Oracle Database - Enterprise Edition - Version 10.2.0.5 and later  
Oracle ODBC Driver - Version 10.2.0.5 and later  
Information in this document applies to any platform.  
\*\*\*Checked for relevance on 20-Jul-2016\*\*\*

**GOAL**

How to use Oracle Wallets / Secure External Password Store (SEPS) with Oracle ODBC Driver

TIP: Oracle Wallets and SEPS are Database Security Features. The article section relating to ODBC is at the very end. If the Wallet and SEPS are already configured and working then proceed to step 5.

This article shows how to connect an Oracle User (defined in a External Password store) using the Oracle ODBC driver.  SEPS allows a user to connect using syntax similar to OS Authentication  
  
For Example:

sqlplus /@ORCL  
  
or  
  
sqlplus /

SEPS is a new feature from Oracle 10.2.0.  
  
As defined in "Secure External Password Store" (Oracle Database Security Guide)

"Password credentials for connecting to databases can now be stored in a client-side Oracle wallet, a secure software container used to store authentication and signing credentials. This wallet usage can simplify large-scale deployments that rely on password credentials for connecting to databases. When this feature is configured, application code, batch jobs, and scripts no longer need embedded user names and passwords. Risk is reduced because such passwords are no longer exposed in the clear, and password management policies are more easily enforced without changing application code whenever user names or passwords change."

**SOLUTION**

**1. First create a wallet**  
  
This client side wallet will contain the user/password information in a secure certificate. MKSTORE is a command line utility for creating/modifying Oracle Wallets. You can find this utility installed with the Oracle Database and also the Standard Oracle Client (not instant client).

**mkstore -wrl "C:\WALLET" -create**

provide a password when prompted  
  
1.1 For Oracle RDBMS 11.2 it is recommended to use the following instead

**orapki wallet create -wallet "C:\WALLET" -auto\_login\_local**

For additional information see:  
  
[Document 1114599.1](https://support.oracle.com/epmos/faces/DocumentDisplay?parent=DOCUMENT&sourceId=1430666.1&id=1114599.1) How To Prevent The Secure Password Store Wallet From Being Moved to Another Host   
  
**2. Create the password credentials to the wallet**  
  
In this example the tnsnames alias is "ORCL" and the user is "SCOTT" and password "TIGER" It is recommended the Password parameter is omitted and entered via a prompt instead.

**mkstore -wrl "C:\WALLET" -createCredential ORCL SCOTT TIGER**

**3. Optionally list the credentials to ensure it's correct.**

**mkstore -wrl "C:WALLET" -listCredential**

**4. Add the following to the SQLNET.ora**

NOTE: it is important to keep the indentation for the file to be interpreted correctly.

**WALLET\_LOCATION =  
  (SOURCE =  
  (METHOD = FILE)  
  (METHOD\_DATA =   
  (DIRECTORY = c:\wallet)  
    )  
  )  
  
SQLNET.WALLET\_OVERRIDE = TRUE  
SSL\_CLIENT\_AUTHENTICATION = FALSE  
SSL\_VERSION = 0**

Test with a SQL\*Plus connection   
  
SQLPLUS /@ORCL  
  
  
**5.  Test with ODBC Driver Connection**

In the ODBC Administrator create a DSN and leave the username filed empty. The "Test Connection" button should succeed.  
  
You can also test with a simple  VB Script. Simply copy and paste the following into a file "test.vbs" and execute it by double clicking the file in windows explorer.

set con = createobject("adodb.connection")  
con.open "dsn=my\_dsn\_name\_here"  
set rs = createobject("adodb.recordset")  
rs.open "SELECT USER FROM DUAL",con  
wscript.echo rs.fields(0).value

TIP: This does not work with the Microsoft ODBC driver (deprecated). With the Microsoft ODBC Driver you may see ORA-1017 or  ORA-1019. In short Microsoft ODBC is written using OCI7.