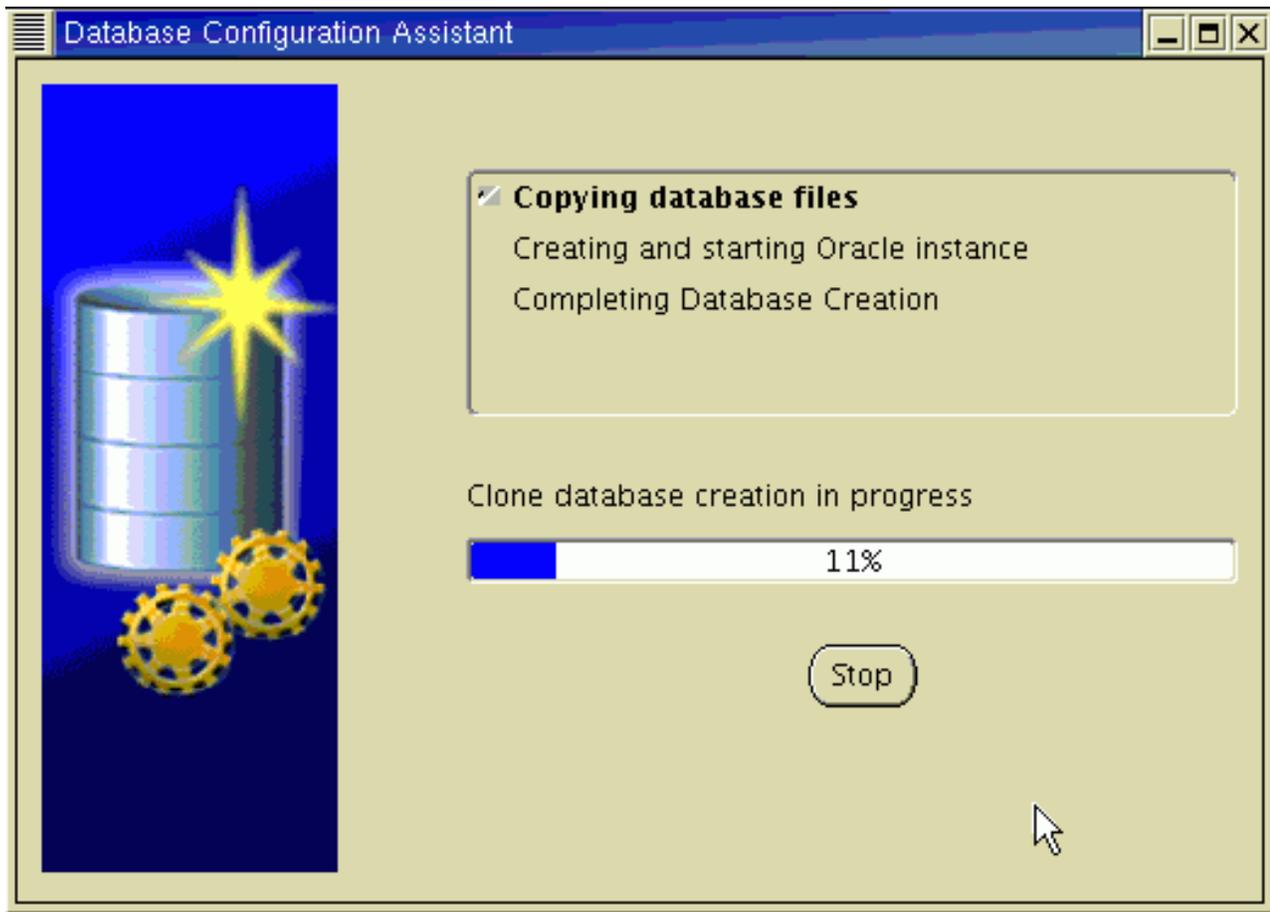
At the Confirmation screen, click **OK** to start the database creation.

15.



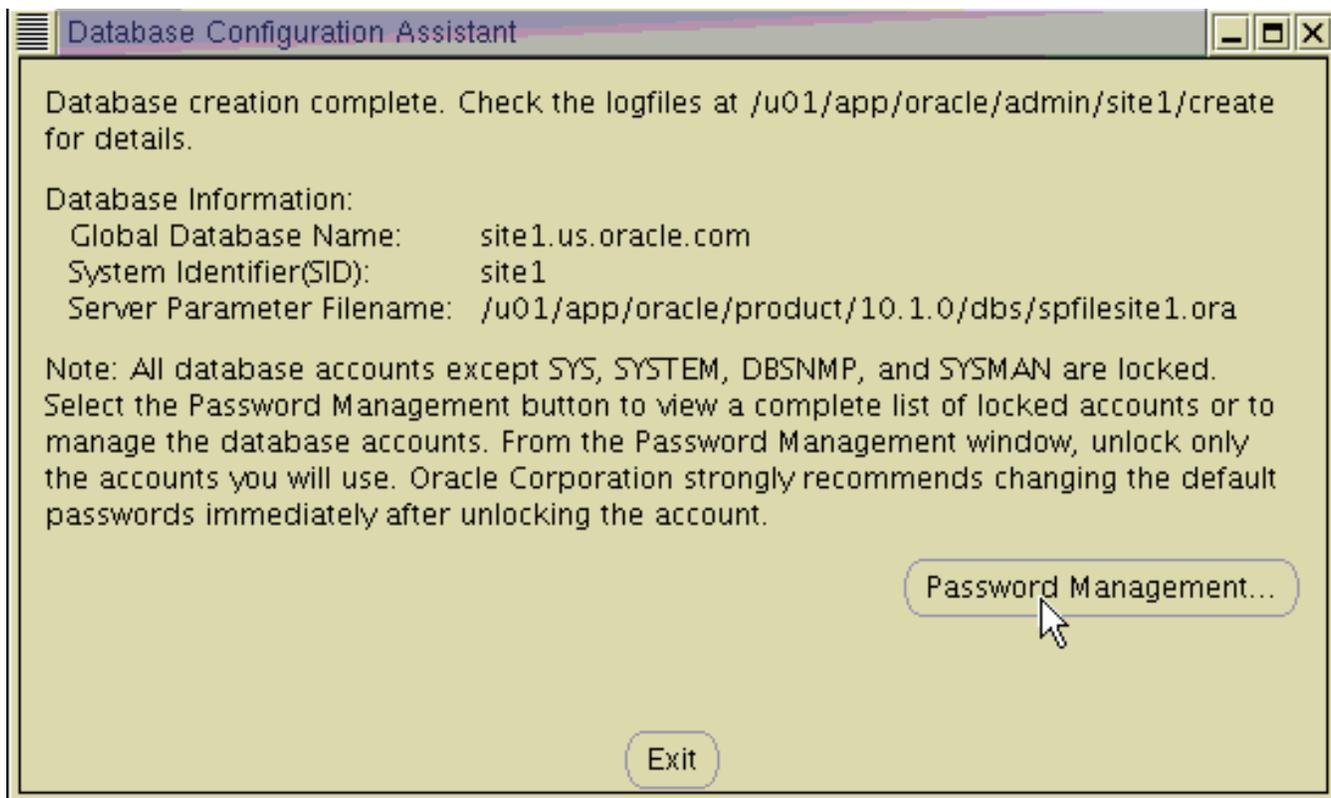
You will see the progress window appear.

16.

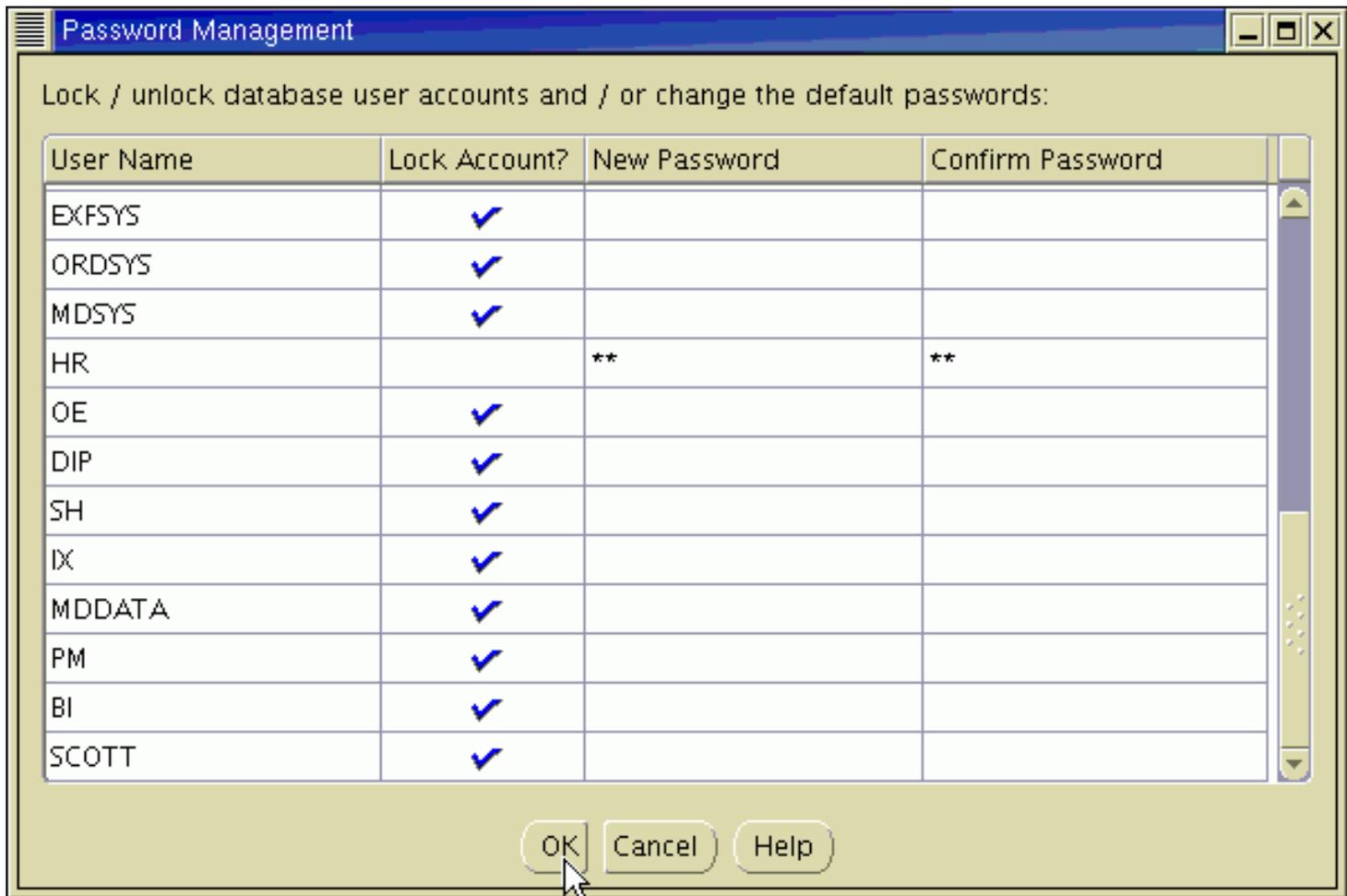


You need to unlock the HR user. Click **Password Management**.

17.

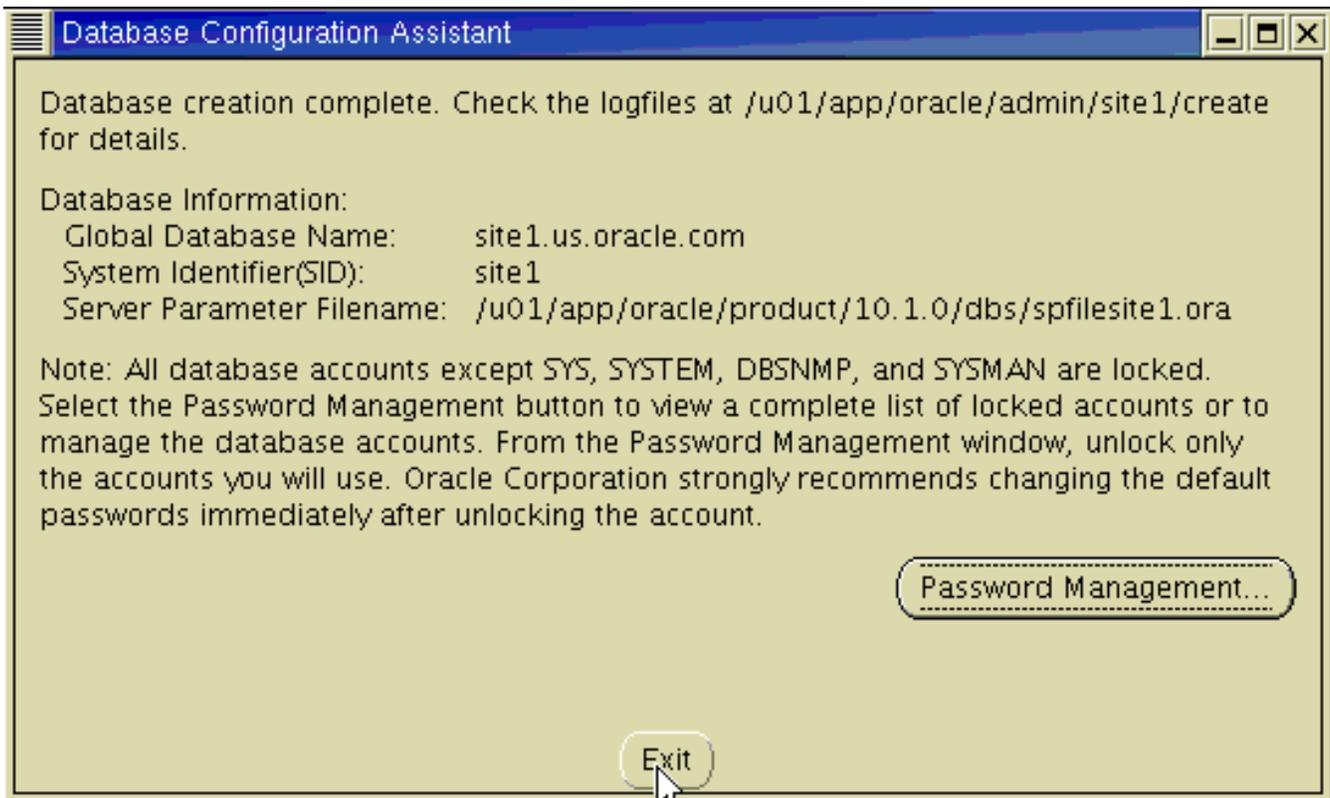


18. Scroll down to the **HR** user, click the **Unlock Account?** column and enter **HR** in both the new password and confirm password fields and click **OK** .



After the database has been created, click **Exit** .

19.



You want to perform the same steps again for the **site2.us.oracle.com** database.

20.

Creating the Tablespace to be Transported on the Source Database

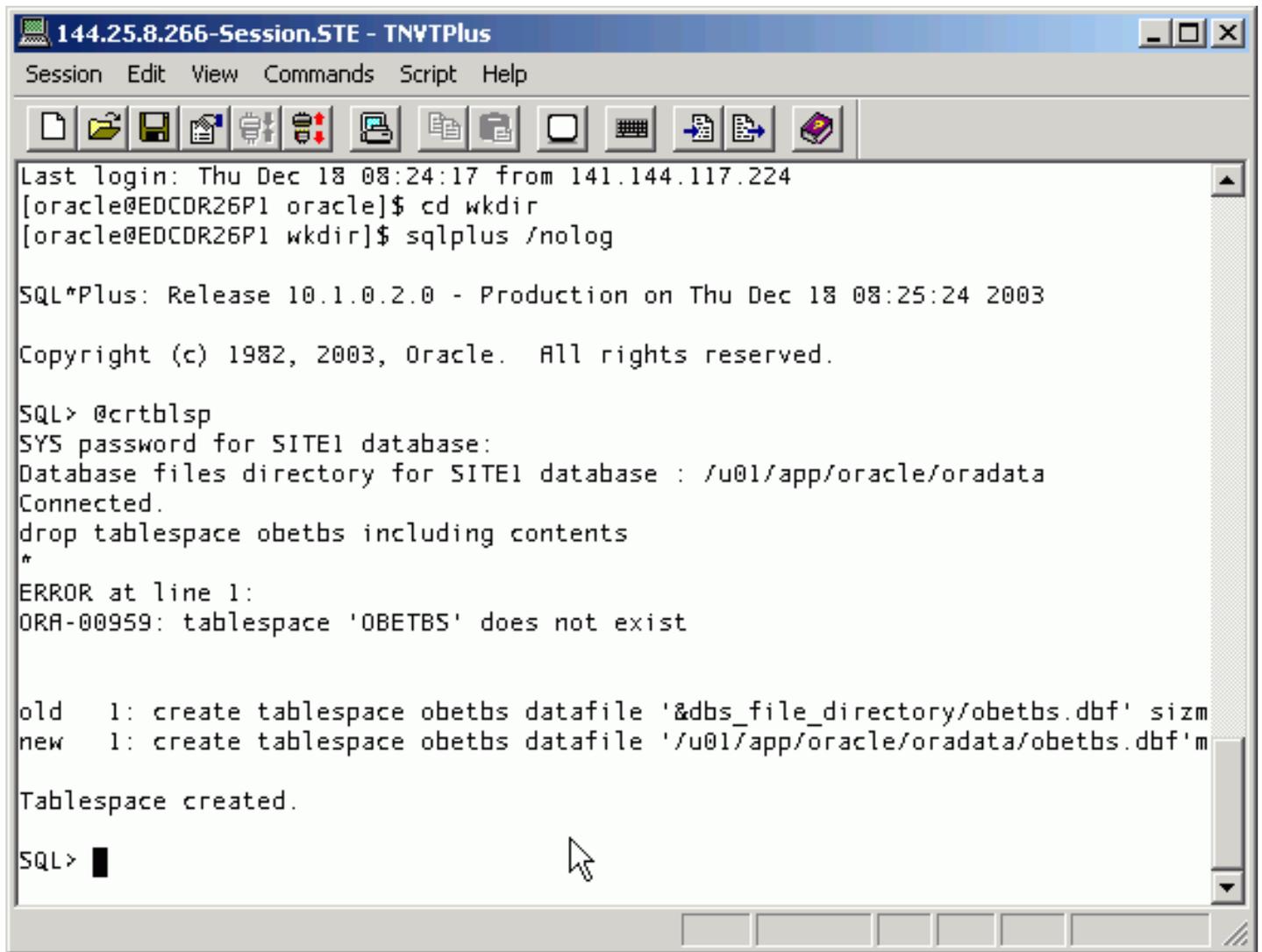
To show the new transportable tablespace capability using Streams, you need to create a Tablespace on the SITE1 database. Perform the following steps:

1. From a terminal window, execute the following command(s):

```
cd wkdir
sqlplus /nolog
@crtblsp
```

The **crtblsp.sql** script contains the following:

```
connect sys/<password>@SITE1 as sysdba
create tablespace obetbs datafile '<oradata_directory>/obetbs.dbf' size 20m;
```

A screenshot of a TNVTPlus terminal window. The title bar reads "144.25.8.266-Session.SITE - TNVTPlus". The menu bar includes "Session", "Edit", "View", "Commands", "Script", and "Help". The toolbar contains various icons for file operations and terminal functions. The terminal text shows a user logging in from 141.144.117.224, navigating to a directory, and starting SQL*Plus. The user attempts to drop a tablespace named 'OBETBS' but receives an error: "ORA-00959: tablespace 'OBETBS' does not exist". The user then creates the tablespace with the command: "create tablespace obetbs datafile '&dbms_file_directory/obetbs.dbf' size new 1: create tablespace obetbs datafile '/u01/app/oracle/oradata/obetbs.dbf'm". The output confirms "Tablespace created." and the prompt "SQL>" is visible with a cursor.

```
144.25.8.266-Session.SITE - TNVTPlus
Session Edit View Commands Script Help

Last login: Thu Dec 18 08:24:17 from 141.144.117.224
[oracle@EEDCDR26P1 oracle]$ cd wkdir
[oracle@EEDCDR26P1 wkdir]$ sqlplus /nolog

SQL*Plus: Release 10.1.0.2.0 - Production on Thu Dec 18 08:25:24 2003

Copyright (c) 1982, 2003, Oracle. All rights reserved.

SQL> @crtblsp
SYS password for SITE1 database:
Database files directory for SITE1 database : /u01/app/oracle/oradata
Connected.
drop tablespace obetbs including contents
*
ERROR at line 1:
ORA-00959: tablespace 'OBETBS' does not exist

old 1: create tablespace obetbs datafile '&dbms_file_directory/obetbs.dbf' size
new 1: create tablespace obetbs datafile '/u01/app/oracle/oradata/obetbs.dbf'm

Tablespace created.

SQL> █
```

Creating the Users at the Source and Destination Databases

You need to create a Streams Administrator and General User on both the SITE1 and SITE2 databases. Perform the following steps:

1. From your SQL*Plus session, execute the following SQL command(s):

```
@crusers
```

The `crusers.sql` script contains the following:

```
REM
REM Create STRMADMIN and OBE users at SITE1.US.ORACLE.COM
REM

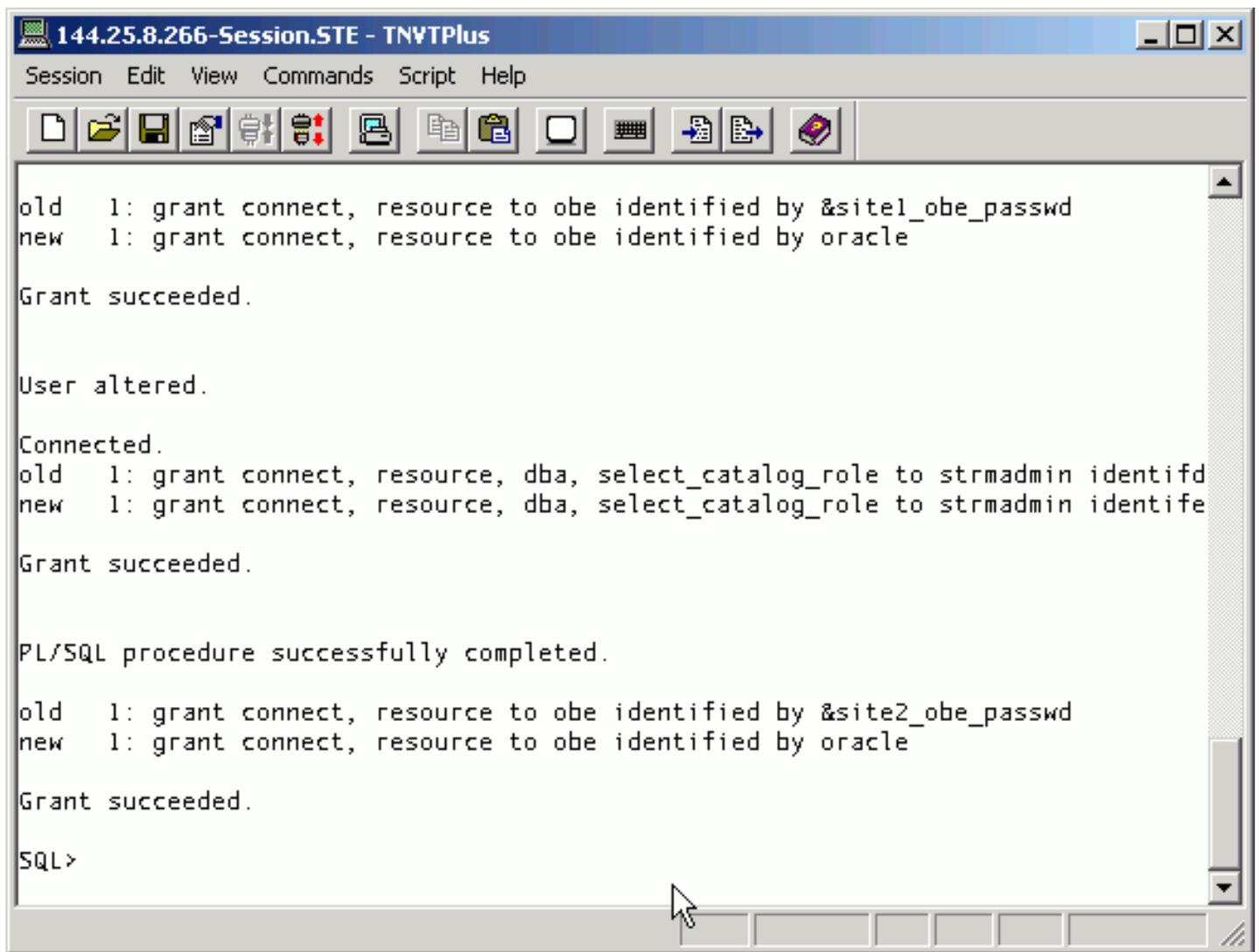
grant connect, resource, dba, select_catalog_role to strmadmin
  identified by <password>;
exec dbms_streams_auth.grant_admin_privilege('STRMADMIN',TRUE);

grant connect, resource to obe identified by <password>;
alter user OBE default tablespace OBETBS;

REM
REM Create STRMADMIN and OBE users at SITE2.US.ORACLE.COM
REM

connect sys/<password>@SITE2 as sysdba
grant connect, resource, dba, select_catalog_role to strmadmin
  identified by <password>;
exec dbms_streams_auth.grant_admin_privilege('STRMADMIN',TRUE);

grant connect, resource to obe identified by <password>;
```



The screenshot shows a terminal window titled "144.25.8.266-Session.STE - TNVTPlus". The window has a menu bar with "Session", "Edit", "View", "Commands", "Script", and "Help". Below the menu bar is a toolbar with various icons for file operations and execution. The main text area contains the following SQL commands and their output:

```
old 1: grant connect, resource to obe identified by &site1_obe_passwd
new 1: grant connect, resource to obe identified by oracle

Grant succeeded.

User altered.

Connected.
old 1: grant connect, resource, dba, select_catalog_role to strmadmin identifd
new 1: grant connect, resource, dba, select_catalog_role to strmadmin identife

Grant succeeded.

PL/SQL procedure successfully completed.

old 1: grant connect, resource to obe identified by &site2_obe_passwd
new 1: grant connect, resource to obe identified by oracle

Grant succeeded.

SQL>
```

Setting Up Streams on the Destination Database

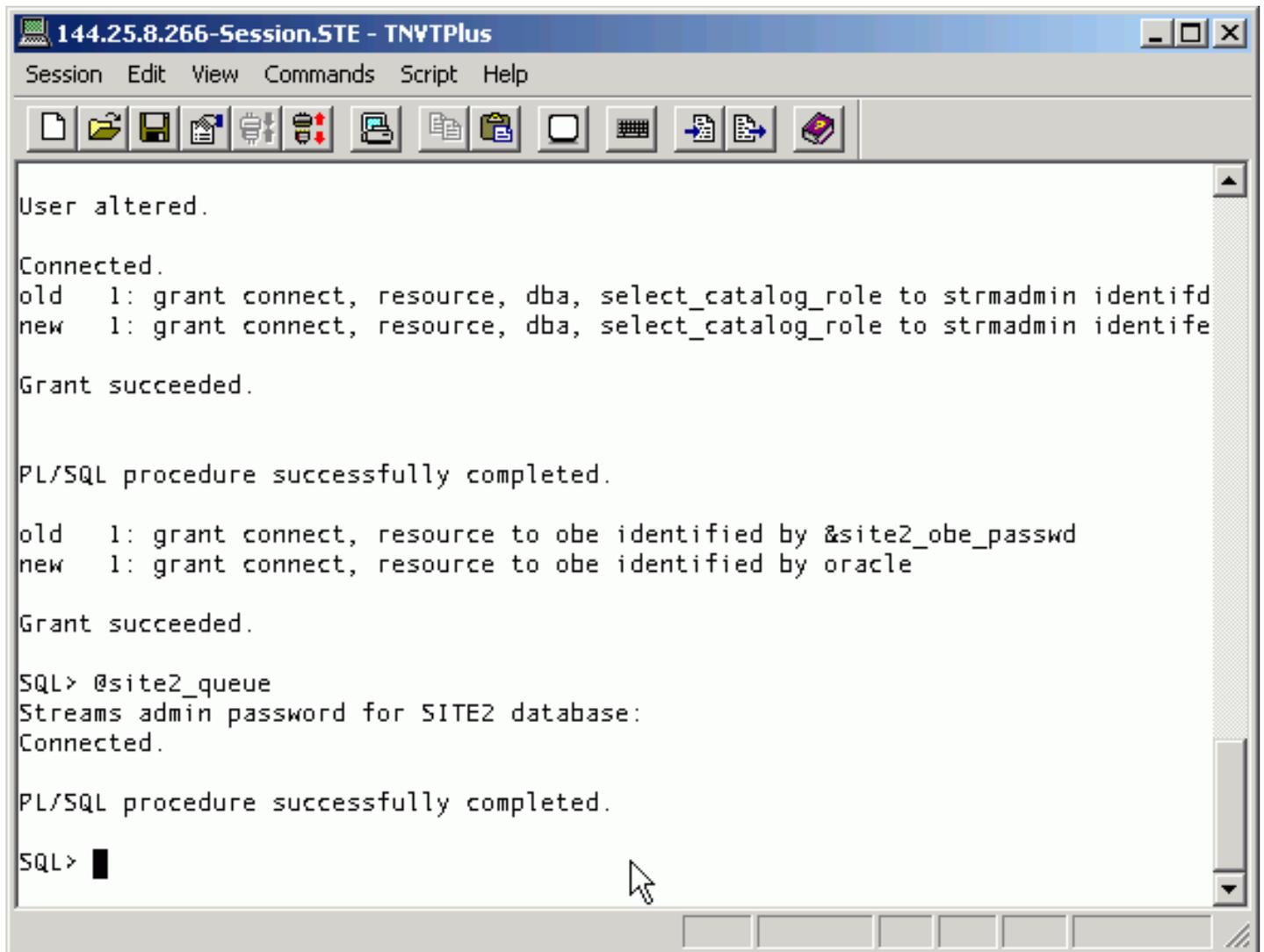
You need to now setup the Streams Queues, database links and directories on the SITE2 database. Perform the following steps:

1. You need to first setup your queue. From your SQL*Plus session, execute the following SQL commands:

```
@site2_queue
```

The `site2_queue.sql` script contains the following:

```
connect STRMADMIN/<password>@SITE2
exec dbms_streams_admin.set_up_queue();
```



The screenshot shows a SQL*Plus session window titled "144.25.8.266-Session.STE - TNVTPplus". The window contains the following text:

```
Session Edit View Commands Script Help
User altered.
Connected.
old 1: grant connect, resource, dba, select_catalog_role to strmadmin identifd
new 1: grant connect, resource, dba, select_catalog_role to strmadmin identife
Grant succeeded.
PL/SQL procedure successfully completed.
old 1: grant connect, resource to obe identified by &site2_obe_passwd
new 1: grant connect, resource to obe identified by oracle
Grant succeeded.
SQL> @site2_queue
Streams admin password for SITE2 database:
Connected.
PL/SQL procedure successfully completed.
SQL> █
```

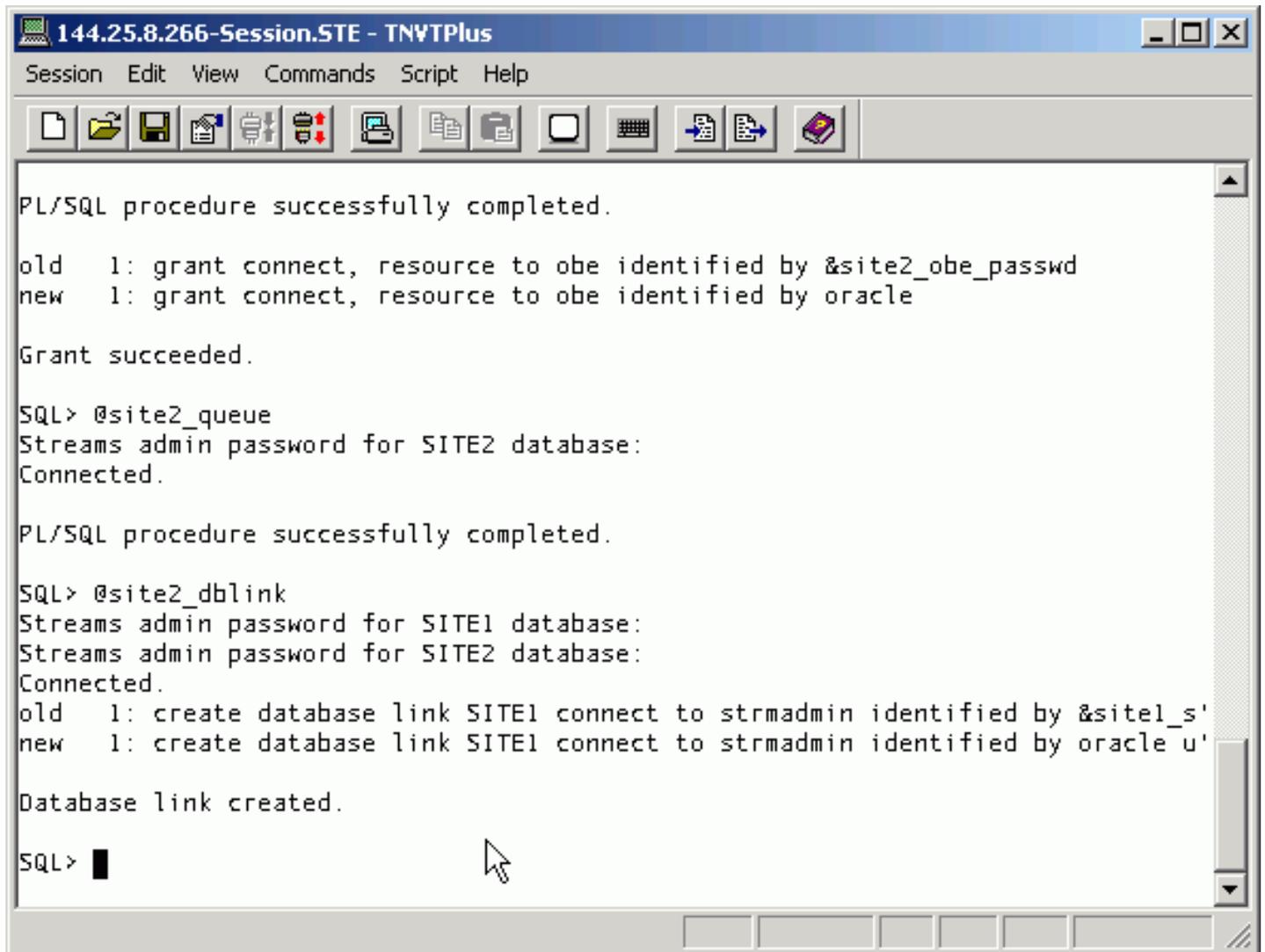
- Now you need to create your database link to the site1 database. From your SQL*Plus session, execute the following SQL commands:

```
@site2_dblink
```

The `site2_dblink.sql` script contains the following:

```
connect STRMADMIN/<password>@SITE2
```

```
create database link site1 connect to strmadmin identified by <password> using 'site1';
```



The screenshot shows a SQL*Plus session window titled "144.25.8.266-Session.STE - TNVTPlus". The window contains the following text:

```
PL/SQL procedure successfully completed.

old   1: grant connect, resource to obe identified by &site2_obe_passwd
new   1: grant connect, resource to obe identified by oracle

Grant succeeded.

SQL> @site2_queue
Streams admin password for SITE2 database:
Connected.

PL/SQL procedure successfully completed.

SQL> @site2_dblink
Streams admin password for SITE1 database:
Streams admin password for SITE2 database:
Connected.
old   1: create database link SITE1 connect to strmadmin identified by &site1_s'
new   1: create database link SITE1 connect to strmadmin identified by oracle u'

Database link created.

SQL> █
```

- The last step to setting up streams on the site2 database is to create the directories where the files will go. From your SQL*Plus session, execute the following SQL commands:

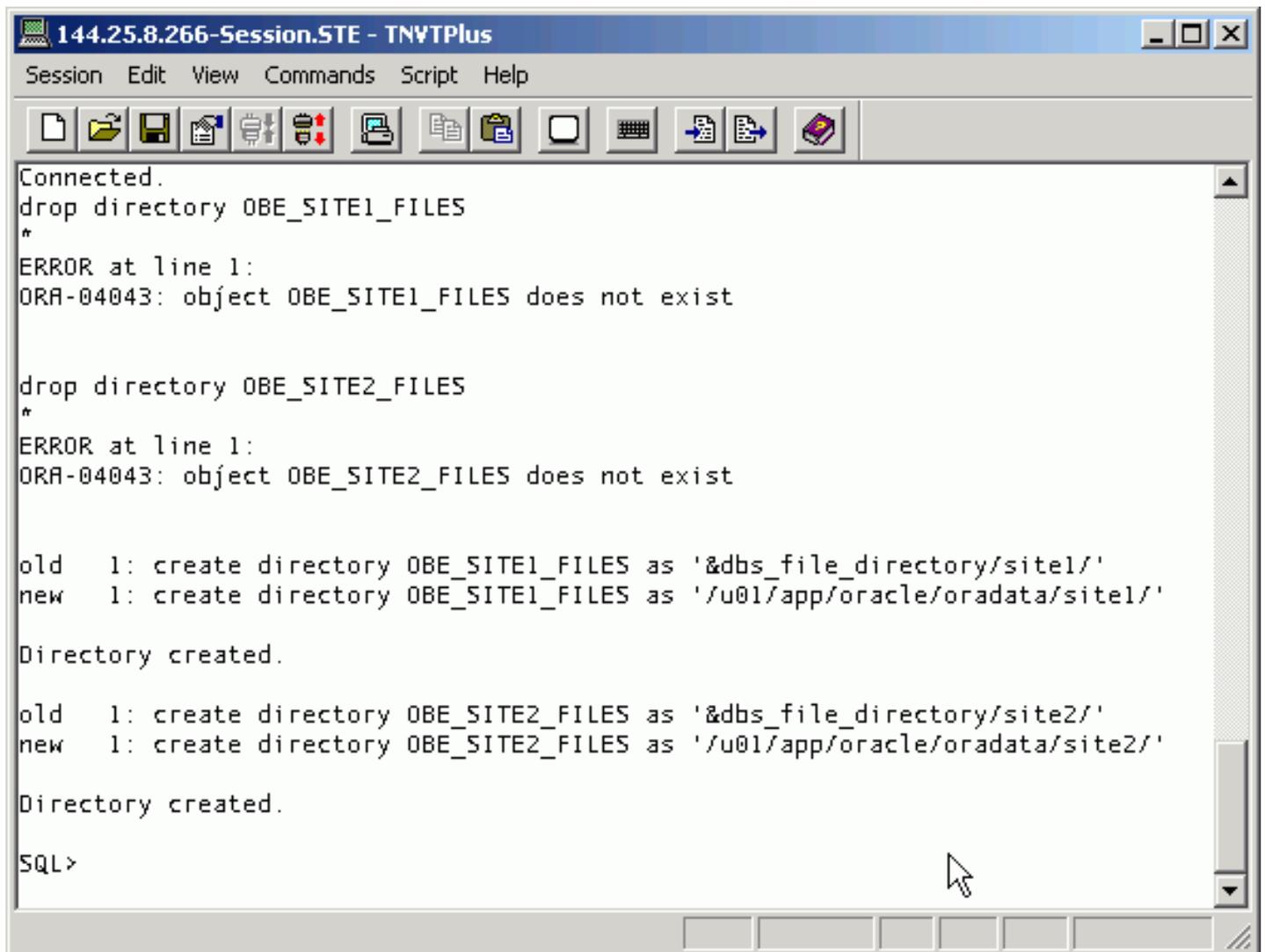
@site2_directory

The `site2_directory.sql` script contains the following:

```
connect STRMADMIN/<password>@SITE2
```

```
drop directory OBE_SITE1_FILES;  
drop directory OBE_SITE2_FILES;
```

```
create directory OBE_SITE1_FILES as '<oradata_directory>/site1/';  
create directory OBE_SITE2_FILES as '<oradata_directory>/site2/';
```



The screenshot shows a SQL*Plus session window titled "144.25.8.266-Session.STE - TNVTPPlus". The window contains the following text:

```
Connected.  
drop directory OBE_SITE1_FILES  
*  
ERROR at line 1:  
ORA-04043: object OBE_SITE1_FILES does not exist  
  
drop directory OBE_SITE2_FILES  
*  
ERROR at line 1:  
ORA-04043: object OBE_SITE2_FILES does not exist  
  
old 1: create directory OBE_SITE1_FILES as '&dbms_file_directory/site1/'  
new 1: create directory OBE_SITE1_FILES as '/u01/app/oracle/oradata/site1/'  
Directory created.  
  
old 1: create directory OBE_SITE2_FILES as '&dbms_file_directory/site2/'  
new 1: create directory OBE_SITE2_FILES as '/u01/app/oracle/oradata/site2/'  
Directory created.  
  
SQL>
```

Setting Up Streams on the Source Database

You need to now setup the Streams Queues, database links and directories on the SITE1 database. Perform the following steps:

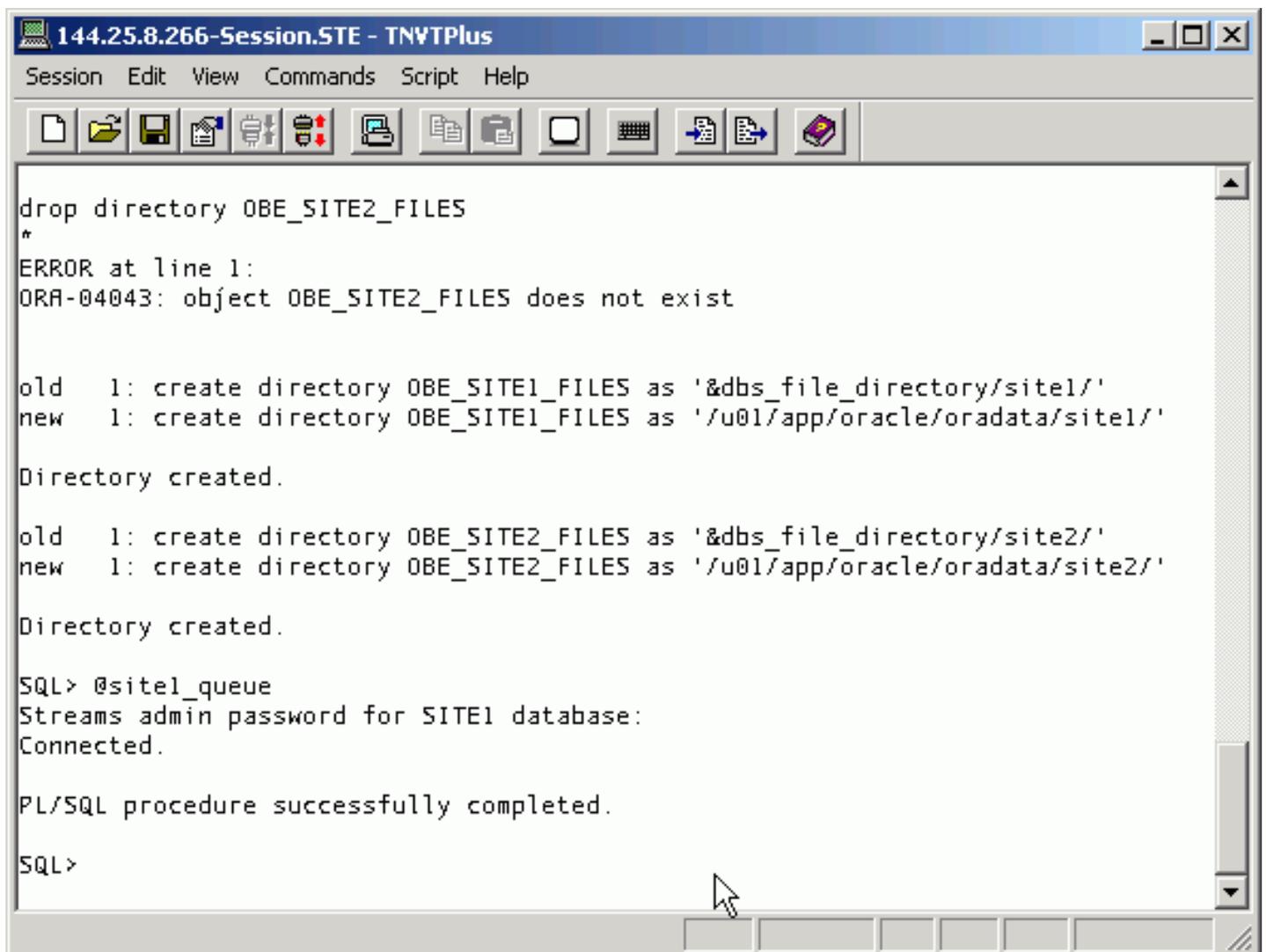
1. You need to first setup your queue. From your SQL*Plus session, execute the following SQL commands:

```
@site1_queue
```

The `site1_queue.sql` script contains the following:

```
connect STRMADMIN/<password>@SITE1

exec dbms_streams_admin.set_up_queue();
```



```
144.25.8.266-Session.SITE - TNVTPPlus
Session Edit View Commands Script Help

drop directory OBE_SITE2_FILES
*
ERROR at line 1:
ORA-04043: object OBE_SITE2_FILES does not exist

old 1: create directory OBE_SITE1_FILES as '&dbms_file_directory/site1/'
new 1: create directory OBE_SITE1_FILES as '/u01/app/oracle/oradata/site1/'

Directory created.

old 1: create directory OBE_SITE2_FILES as '&dbms_file_directory/site2/'
new 1: create directory OBE_SITE2_FILES as '/u01/app/oracle/oradata/site2/'

Directory created.

SQL> @site1_queue
Streams admin password for SITE1 database:
Connected.

PL/SQL procedure successfully completed.

SQL>
```

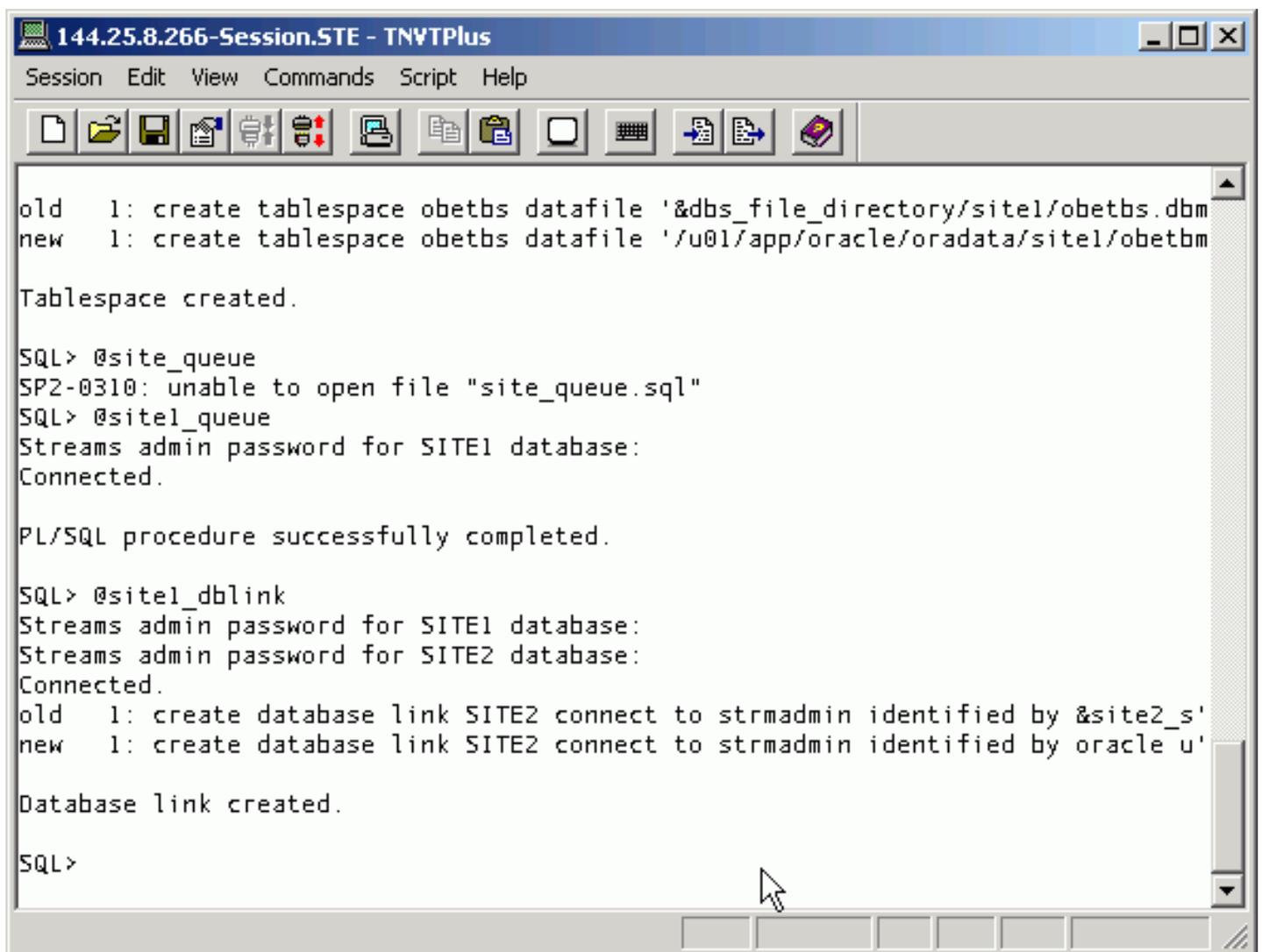
2. Now you need to create your database link to the SITE2 database. From your SQL*Plus session, execute the following SQL commands:

```
@site1_dblink
```

The `site1_dblink.sql` script contains the following:

```
connect STRMADMIN/<password>@SITE1

create database link site2 connect to strmadmin
  identified by <password> using 'site2';
```



The screenshot shows a SQL*Plus session window titled "144.25.8.266-Session.STE - TNVTPlus". The window contains the following text:

```
old 1: create tablespace obetbs datafile '&db_file_directory/site1/obetbs.dbm'
new 1: create tablespace obetbs datafile '/u01/app/oracle/oradata/site1/obetbs.dbf'

Tablespace created.

SQL> @site_queue
SP2-0310: unable to open file "site_queue.sql"
SQL> @site1_queue
Streams admin password for SITE1 database:
Connected.

PL/SQL procedure successfully completed.

SQL> @site1_dblink
Streams admin password for SITE1 database:
Streams admin password for SITE2 database:
Connected.
old 1: create database link SITE2 connect to strmadmin identified by &site2_s'
new 1: create database link SITE2 connect to strmadmin identified by oracle u'

Database link created.

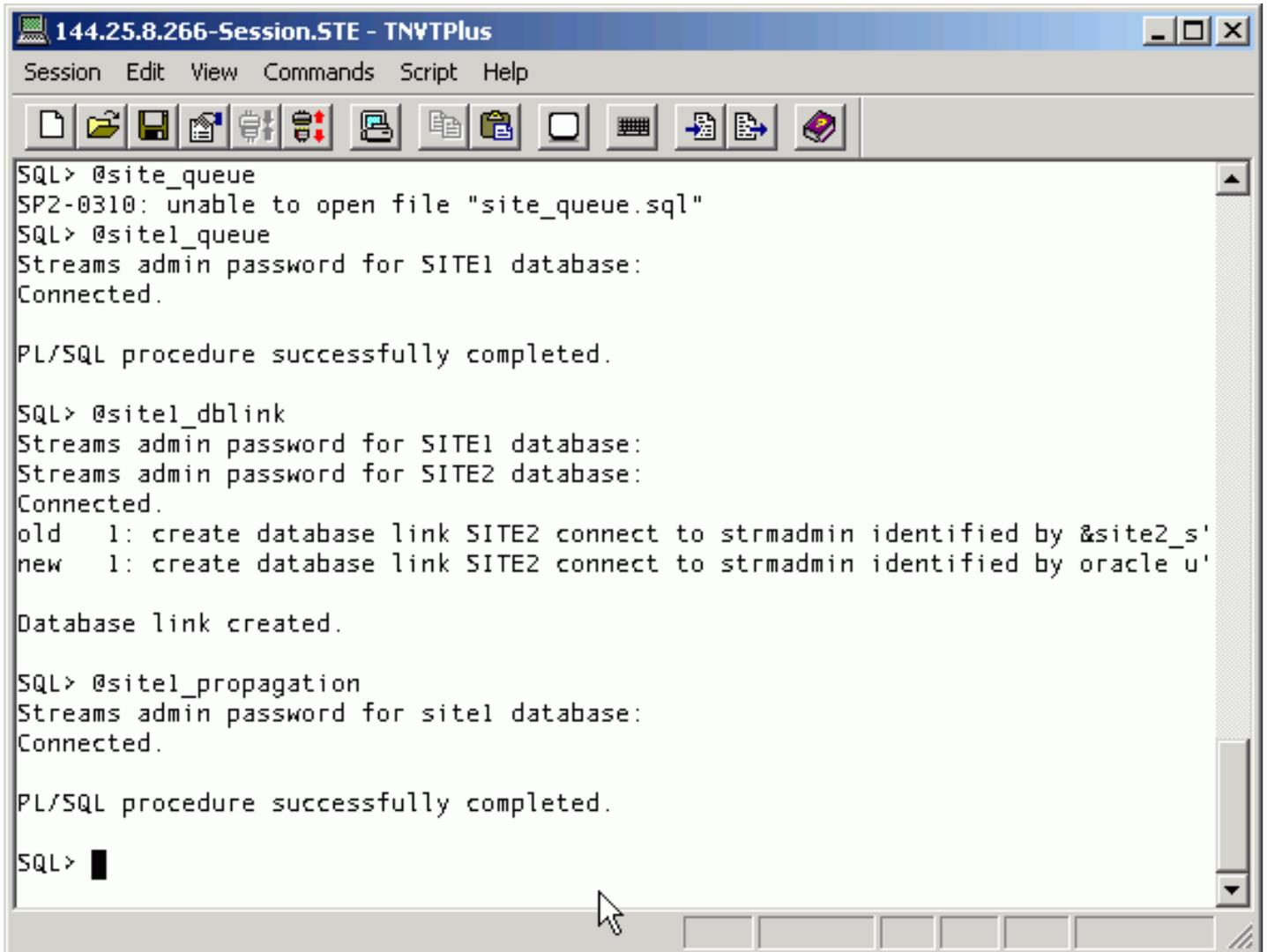
SQL>
```

2. You also need to create the propagation on your SITE2 database. From your SQL*Plus session, execute the following SQL commands:

```
@site1_propagation
```

The `site1_propagation.sql` script contains the following:

```
connect STRMADMIN/<password>@SITE1
exec dbms_propagation_admin.create_propagation('STRMADMIN_SITE1_STRM',
  'STRMADMIN.STREAMS_QUEUE', 'STRMADMIN.STREAMS_QUEUE', 'SITE2', null);
```



```
144.25.8.266-Session.SITE - TNVTPPlus
Session Edit View Commands Script Help

SQL> @site_queue
SP2-0310: unable to open file "site_queue.sql"
SQL> @site1_queue
Streams admin password for SITE1 database:
Connected.

PL/SQL procedure successfully completed.

SQL> @site1_dblink
Streams admin password for SITE1 database:
Streams admin password for SITE2 database:
Connected.
old 1: create database link SITE2 connect to strmadmin identified by &site2_s'
new 1: create database link SITE2 connect to strmadmin identified by oracle u'

Database link created.

SQL> @site1_propagation
Streams admin password for site1 database:
Connected.

PL/SQL procedure successfully completed.

SQL> █
```

- Next you need to create the directories on the SITE1 database where the files will go. From your SQL*Plus session, execute the following SQL commands:

```
@site1_directory
```

The `site1_directory.sql` script contains the following:

```
connect STRMADMIN/<password>@SITE1

drop directory OBE_SITE1_FILES;

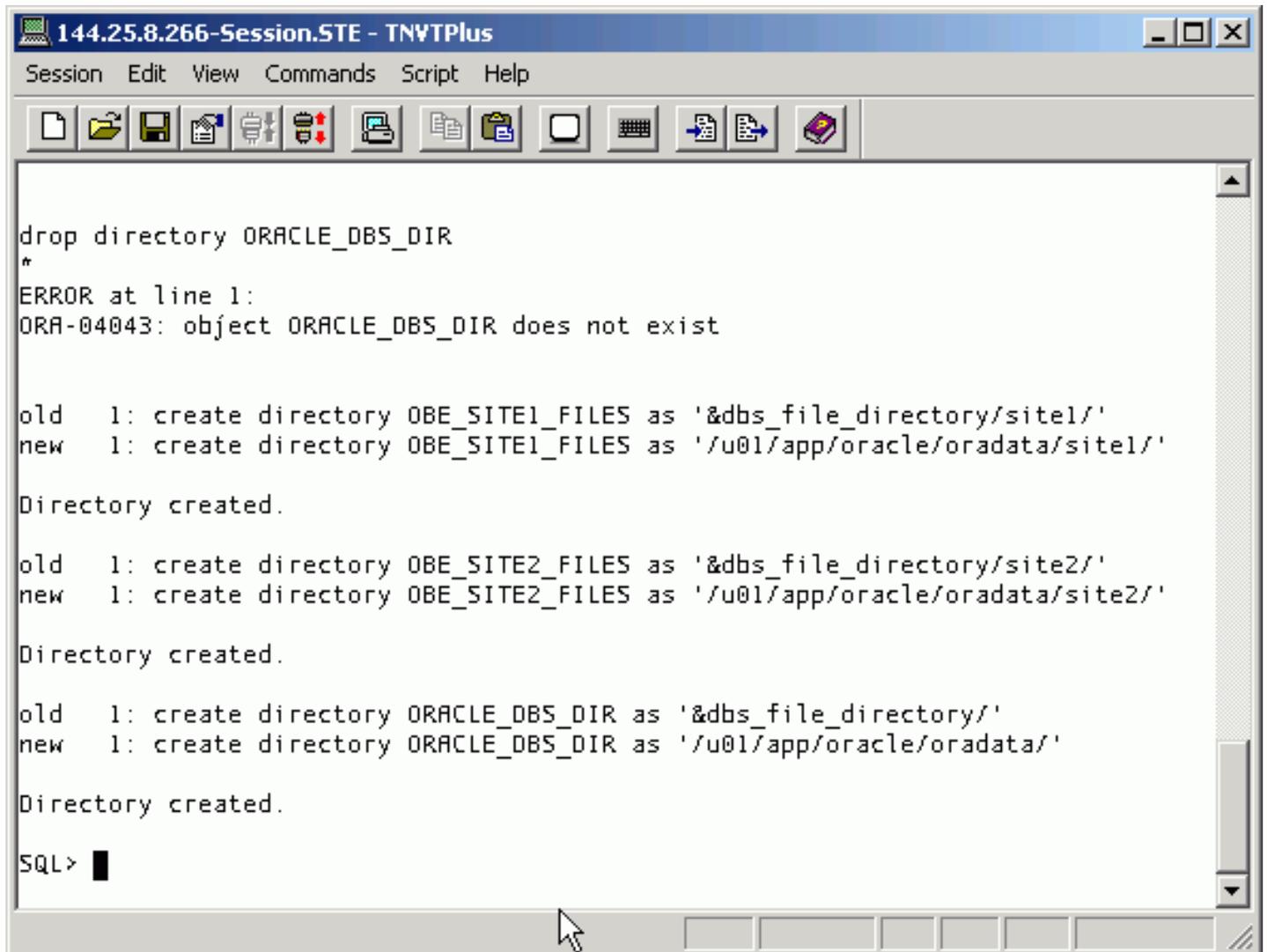
drop directory OBE_SITE2_FILES;

drop directory ORACLE_DBS_DIR ;

create directory OBE_SITE1_FILES as '<oradata_directory>/site1/';

create directory OBE_SITE2_FILES as '<oradata_directory>/site2/';

create directory ORACLE_DBS_DIR as '<oradata_directory>/';
```



```
144.25.8.266-Session.STE - TNYTPlus
Session Edit View Commands Script Help

drop directory ORACLE_DBS_DIR
*
ERROR at line 1:
ORA-04043: object ORACLE_DBS_DIR does not exist

old 1: create directory OBE_SITE1_FILES as '&dbms_file_directory/site1/'
new 1: create directory OBE_SITE1_FILES as '/u01/app/oracle/oradata/site1/'

Directory created.

old 1: create directory OBE_SITE2_FILES as '&dbms_file_directory/site2/'
new 1: create directory OBE_SITE2_FILES as '/u01/app/oracle/oradata/site2/'

Directory created.

old 1: create directory ORACLE_DBS_DIR as '&dbms_file_directory/'
new 1: create directory ORACLE_DBS_DIR as '/u01/app/oracle/oradata/'

Directory created.

SQL>
```

- The last step is to create a table in the tablespace and put some data in it for streams to transport. From your SQL*Plus session, execute the following SQL commands:

@loaddata

The `loaddata.sql` script contains the following:

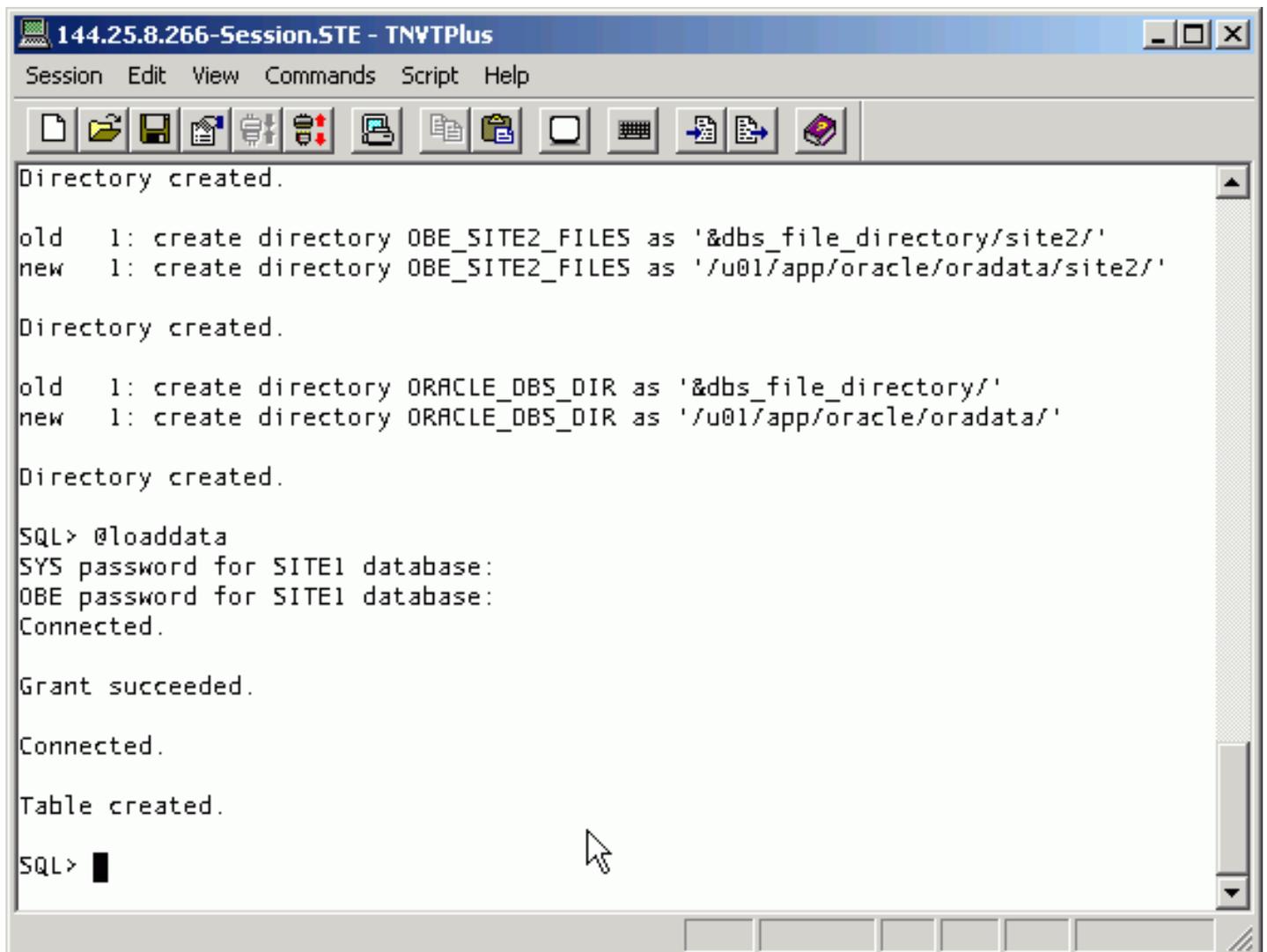
```
connect sys/<password>@SITE1 as sysdba

grant all on hr.employees to obe;

connect obe/<password>@SITE1

create table employees as

select * from hr.employees;
```

A screenshot of a TNVTPlus SQL*Plus session window. The title bar reads "144.25.8.266-Session.SITE - TNVTPlus". The menu bar includes "Session", "Edit", "View", "Commands", "Script", and "Help". The toolbar contains icons for file operations and database actions. The main text area shows the following output:

```
Directory created.

old 1: create directory OBE_SITE2_FILES as '&dbms_file_directory/site2/'
new 1: create directory OBE_SITE2_FILES as '/u01/app/oracle/oradata/site2/'

Directory created.

old 1: create directory ORACLE_DBS_DIR as '&dbms_file_directory/'
new 1: create directory ORACLE_DBS_DIR as '/u01/app/oracle/oradata/'

Directory created.

SQL> @loaddata
SYS password for SITE1 database:
OBE password for SITE1 database:
Connected.

Grant succeeded.

Connected.

Table created.

SQL> █
```

Maintaining Tablespaces

Now you are ready to run streams that will transport the tablespace from the SITE1 database to the SITE2 database. Perform the following steps:

1. First, you want to view the contents of the database to see that there is data in the SITE1 database that is not in the SITE2 database. From your SQL*Plus session, execute the following script:

```
@chkdb
```

The `chkdb.sql` script contains the following:

```
connect obe/<password>@SITE1
```

```
select * from employees;
```

```
connect obe/<password>@SITE2
```

```
select * from employees;
```

```

101          110
      206 William          Gietz
WGIETZ          515.123.8181          07-JUN-94 AC_ACCOUNT          8300
      205          110

EMPLOYEE_ID FIRST_NAME          LAST_NAME
-----
EMAIL          PHONE_NUMBER          HIRE_DATE JOB_ID          SALARY
-----
COMMISSION_PCT MANAGER_ID DEPARTMENT_ID
-----

107 rows selected.

Connected.
select * from employees
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>

```

Note that there is no employees table in the SITE2 database.

- Now you want to run the streams process to transport the tablespace. From your SQL*Plus session, execute the following script:

```

connect strmadmin/<password>@SITE1
@maintain_ts

```

The `maintain_ts.sql` script contains the following:

```

BEGIN

  DBMS_STREAMS_ADM.MAINTAIN_SIMPLE_TABLESPACE (

```

```

tablespace_name=>'OBETBS' ,

source_directory_object=>'OBE_SITE1_FILES' ,

destination_directory_object=>'OBE_SITE2_FILES' ,

destination_database=>'SITE2.US.ORACLE.COM' ,

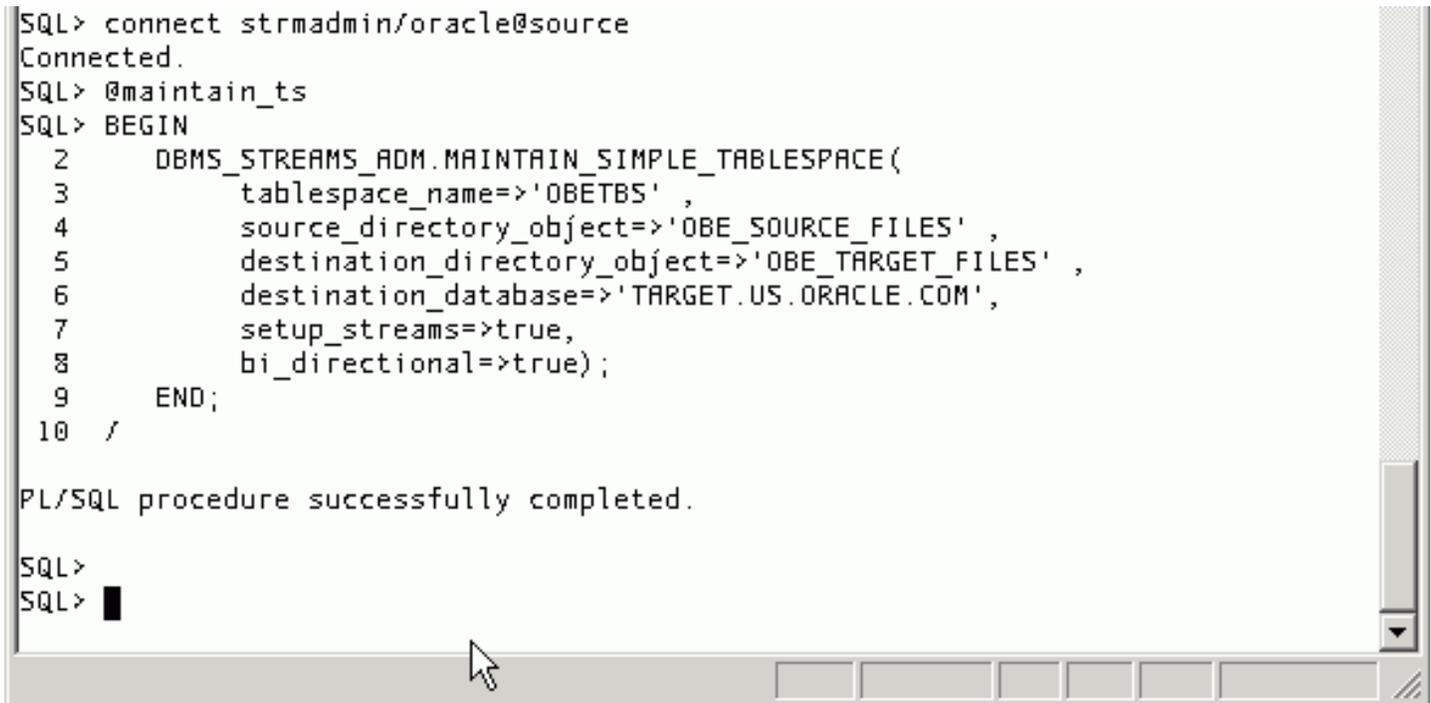
setup_streams=>>true,

bi_directional=>>true);

END;

/

```



```

SQL> connect strmadmin/oracle@source
Connected.
SQL> @maintain_ts
SQL> BEGIN
  2     DBMS_STREAMS_ADM.MAINTAIN_SIMPLE_TABLESPACE(
  3         tablespace_name=>'OBETBS' ,
  4         source_directory_object=>'OBE_SOURCE_FILES' ,
  5         destination_directory_object=>'OBE_TARGET_FILES' ,
  6         destination_database=>'TARGET.US.ORACLE.COM',
  7         setup_streams=>>true,
  8         bi_directional=>>true);
  9     END;
10 /

PL/SQL procedure successfully completed.

SQL>
SQL> █

```

3. Once the script finishes you can rerun the script to check the database. From your SQL*Plus session, execute the following script:

```
@chkdb
```

The `chkdb.sql` script contains the following:

```

connect obe/<password>@SITE1

select * from employees;

connect obe/<password>@SITE2

```

```
select * from employees;
```

```

144.25.8.266-Session.STE - TNVTPlus
Session Edit View Commands Script Help
[Icons]
EMAIL                PHONE_NUMBER        HIRE_DATE  JOB_ID            SALARY
-----
COMMISSION_PCT  MANAGER_ID  DEPARTMENT_ID
-----
      205 Shelley                Higgins
SHIGGINS                515.123.8080        07-JUN-94  AC_MGR            12000
                        101                110
      206 William                Gietz
WGIETZ                515.123.8181        07-JUN-94  AC_ACCOUNT        8300
                        205                110

EMPLOYEE_ID  FIRST_NAME        LAST_NAME
-----
EMAIL                PHONE_NUMBER        HIRE_DATE  JOB_ID            SALARY
-----
COMMISSION_PCT  MANAGER_ID  DEPARTMENT_ID
-----

107 rows selected.

SQL>
SQL>

```

Note that you now have data in the SITE2 database.

 **Move your mouse over this icon to hide all screenshots**