Implementing Oracle Transparent Data Encryption in PeopleTools 8.52

This section contains an overview and discusses how to:

- Determine fields to encrypt.
- Set up the Oracle Wallet.
- Set the encryption algorithm.
- Encrypt fields.
- Manage fields encrypted for TDE.

See Also

For more information on Oracle's Transparent Data Encryption feature refer to*Oracle® Database Advanced Security Administrator's Guide*

Understanding Transparent Data Encryption

PeopleTools enables you to implement Oracle's Transparent data encryption (TDE) feature to encrypt the columns you select, enhancing the security of your PeopleSoft applications.

Transparent data encryption (TDE) enables encryption of sensitive data in database columns as it is stored in the operating system files. It provides for secure storage and management of encryption keys in a security module located outside database, separating ordinary program functions from those that pertain to security, such as encryption.

This separation enables you to divide administration duties between DBAs and security administrators, which is a strategy that enhances security because no administrator is granted comprehensive access to data. For example, one administrator manages only the keys, while another manages only the database.

TDE is a key-based access control system enforcing authorization using these keys:

Key	Description
Table	For each database table that contains encrypted columns, there is one encryption key used to encrypt all the columns, regardless of the number of encrypted columns in a given table.

Key	Description
Master	Each table's column encryption key is, in turn, encrypted with the database server's master key. The Master key is stored in an Oracle wallet, which is part of the external security module.

TDE is transparent to the application, and no views or additional tables are required. The application logic associated with SQL and table access will continue to work without modification.

To implement this feature within your PeopleSoft application, you need to:

- Determine the fields that are candidates for TDE.
- Set up the Oracle wallet.
- Set the encryption algorithm.
- Encrypt fields.

Note. This feature is available for Oracle databases running 10g R2 and later. Oracle did not provide this feature on any earlier version.

Determining Fields to Encrypt

Examples of information that are candidates for TDE include:

- Names.
- Contact information (address, telephone number, email address, and so on).
- Credit card number.
- Passport number.
- Driver’s license number.
- Age.
- Salary.
- Academic grades, scores, marks, and so on.

Note. Depending on the type of business and country in which you are running your PeopleSoft applications, there may be specific types of information, PII, that needs to be encrypted to comply with regulatory standards.

See Also

Your PeopleSoft application documentation

Managing the Oracle Wallet

With TDE, each individual table has its own table key, which is used to encrypt the selected columns in that table. Each table key is, in turn, encrypted using the TDE master key. The TDE

master key is stored and protected outside the database in an Oracle Wallet, which is a container that stores authentication and signing credentials, including:

- TDE master key.
- PKI private keys.
- Certificates.
- Trusted certificates for SSL.

Encrypted table keys are placed in the data dictionary. When a user enters data into the column defined as encrypted, the Oracle database retrieves the master key from the wallet, decrypts the encryption key for that table from the data dictionary, uses that encryption key on the input value, and stores the encrypted data in the database.

Setting up the Oracle Wallet

Before implementing TDE, creating an Oracle Wallet is required.

Warning! After implementing TDE, the Oracle Wallet must be opened each time a database instance starts (or has been restarted) or else TDE will not work. If the wallet is not open, users will see error messages if they attempt to access any data encrypted using TDE.

To set up an Oracle Wallet for TDE:

1. Specify the wallet location.

   By default, the wallet is created in the directory $ORACLE_BASE/admin/$ORACLE_SID/wallet.

   So, if $ORACLE_BASE is /ds1/product/oracle and $ORACLE_SID is HRDMO, then the wallet will be stored in the directory /ds1/product/oracle/admin/HRDMO/wallet.

   You can set a different directory by specifying it in the sqlnet.ora file located in $ORACLE_HOME/network/admin. For instance, if you want the wallet to be in /orawall directory, place the following lines in the sqlnet.ora file:

```
ENCRYPTION_WALLET_LOCATION =

 (SOURCE=

   (METHOD=file)

     (METHOD_DATA=

       (DIRECTORY=/orawall)))
```

   Note. Oracle recommends adding this location to regular backup utility.

2. Create the wallet.

   Issue the following SQL as a user with the ALTER SYSTEM privilege, such as SYSTEM, SYS, or SYSDBA. In this example, HRMSTKEY is the password.

   ```
   alter system set encryption key

   authenticated by "HRMSTKEY";
   ```

   The preceding command creates the wallet in the specified location, sets the password of the wallet as HRMSTKEY, and opens the wallet for TDE to store and retrieve the master key.

   Note. The password is case-sensitive and must be enclosed in double quotes. The password doesn't show up in clear text in any dynamic performance views or logs.

Opening and Closing the Wallet

After you create the wallet and set the password, every time you start the database, you'll have to open the wallet explicitly, using SYS, SYSTEM, or SYSDBA accounts.

For example,

```
alter system set encryption wallet open authenticated by "HRMSTKEY";
```

To close the wallet:

```
alter system set encryption wallet close;
```

Setting the Encryption Algorithm 

You set the desired encryption algorithm used by TDE on the PeopleTools Options page in the Database Encryption Algorithm edit box.

Access the PeopleTools Options page (PeopleTools, Utilities, Administration, PeopleTools Options).

The algorithms you can enter are:

- Advanced Encryption Standard algorithm with a 128-bit, 192-bit, or 256-bit key.
- Triple DES algorithm with a 168-bit key.

Specify the desired algorithm by entering one of the following values into the Database Encryption Algorithm edit box exactly as it appears below:

- AES128
- AES192

- AES256
- 3DES168

Note. You must specify an encryption algorithm to enable the Encrypt option for a field definition in Application Designer.

Encrypting Fields

You encrypt fields in Application Designer by selecting the Encrypt check box on a field definition, and then creating a table or altering an existing table.

Note. The Encrypt check box is enabled only on Oracle databases running version 10g R2 or later that also have an encryption algorithm specified in the Database Encryption Algorithm edit box on the PeopleTools Options page.

These PeopleSoft field types can be encrypted:

- Character
- Long Character (see note below)
- Number
- Signed number
- Date
- DateTime
- Time

Note. Long Character field types may only take advantage of TDE when the following conditions are true: the field length is greater than 0 and less than 1334 *and* the Raw Binary field attribute *is not* set.

These PeopleSoft field types *can not* be encrypted:

- Image
- Image reference
- Attachment

After you define the field to be encrypted, and either create a table or alter an existing table containing that field definition, the Build feature generates DDL SQL containing the ENCRYPT clause in the following syntax:

```
ENCRYPT using 'ALGORITHM'
```

For example,

```
ALTER TABLE  PS_AM_BI_HDR

MODIFY (CR_CARD_NBR ENCRYPT using 'AES256' NO SALT);
```

Note. If you are using Oracle Database version 10.2.0.4 or higher, the syntax includes the NOMAC parameter. For example, ALTER TABLE PS_AM_BI_HDR MODIFY (CR_CARD_NBR ENCRYPT using 'AES192' 'NOMAC' NO SALT);

The NOMAC parameter reduces the storage requirements and provides improved performance.

See your Oracle database documentation for more information on NOMAC.

When DDL SQL containing the ENCRYPT clause is run against the database, Oracle:

- creates a cryptographically secure encryption key for the table containing the column.
- encrypts the clear text data in the column, using the specified encryption algorithm.

Managing Fields Encrypted for TDE 

This section covers these topics related to the ongoing maintenance of encrypted fields:

- Decrypting fields.
- Regenerating an encryption key.
- Upgrading TDE encrypted fields.

Decrypting Fields

If you decide that you no longer want a field encrypted for TDE, you can issue a SQL ALTER operation using the DECRYPT clause. For example, assume you wanted to decrypt the SSN field on the ACCOUNT table.

```
ALTER TABLE ACCOUNT MODIFY (SSN DECRYPT);
```

Regenerating An Encryption Key

Situations where you might consider regenerating a table encryption key include:

- You suspect a table key has been compromised.
- You want to take advantage of a different encryption algorithm.

You regenerate a table encryption key by issuing a SQL ALTER operation using the REKEY clause. For example, assume you wanted to rekey the PS_AM_BI_HDR table to take advantage of AES256.

```
ALTER TABLE PS_AM_BI_HDR REKEY using 'AES256';
```

This creates a new table key and recreates the encrypted column values using the new table key.

Upgrading TDE Encrypted Fields

All metadata field definitions are delivered with no-encryption attributes enabled. PeopleSoft applications will not deliver any metadata indicating encryption enabled for any field for an initial installation database file, project, or a PeopleTools or PeopleSoft application patch.

If you customize the field by adding TDE encryption, you need to keep track of the fields and associated record definitions and ensure that you maintain the desired encryption status through any upgrades that you perform.

See Your PeopleSoft upgrade documentation

Altering Tables With TDE Encrypted Fields

When altering tables with TDE encrypted fields using the Alter in Place option, Application Designer automatically switches the Index Creation Options selection to Recreate index only if modified even if you specifically select Recreate index if it already exists in the Build Settings dialog box.

Protecting and Managing PeopleSoft Applications with Database Vault

This section provides an overview and discusses:

- Restricting access for the access ID.
- Restricting access for PSFTDBA ID.
- Using multiple, alternate, access IDs.

Understanding Oracle Database Vault

Oracle Database Vault provides an extra layer of security that protects a database against insider security threats. One of Database Vault's key features is that it protects PeopleSoft application data from being accessed by super-privileged users, such as DBA or system administrators, but it still allows them to maintain the Oracle database.

A super-privileged user, such as a DBA, should not have access to PeopleSoft application data. Application data can include salary, identification numbers, credit card numbers, and other personal information. On the other hand, the DBA must still be able to perform database maintenance, such as back up and recovery. Database Vault allows DBAs to do their jobs, but does not allow the DBA to have access to application data.

PeopleTools has validated the use of Oracle Database Value with PeopleSoft applications. From that validation effort we've provided sample PeopleSoft DB Vault security policies. The sample policies are available on Oracle Technology Network (OTN).

See http://www.oracle.com/technology/software/products/database_vault/index.html

These sample policies lock the database to allow all PeopleSoft application processes to access the database, while restricting any super user, like a DBA, from viewing the data using any

Oracle delivered query tool. These policies illustrate a minimal usage of Database Vault functionality and may be modified or enhanced based on your specific level of required database security. The following table illustrates how the implementation of the example Database Vault policies affects the PeopleSoft Access ID and end-users, such as VP1 or PS.

User Account	Database Vault
SYSADM (Peoplesoft Access ID)	Before Database Vault, the Oracle DBA would use the Access ID for all database maintenance tasks, and they could view all of the data in the database. For example, a DBA might have used the PeopleSoft Access ID during all system testing to query the database when they needed to verify data in the database. Once Database Vault is enabled, the Access ID will no longer be able to access SQL*Plus, for example.
PSFTDBA (Account for DBAs)	With Database Vault enabled, the Oracle DBA responsible for applying PeopleSoft upgrades will no longer use the PeopleSoft Access ID. The DBA will now use the new PSFTDBA account to login to SQL*Plus and perform database maintenance tasks. The PSFTDBA account does not allow the DBA to run SELECT statements on the database tables, but INSERT, UPDATE, and DELETE are allowed.
General PeopleSoft user IDs (VP1, PS, and so on)	The PeopleSoft "end-user" IDs, such as VP1, are *not* affected by Database Vault. Database Vault is transparent to VP1 and other PeopleSoft end-users.

Restricting Access For the Access ID

In the PeopleSoft system, the access ID is the Oracle owner of all schema objects in a PeopleSoft database. With Database Vault you can restrict Oracle users other than the access ID from having 'SELECT' privilege on any access ID objects.

This restrictive usage is supported by using the sample PeopleSoft Database Vault security policies. When the sample PeopleSoft Database Vault security policies are implemented and Database Vault is enabled on a PeopleSoft database running on Oracle, the policies allow the access ID to do everything it currently needs to do on behalf of PeopleSoft components.

By design, all DML including SELECT DML is allowed by the access ID if the DML is issued through a "known" PeopleTools component, as defined in the sample PeopleSoft Database Vault security policies.

SELECT DML access is restricted for the access ID if not executed through a defined PeopleTools component.

SQLPlus and other ad hoc query tools are not explicitly defined in the sample policies and therefore cannot be used to issue SELECT DML against the database.

Restricting Access For PSFTDBA ID

The sample policies and scripts provide for non-access ID access to the database through the Oracle user, PSFTDBA. This user is intended to be used when you need SQLPLUS access to the system.

In order for DBAs to perform system maintenance, upgrade tasks, and so on, the sample policy scripts create the PSFTDBA account. With this account the following actions are allowed on database tables:

- INSERT
- UPDATE
- DELETE

The sample PeopleSoft Database Vault security policies restrict the PSFTDBA ID from performing a SELECT against the access ID's objects. If you use the PSFTDBA account to run a SELECT statement, an error message similar to the following appears:

```
sp-hp15:$ sqlplus PSFTDBA/PSFTDBA@Q8501123



SQL*Plus: Release 11.1.0.6.0 - Production on Wed Apr 9 10:45:36 2008



Copyright (c) 1982, 2007, Oracle.  All rights reserved.



Connected to:

Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - 64bit Production

With the Partitioning, Oracle Label Security, Oracle Database Vault and

Real Application Testing options



SQL> select * from Q8501123.PSSTATUS;

select * from Q8501123.PSSTATUS
```

```
                         *

ERROR at line 1:

ORA-01031: insufficient privileges
```

The PSFTDBA ID is designed so that your DBA's use it rather than the access ID to increase security when performing database maintenance. When performing some tasks, keep in mind that PSFTDBA does not have sufficient access to the database to perform all PeopleSoft maintenance tasks, such as all upgrade tasks.

For example, when running SQRs from the workstation, the PSFTDBA user ID cannot run SELECTs on the database to generate reports. This is a defined PeopleSoft Database Vault policy restriction. SQR's should be run as scheduled Process Scheduler jobs on the server. Also, when applying PeopleSoft upgrades involves some steps that require access to the database using the access ID. For example, in some cases you need to run Data Mover in bootstrap mode using the access ID/password. Data Mover scripts cannot be run as PSFTDBA. In these cases, the key limitation to keep in mind is that the PSFTDBA ID cannot run a select against any access ID owned tables, which includes tables required for Data Mover to log in to the system.

In cases, where you need SELECT access for certain features (SQR, Data Mover, and so on) you can configure a set of specific, alternative ID's to be used for PeopleSoft upgrade tasks while still remaining in compliance with the Database Vault policies.

Using Multiple Alternate Access IDs 

The sample PeopleSoft Database Vault security policies provide protection of highly sensitive information in the PeopleSoft tables from database "super users." In some cases, you may need a more tailored access, such as in the cases of upgrades, patching, auditing, and the separation of duties for the PeopleSoft Access ID.

You can leverage Database Vault so that PeopleSoft tables, procedures and triggers could be protected can still be protected while allowing special access to complete upgrade and maintenance tasks. The privileges in the Database Vault PeopleSoft template can be given to the multiple, alternate, access IDs. By using multiple, alternate, access IDs to perform PeopleSoft maintenance, you can mitigate the issues involved with distributing the password of the base access ID to multiple users.

The multiple, alternate, access IDs (PSFTDBAnn) technique has been tested with Database Vault in the field on PeopleSoft installations and offers a solution where unique, identifiable accounts can be used to perform PeopleSoft patching and upgrades. These accounts can be limited to the modules and machine names from which the PSFTDBAnn ID can run. These accounts also can be heavily audited, to make sure that they do not introduce malicious code, which removes the need to implement heavy auditing on the base access ID account.

With multiple, alternate, access IDs you can:

- Use multiple, alternate, access IDs that can be used just for PeopleSoft upgrade/maintenance.
- Create PSFTDBAnn accounts that have many auditing options enabled, not affecting the use of the production access ID (SYSADM).
- Use Oracle's Audit Vault to enhance the separation-of-duties when it comes to centrally managing audit information.
- Configure Database Vault so that the PSFTDBAnn account can be restricted to particular machines and times, and so on.
- Tale advantage of Database Vault's strictly DBA account(s) (PSFTDBA) that can modify the database and system, but not issue selects on tables in the PeopleSoft Realm. The PSFTDBA account can do many DBA activities such as alter tablespaces, examine performance, start and stop the system, but not see sensitive information. The PSFTDBA account can apply Oracle Database Patches, but not apply PeopleSoft type of patches.
- Restrict knowledge of the access ID password, as it is no longer required for PeopleSoft maintenance.
- Address many more of the concerns third party auditors are identifying on systems that contain highly sensitive information in PeopleSoft applications.

In the following examples, the unofficial account "PSFTDBAn" represents multiple access IDs, although it can be almost any name. The PSFTDBAn accounts need to retain the ability to do 'SELECTS' on PeopleSoft objects. This technique leverages a protected Login Trigger that alters the CURRENT_SCHEMA, so that the PSFTDBAn accounts can *act* as the access ID (SYSADM) account, but preserve the user's identity (PSFTDBA1) when running any commands.

To configure multiple, alternate, access IDs:

1. Create one to 'n' multiple, alternate, access IDs (authorized Oracle USERS):

```
  create user psftdba1 identified by oracle_1;
  create user psftdba2 identified by oracle_1;
  create user psftdba3 identified by oracle_1;
```

5. Grant minimal privilidges to these alternate authorized USERS:

```
  grant connect,resource to psftdba1;
  grant connect,resource to psftdba2;

grant connect,resource to psftdba3;
```

8. CREATE an Oracle instance level logon trigger to issue an ALTER SESSION SET CURRENT_SCHEMA whenever an alternative authorized user logs into the instance.

```
  drop trigger psft_login_trg;
  create or replace trigger psft_login_trg
  after logon on database

  begin
  -- * use dvf if in a database vault environemnt.
  -- * database vault would also help protect the peoplesoft realm, and
  logon trigger, and so on
  -- if dvf.f$session_user in ('PSFTDBA1' , 'PSFTDBA2', 'PSFTDBA3') then
  if sys_context('userenv','session_user') in in ('PSFTDBA1' ,
 'PSFTDBA2',
   'PSFTDBA3') then
```

```
20.  execute immediate 'alter session set current_schema=’SYSADM';
21.  end if;
22.  end;
23.  /
24.
```

Every time one of the alternative authorized USERs logs into the instance, an ALTER SESSION SET CURRENT_SCHEMA=ACCESSID is issued. From here on in any operation performed that is unqualified would be done in the ACCESSID schema.

For example, if the 'PSFTDBA1' were logged into the database directly using SQLPLUS or indirectly using Data Mover, then any 'VALID' operation performed that is unqualified would be done in the ACCESSID schema. All of ‘PSFTDBA1's actions on the database could be audited if the Oracle Auditing facility (Audit Vault) were used. If you need to verify you have database connectivity, you can use the PSFTDBAn account for your test. Data Mover and SQR testing from the workstation will be able to use the PSFTDBAn account.

Working With Oracle 11g Security Features

Oracle 11g introduces security features, which from a database security perspective, increase restrictions for database access. These changes are part of the "Secure By Default" configuration of 11g. These changes include setting a defined limit for the PASSWORD_LIFE_TIME and PASSWORD_GRACE_TIME associated with the default profile. This section discusses how PeopleSoft systems are affected and what your options are.

Understanding Default Profiles

All Oracle users created in an instance are assigned a default profile, such as the default profile delivered with 11g. There are differences between the default profiles for 10g and 11g.

Oracle Database Version	Default Profile Values
Oracle 10g	PASSWORD_LIFE_TIME: UNLIMITED PASSWORD_LOCK_TIME: UNLIMITED PASSWORD_GRACE_TIME: UNLIMITED
Oracle 11g	PASSWORD_LIFE_TIME: 180 PASSWORD_LOCK_TIME: 1 PASSWORD_GRACE_TIME: 7

For pre-11g Oracle releases, the default profile did not specify a PASSWORD_LIFE_TIME limit. As such, by default, the password for a given Oracle user never expired.

PASSWORD_LOCK_TIME and PASSWORD_GRACE_TIME were also unlimited. For 11g, the default profile has a PASSWORD_LIFE_TIME of 180 days. PASSWORD_LOCK_TIME and PASSWORD_GRACE_TIME also have limits.

For a PeopleSoft installation on the Oracle platform, several Oracle user IDs are created during the installation. Those Oracle users are:

- ACCESSID (default is SYSADM)
- CONNECT ID (default is people)
- PS (owns the PSDBOWNER table)

The ACCESSID is the schema owner for all database objects related to a specific PeopleSoft application installation. The ACCESSID and ACCESSID password are stored and encrypted in the PeopleSoft security table PSACCESSPRFL.

```
SQL> descr SYSADM.PSACCESSPRFL

 Name                                      Null?    Type

 ----------------------------------------- -------- ----------------------



 SYMBOLICID                                NOT NULL VARCHAR2(8 CHAR)

 VERSION                                   NOT NULL NUMBER(38)

 ACCESSID                                  NOT NULL VARCHAR2(16 CHAR)

 ACCESSPSWD                                NOT NULL VARCHAR2(16 CHAR)

 ENCRYPTED                                 NOT NULL NUMBER(38)


SQL> SELECT  * from SYSADM.PSACCESSPRFL;;


SYMBOLIC       VERSION        ACCESSID               ACCESSPSWD
       ENCRYPTED

--------               ----------             ----------------       ---------
-------                ----------

SYSADM1           7             sBzLcYlPrag=     sBzLcYlPrag=                1
```

The connect ID is a pseudo logon which allows PeopleSoft to associate multiple PeopleSoft user IDs to the same connect ID. The connect ID has the minimum privileges required to connect to

the database (only SELECT privileges on specific PeopleTools tables). After a connection has been established using the connect ID, PeopleSoft security uses the PeopleSoft user ID to control access to objects in the database. The PeopleSoft signon process validates the connect ID on the server, rather than the user ID. The connect ID simplifies database security maintenance, as you don't need to maintain access for all PeopleSoft users, just for the connect ID.

The PS ID is used once, during PeopleSoft database creation, to create the PSDBOWNER table. Once this table has been created, read access and write privileges are made public to everyone, then the PS user ID privileges are revoked.

Encountering Issues Related to Oracle 11g Security 

When the PASSWORD_LIFE_TIME has been reached, the PeopleSoft Oracle users (in this case the PeopleSoft ACCESSID and CONNECT ID) will be locked out of the database. This means that any PeopleSoft process cannot access the database, such as application server, Process Scheduler, COBOL, Data Mover, and so on.

If this occurs you will see any of the following Oracle database error messages:

```
ORA-28000: the account is locked

Cause: The user has entered wrong password consequently for maximum number
of⇒

 times specified by the user's

profile parameter FAILED_LOGIN_ATTEMPTS, or the DBA has locked the account

Action: Wait for PASSWORD_LOCK_TIME or contact DBA




ORA-28001: the password has expired

Cause: The user's account has expired and the password needs to be changed

Action: change the password or contact the DBA




ORA-28002 the password will expire within string days

Cause: The user's account is about to about to expire and the password needs
to be
```

```
 changed.

Action: Change the password or contact the database administrator.
```

These messages may appear in a SQL trace, an application server log, a Process Scheduler log, or in an error message in the GUI when attempting to access the database (signon to Application Designer or Data Mover). The following are some select examples of what you can expect to see in log and trace files.

The trace will show the login failing as follows:

```
CONNECTID.




2-4     13.06.56    1.581000 Cur#0.6060.QE849C42 RC=28001 Dur=1.581000

Connect=Primary/QE849C42/people/

2-5     13.06.56    0.000000 Cur#0.6060.QE849C42 RC=-1 Dur=0.000000 XER

rtncd=761802124 msg=

2-6     13.06.56    0.000000 Cur#0.6060.QE849C42 RC=0 Dur=0.000000 ERR
rtncd=28001

 msg=ORA-28001: the password has expired
```

The following illustrates an application server or Process Scheduler boot with passwords already expired:

```
PeopleTools 8.xx.07 Client Trace - 2008-10-24



PID-Line  Time        Elapsed  Trace Data...

--------  --------  ---------- -------------------->

 1-1      14.25.45             Tuxedo session opened {oprid='QEDMO',

appname='Two Tier', addr='//TwoTier:7000', open at 01C67EC8, pid=4956}

 1-2      14.25.45    0.058000 Cur#0.4956.QE849C41 RC=0 Dur=0.003000 ---
router

 PSORA load succeeded
```

```
 1-3      14.25.45    0.155000 Cur#0.4956.QE849C41 RC=0 Dur=0.155000 INI

 1-4      14.25.45    0.192000 Cur#0.4956.QE849C41 RC=28002 Dur=0.192000

Connect=Primary/QE849C41/people/

 1-5      14.25.45    0.000000 Cur#0.4956.QE849C41 RC=-1 Dur=0.000000 XER

rtncd=761800508 msg=

 1-6      14.25.45    0.000000 Cur#0.4956.QE849C41 RC=0 Dur=0.000000 ERR

rtncd=28002 msg=ORA-28002:

the password will expire within 7 days

 1-7      14.25.48    2.718000 Cur#0.4956.notSamTran RC=0 Dur=0.000000 DON

 1-8      14.25.51    2.742000 Tuxedo session opened { DisconnectAll
at01C67EC8,

 pid=4956}
```

The following illustrates a client trace of a application server or Process Scheduler boot:

```
PeopleTools 8.49.07 Client Trace - 2008-10-24



PID-Line  Time        Elapsed  Trace Data...

--------  --------  ---------- -------------------->

 1-1      14.30.38             Tuxedo session opened {oprid='QEDMO',

appname='Two Tier', addr='//TwoTier:7000', open at 01C67EC8, pid=3328}

 1-2      14.30.38    0.056000 Cur#0.3328.QE849C41 RC=0 Dur=0.004000 ---
router

 PSORA load succeeded

 1-3      14.30.38    0.238000 Cur#0.3328.QE849C41 RC=0 Dur=0.238000 INI

 1-4      14.30.38    0.529000 Cur#1.3328.QE849C41 RC=0 Dur=0.529000

Connect=Primary/QE849C41/people/

 1-5      14.30.38    0.036000 Cur#1.3328.QE849C41 RC=0 Dur=0.000000 GET

type=1003 dbtype=4
```

```
 1-6      14.30.38    0.000000 Cur#1.3328.QE849C41 RC=0 Dur=0.000000 GET

type=1004 release=11

 1-7      14.30.38    0.076000 Cur#1.3328.QE849C41 RC=0 Dur=0.000000 COM

Stmt=

SELECT OWNERID FROM PS.PSDBOWNER

WHERE DBNAME=:1

.

1-41     14.30.40    0.200000 Cur#1.3328.QE849C41 RC=0 Dur=0.200000
Disconnect

 1-42     14.30.40    0.251000 Cur#0.3328.QE849C41 RC=28002 Dur=0.220000

Connect=Primary/QE849C41/QE849C41/

 1-43     14.30.40    0.000000 Cur#0.3328.QE849C41 RC=-1 Dur=0.000000 XER

rtncd=18874368 msg=

 1-44     14.30.40    0.000000 Cur#0.3328.QE849C41 RC=0 Dur=0.000000 ERR

rtncd=28002 msg=ORA-28002: the password will expire within 7 days

 1-45     14.30.42    2.293000 Cur#0.3328.notSamTran RC=0 Dur=0.000000 DON

 1-46     14.30.43    0.788000 Tuxedo session opened { DisconnectAll

at01C67EC8, pid=3328}
```

The failure and return of the GRACE PERIOD warning message gives you time to react before the password actually expires, enabling you to be proactive and reset or change the ACCESSID and/or the CONNECT ID password(s).

Oracle 11g Security Configuration Options

This section discusses options for dealing with Oracle 11g security, including:

- Setting the PASSWORD_LIFE_TIME to unlimited.
- Creating a PeopleSoft-specific profile.
- Resetting the PeopleSoft installation user IDs.
- Changing the PeopleSoft installation user IDs.

Setting the PASSWORD_LIFE_TIME to Unlimited

You can set the PASSWORD_LIFE_TIME in the default profile to unlimited. If this is done prior to creating the PeopleSoft-specific Oracle user IDs used for the PeopleSoft database installation, then the default behavior will mimic the pre-Oracle 11g behavior.

This can be done by creating the ACCESSID and CONNECT ID using the following command:

```
ALTER PROFILE DEFAULT LIMIT PASSWORD_LIFE_TIME UNLIMITED

;
```

Note. While feasible, this particular solution is counter to the secure by default positioning of Oracle 11g and to regulations requiring periodic changes to important passwords.

Creating a PeopleSoft-Specific Profile

You can create a PeopleSoft-specific profile which sets the PASSWORD_LIFE_TIME to unlimited. Creating the new PeopleSoft profile should be done when you create the database rather than altering PeopleSoft users from the default profiles to the PeopleSoft-specific profiles. Switching the a PeopleSoft-specific profile after you have created the PeopleSoft-specific users expired password limits does not automatically modify the expiry_date column in USER_USERS (done when creating the users with the default profile).

Create the ACCESSID and CONNECT ID user IDs using the delivered scripts, PS_HOME/scripts/PSADMIN.SQL and PS_HOME/scripts/CONNECT.SQL. After doing so, the PeopleSoft Oracle user IDs would have the default profile assigned. Alter the ACCESSID and CONNECT ID user IDs to make use of the alternate profile rather than the default. This can be done using the following commands:

```
CREATE PROFILE PSPROFILE LIMIT  PASSWORD_LIFE_TIME UNLIMITED

;
```

This creates the PSPROFILE profile with password limits values set. All values not explicitly listed are derived from the default profile.

The following statements alter both the default ACCESSID and CONNECT ID to utilize the PSPROFILE profile with the password limit set for PASSWORD_LIFE_TIME to unlimited:

```
ALTER USER SYSADM PROFILE PSPROFILE

;

ALTER USER PEOPLE  PROFILE PSPROFILE

;
```

Note. While feasible, this solution will allow the profile expiration behavior to mimic the pre-Oracle 11g behavior, but this runs counter to the intent of regulations that require changing critical passwords on a regular basis.

Resetting the PeopleSoft Installation User IDs

You can reset the PeopleSoft installation Oracle user ID passwords (the ACCESSID and CONNECT ID) in all of the places it needs to be reset. After the passwords expire, reset them to the original value. You can reset the password using the PASSWORD command or by ALTER USER command.

Note. If using Database Vault, then only the database vault account manager can reset the account, because the access ID cannot login to SQLPLUS to change the password.

Note. While feasible, this option runs counter to the intent of regulations that require changing critical passwords on a regular basis.

Changing the PeopleSoft Installation User IDs

The recommended option is to change the PeopleSoft installation required Oracle user ID passwords (the ACCESSID and CONNECT ID) after they have expired, and reflect those changes in all required locations. This option enables you to conform to regulations that require changing critical passwords on a regular basis.

If the password expires and an Oracle user ID password is changed within the Oracle database for the ACCESSID or CONNECT ID, the PeopleSoft system will still have the old password stored in the PeopleSoft security metadata tables and configuration files. These changed passwords will have to be reflected in the PeopleSoft security metadata tables and configuration files as well as the database.

At the database level, you can use the PASSWORD and ALTER USER commands to change the ACCESS ID and CONNECT ID passwords. For example:

```
C:\Documents and Settings\>sqlplus people/peop1e@QE849C42


SQL*Plus: Release 10.2.0.3.0 - Production on Tue Oct 21 10:55:57 2008


Copyright (c) 1982, 2006, Oracle.  All Rights Reserved.


ERROR:

ORA-28001: the password has expired
```

```
Changing password for people

New password: <changed to  'peop2e'>

Retype new password: <changed to  'peop2e'>

Password changed


SQL> exit
```

Or,

```
ALTER USER QE849C42 IDENTIFIED BY CHANGEPW ACCOUNT UNLOCK;


User altered.


ALTER USER people IDENTIFIED BY peop2e ACCOUNT UNLOCK;


User altered.


SQL> exit
```

Note. You may also have to include the UNLOCK keyword to unlock the account (if the password retry has been exceeded).

In PeopleTools, open Configuration Manager and change the Connect Password value on the Startup tab.

Then, open Data Mover in bootstrap mode (using the new ACCESSID password) to run the necessary commands to change the ACCESSID passwords on the appropriate PeopleSoft metadata tables. For example,

```
SET LOG c:\temp\changeaccessidpswd.out;

UPDATE PSSTATUS SET OWNERID = 'QE849C42';

UPDATE PSOPRDEFN SET OPERPSWD = OPRID, ACCTLOCK=0, ENCRYPTED = 0;
```

```
UPDATE PSACCESSPRFL SET ACCESSID = 'QE849C42', ACCESSPSWD = 'CHANGEPW',

VERSION = 0, ENCRYPTED = 0;

ENCRYPT_PASSWORD *;
```

Note. For Oracle 11g, the password is case sensitive.

Lastly, apply the connect ID changes to the psprcs.cfg and psappsrv.cfg configurations files and rebuild the domains. For example:

```
[Startup]

;=============================================================================

; Database Signon settings

;=============================================================================

DBName=QEDMO

DBType=ORACLE

UserId=QEDMO

UserPswd==QEDMO

ConnectId=people

ConnectPswd=peop2e

ServerName=
```