

Oracle Public Shutdown for JDEdwards EnterpriseOne

*An Oracle JD Edwards EnterpriseOne
Security Considerations Paper*

November 2007



PURPOSE STATEMENT

This document provides Oracle specific database security considerations relative to your JD Edwards EnterpriseOne system. Many factors should be considered when architecting an all encompassing security solution. This document specifically addresses database security concerns.

This document is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle. Updates and enhancements are provided in accordance with Oracle's Technical Support Policies at:

www.oracle.com/support/collateral/oracle-technical-support-policies.pdf

Oracle Public Shutdown for JDEdwards EnterpriseOne

Table of Contents

PURPOSE STATEMENT.....	i
Securing Your Oracle Database for EnterpriseOne	1
Understanding the JDEdwards EnterpriseOne Security Server.....	1
About the Enclosed Scripts.....	3
Step One: Create Database Roles for EnterpriseOne	7
Replacement of Substitution Variables	8
Running 01_set_run_CreateRolesE1.bat file:	10
Step Two: Grant Table Access to 'Role' / Revoke Table Access from 'public'	11
Replacement of Substitution Variables	11
Running 02_set_run_GrantRevokeE1.bat file:	15
Step Three: Grant Role to EnterpriseOne Proxy Users	17
Replacement of Substitution Variables	17
Running 03_set_run_GrantRoleE1.bat file:.....	20
Step Four: Evaluate & Compare Table Privileges by Role and 'public'	21
Replacement of Substitution Variables	22
Running 04_set_run_CompareTablePrivs.bat:.....	25
Step Five: Update the Schema Owner User Passwords.....	26
Replacement of Substitution Variables	26
Running 05_set_run_UpdSchemaPswdE1.bat file:	28
UNDO :: Granting public to EnterpriseOne Tables (Undo Prior Work)	30
Replacement of Substitution Variables	30
Running 99_set_UNDO_GrntPublicE1.bat file:.....	33
Evaluate & Compare Table Privileges by Role and 'public'	33
Appendix A: Result 02_set_run_GrantRevokeE1.bat.....	36
Appendix B: Purging the DBA_RECYCLEBIN	44
Appendix C: Securing HRM and Payroll Tables	45
Appendix D: Maintaining a Secured Database	47

Oracle Public Shutdown for JDEdwards EnterpriseOne

Securing Your Oracle Database for EnterpriseOne

Oracle has developed a detailed approach to secure your Oracle database for JDEdwards EnterpriseOne. The set of instructions contained herein are supplemented by database scripts. Presently JDEdwards EnterpriseOne delivers their databases providing full access to *public*. *Public* acts as a default role granted to every database user. By implementing this process we will ultimately grant access to each database table to one or more database roles while revoking access to *public*. The database roles will then be associated to each JDEdwards EnterpriseOne's system (proxy) user as deemed appropriate. This will make the database tables accessible only to those database users associated to a particular database role.

In the absence of this security any database user may access any table. These instructions have been put together to demonstrate a methodology to secure your JDEdwards EnterpriseOne Oracle database. **Note: Oracle JDEdwards EnterpriseOne provides a downloadable SAR 8289283. This is available via Customer Connection / Update Center / JD Edwards EnterpriseOne section. Enter the SAR number in the search field. The download contains a set of scripts to supplement this "Oracle Public Shutdown for JDEdwards EnterpriseOne" document.**

Individuals who implement this solution should have the following qualifications:

- An understanding of Oracle database administration and management concepts.
- A knowledge of EnterpriseOne Common Foundation, Configurable Network Computing (CNC), and System Administration concepts.
- An understanding of one's business needs, development methodology, and version control.
- A familiarity with any integrated third-party product interacting with JDEdwards EnterpriseOne. This will permit one to evaluate whether the security mechanisms that are put in place might compromise its use. Also, one could leverage these same concepts to secure further any mechanisms facilitating product integration.

THIS PUBLICATION IS PROVIDED "AS IS" without warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. This publication could include technical inaccuracies or typographical errors. This publication does not make recommendations, implied or actual. It provides guidelines; however, due the wide variety of networking, hardware and software configurations found in EnterpriseOne/OneWorld installations, no guarantees can be made that specific results are achievable in any particular installation. Changes are periodically added to the information herein. These changes will be incorporated in new editions of the publication. J.D. Edwards may make improvements and/or changes at any time to the product(s) and/or the program(s) described in this publication.

Understanding the JDEdwards EnterpriseOne Security Server

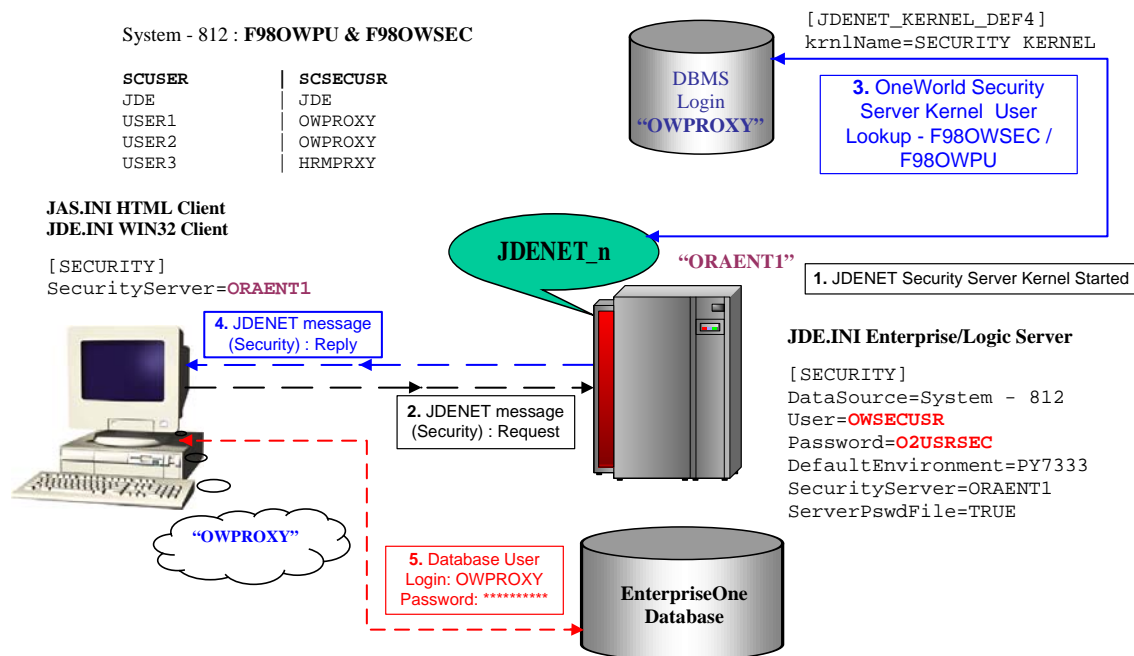
The JDEdwards EnterpriseOne Security Server makes it possible to control and limit explicit database access. This is accomplished by separating JDEdwards EnterpriseOne users from the platform database user. With the exception of JDE (and PSFT) the JDEdwards EnterpriseOne users are strictly ERP product users. Each user will be associated with a database (proxy) user. The association between database users to ERP user can be one to one or one to many. For example, it is possible to create a set of database users grouped by major product suite, e.g. JDE, JDEAEC, JDECNC, JDECRM, JDEDST, JDEFIS, JDEFIN, JDEHRM, JDEMFG, JDEPAY, JDESCM, etc. One of these database users, e.g. JDEFIN could be associated with employees in the Financials department. Thus, the relationship between database (proxy) users to ERP users in this case is one to many.

An EnterpriseOne ERP user would not have explicit access to the database. Rather they must login through EnterpriseOne in order to access the database tables via available applications. While the JDEdwards EnterpriseOne Security Server is running, ERP users may login via HTML, a WIN32 developer client, etc. using a valid ERP set of credentials, e.g. userid/password.

One or more Security Servers can be configured within your JDEdwards EnterpriseOne solution. The Security Server is one of many configurable kernel services run by an Enterprise (Logic, Application, or Batch) server. A JDEdwards EnterpriseOne JAS HTML, Portal, WIN32 developer clients, etc. each interact with our Security Server in order to gain access to the JDEdwards EnterpriseOne services and ultimately the database via product applications. The Security Server initially authenticates the ERP product user's credentials and then provides an encrypted proxy user and password set of credentials which are passed via our code to access database tables.

The JDEdwards EnterpriseOne Security Server process is illustrated in Figure 1 below. To begin the JDENET_K Security Server kernel must be running. Then a request from an EnterpriseOne appliance is sent to the Security Server sending ERP user login credentials. The Security Server kernel checks the F98OWSEC and F98OWPU tables to authenticate the ERP user's credentials and then returns the associated proxy user's credentials in an encrypted stream. The proxy user's credentials are returned to the EnterpriseOne appliance allowing the appliance to connect to the database using a database user account.

Figure 1: JDEdwards EnterpriseOne Security Server



The relationship between JDEdwards EnterpriseOne ERP users and proxy users is maintained in table F98OWSEC (application P98OWSEC for OneWorld Security). In releases 8.9 and higher table F98OWPU was introduced to further segment the system (proxy) user credential information. The F98OWPU table is accessed via application P980001 for *Work with OneWorld System Users*. In the sample query shown below the first two columns illustrate how the ERP user and proxy user are cross-referenced. The third column maps to the related system (proxy) user table.

Figure 2: Security Server Tables (F98OWSEC & F98OWPU)

Query: SELECT T0.SCUSER, T0.SCSECUSR, T1.PUSECUSR
FROM SY812.F98OWSEC T0, SY812.F98OWPU T1
WHERE T0.SCSECUSR = T1.PUSECUSR;

Sample Result Set:

SCUSER	SCSECUSR	PUSECUSR
JDE	JDE	JDE
JDEUSR1	JDEUSR1	JDEUSR1
JDEUSR2	JDEUSR2	JDEUSR2
JDEUSR3	JDEFIN	JDEFIN
JDEUSR4	JDEHRM	JDEHRM
KJUDSON	JDEPAY	JDEPAY
KPOND	JDEHRM	JDEHRM
BWILSON	JDEFIN	JDEFIN
MOCONNER	JDESCM	JDESCM
JKOGER	JDEMFG	JDEMFG
JMAKUSKY	JDECNC	JDECNC
...		

About the Enclosed Scripts

Scripts have been provided as guidance and to supplement this process. These scripts are iterative in process and may be modified further to meet your specific business needs. These scripts are intended to provide guidance to the methodology described herein.

Table 1: About the Enclosed Scripts

Script Name	Script Purpose
01_set_run_CreateRolesE1.bat	Create database roles for use by EnterpriseOne (E1).
02_set_run_GrantRevokeE1.bat	Grant access to specified role for each E1 tables while revoking access from <i>public</i> for the same table.
03_set_run_GrantRoleE1.bat	Associate database role with E1 system (proxy) user.
04_set_run_CompareTablePrivs.bat	Check table privileges by database role(s) and <i>public</i> .
05_set_run_UpdSchemaPswdE1.bat	Change the default schema owner password.
99_set_UNDO_GrntPublicE1.bat	Undo revocation of table access from <i>public</i> by granting access back to <i>public</i> .

Script Requirements

- These scripts must be run from a Windows machine which has an Oracle client installed. The Oracle client should match the Oracle database with regard to release level, e.g. 10g.
- The Oracle client must have the EnterpriseOne database SID configured via Network Configuration Assistant, e.g. tnsnames.ora.
- All of the scripts require oracle user credentials. In some cases the user account '*system*' is a requirement. Ultimately the user account information provided within the scripts will be used to login to the database in order to execute an associated script. A sample sqlplus login is shown in Figure 3.

Figure 3: Sample SQLPLUS Login

```
C:\>sqlplus system/oracle1@mlsan221

SQL*Plus: Release 10.2.0.2.0 - Production on Thu Nov 1 17:52:56 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL>

~~~ above login w/ system account | below login w/schema owner account ~~~

C:\>sqlplus proddta/proddta@mlsan221

SQL*Plus: Release 10.2.0.2.0 - Production on Thu Nov 1 18:07:31 2007

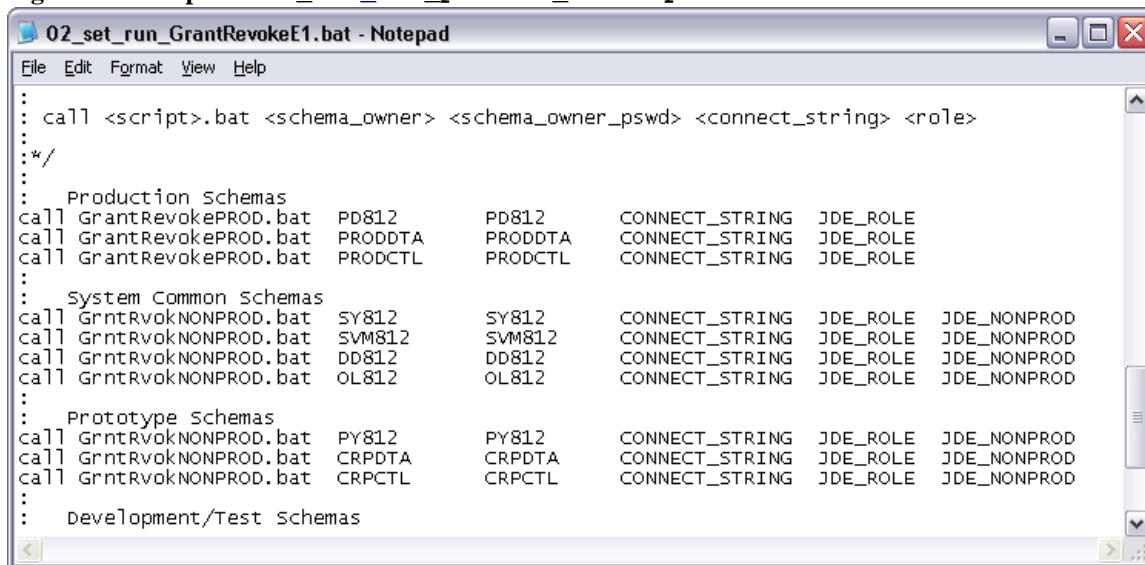
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL>
```

Each set of scripts is comprised of at least three components. The first of these is a batch file with the name **nn_set_run_process_description.bat**. This batch file allows the individual who implements this solution to set specific parameters intended for the described process. These parameters include required login information (user/password@connect_string). In the example below the script calls a particular batch file and passes a series of arguments to this file.

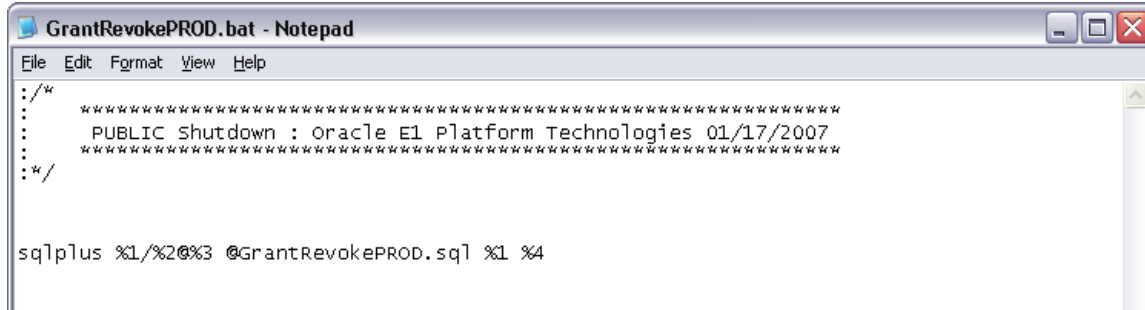
Figure 4: Example of nn_set_run_process_description.bat



```
02_set_run_GrantRevokeE1.bat - Notepad
File Edit Format View Help
:
: call <script>.bat <schema_owner> <schema_owner_pswd> <connect_string> <role>
:
: */
:
: Production Schemas
call GrantRevokePROD.bat PD812 PD812 CONNECT_STRING JDE_ROLE
call GrantRevokePROD.bat PRODDTA PRODDTA CONNECT_STRING JDE_ROLE
call GrantRevokePROD.bat PRODCtrl PRODCtrl CONNECT_STRING JDE_ROLE
:
: System Common Schemas
call GrntRvokNONPROD.bat SY812 SY812 CONNECT_STRING JDE_ROLE JDE_NONPROD
call GrntRvokNONPROD.bat SVM812 SVM812 CONNECT_STRING JDE_ROLE JDE_NONPROD
call GrntRvokNONPROD.bat DD812 DD812 CONNECT_STRING JDE_ROLE JDE_NONPROD
call GrntRvokNONPROD.bat OL812 OL812 CONNECT_STRING JDE_ROLE JDE_NONPROD
:
: Prototype Schemas
call GrntRvokNONPROD.bat PY812 PY812 CONNECT_STRING JDE_ROLE JDE_NONPROD
call GrntRvokNONPROD.bat CRPDta CRPDta CONNECT_STRING JDE_ROLE JDE_NONPROD
call GrntRvokNONPROD.bat CRPCTL CRPCTL CONNECT_STRING JDE_ROLE JDE_NONPROD
:
: Development/Test Schemas
```

A second batch file called by the aforementioned batch file receives the passed parameters. This second batch file is named simply for the described process which will be invoked, e.g. **process_description.bat**. This batch file performs a sqlplus login and calls a sql file, e.g. **process_description.sql** passing arguments to said SQL file.

Figure 5: Example of `process_description.bat`



```

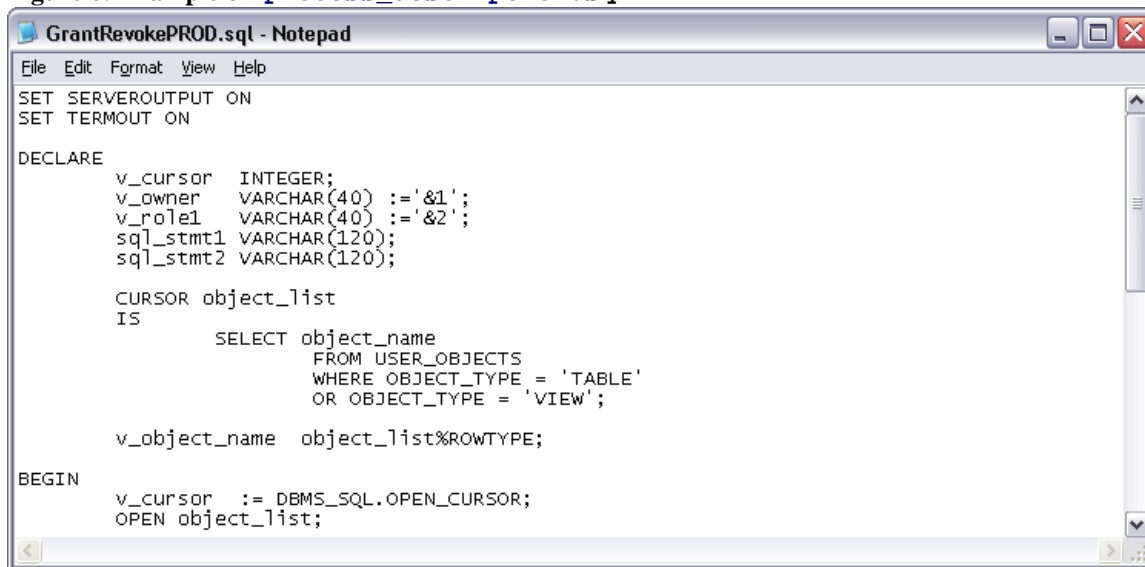
GrantRevokePROD.bat - Notepad
File Edit Format View Help
:/*
:*****
:PUBLIC Shutdown : Oracle E1 Platform Technologies 01/17/2007
:*****
:*/

sqlplus %1/%2@%3 @GrantRevokePROD.sql %1 %4

```

Finally the SQL file that is called will receive and prepare the arguments and query statements. It will then process the SQL query logic. In some cases the logic is quite simple. In other cases a cursor is loaded and opened so as to read in a single value, process that value, and continue to the next value until the cursor has fetched its last record.

Figure 6: Example of `process_description.sql`



```

GrantRevokePROD.sql - Notepad
File Edit Format View Help
SET SERVEROUTPUT ON
SET TERMOUT ON

DECLARE
    v_cursor    INTEGER;
    v_owner     VARCHAR(40) := '&1';
    v_role      VARCHAR(40) := '&2';
    sql_stmt1   VARCHAR(120);
    sql_stmt2   VARCHAR(120);

    CURSOR object_list
    IS
        SELECT object_name
          FROM USER_OBJECTS
         WHERE OBJECT_TYPE = 'TABLE'
            OR OBJECT_TYPE = 'VIEW';

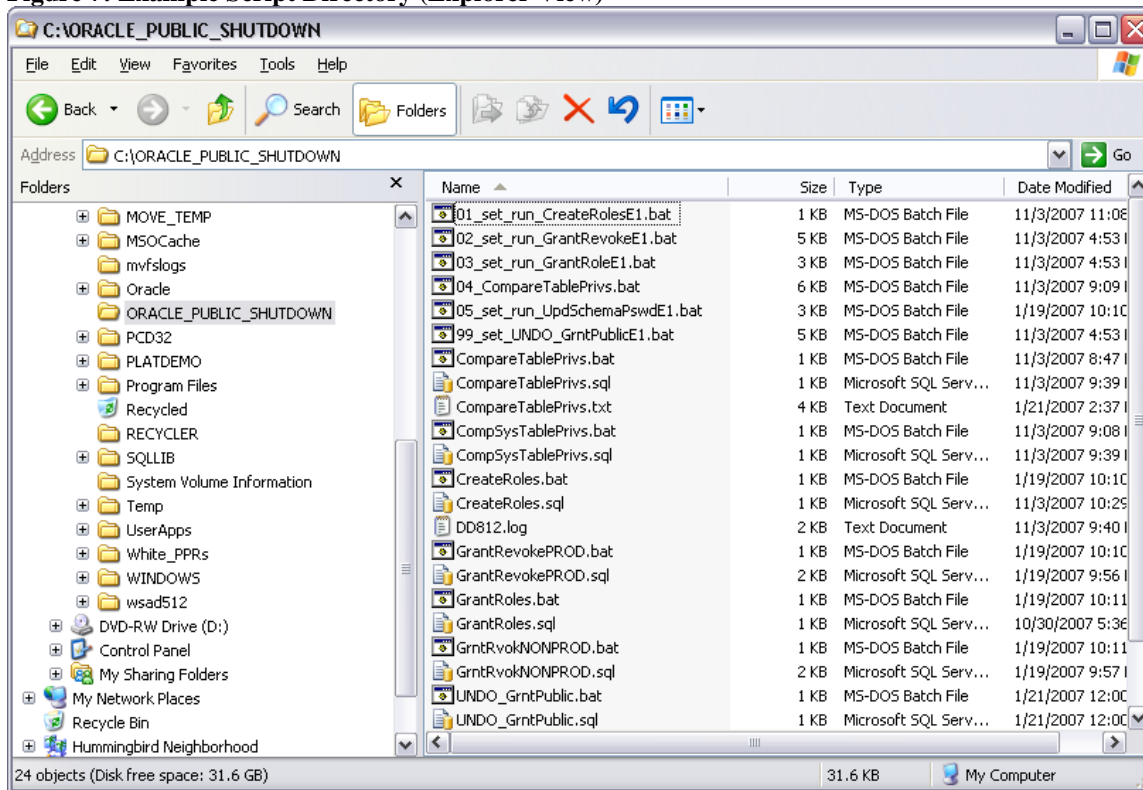
    v_object_name object_list%ROWTYPE;

BEGIN
    v_cursor := DBMS_SQL.OPEN_CURSOR;
    OPEN object_list;

```

Upon invoking each set of scripts the user will be able to monitor the progress and ensure that script was successfully executed. The folder ORACLE_PUBLIC_SHUTDOWN should be moved to a root level on any disk, e.g. C:\ORACLE_PUBLIC_SHUTDOWN.

Figure 7: Example Script Directory (Explorer View)



The scripts should be run from a command windows, e.g. Start | Run | cmd. Change directories to the location of the script folder, e.g. x:\ORACLE_PUBLIC_SHUTDOWN.

Figure 8: Running the Scripts from a Command Window

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\JDE>cd C:\ORACLE_PUBLIC_SHUTDOWN

C:\ORACLE_PUBLIC_SHUTDOWN>
```

Prior to implementing the Oracle Public Shutdown process one should check their existing table privileges via a summarized SQL query. This SQL query requires DBA privileges; consider using login account 'system'. A sample SQL query and result set is shown directly below. Our sample results show that only *public* privileges are associated with each set of EnterpriseOne schema owner tables.

Figure 9: Checking Baseline Table Privileges

Query:

```
SELECT COUNT(DISTINCT(TABLE_NAME)) TABLE_NO, OWNER, GRANTEE
FROM DBA_TAB_PRIVS WHERE OWNER NOT LIKE '%SYS%'
GROUP BY OWNER, GRANTEE ORDER BY 2;
```

Sample Result Set:

```
SQL> SELECT COUNT(DISTINCT(TABLE_NAME)) TABLE_NO, OWNER, GRANTEE
      2 FROM DBA_TAB_PRIVS WHERE OWNER NOT LIKE '%SYS%'
      3 GROUP BY OWNER, GRANTEE ORDER BY 2;
```

TABLE_NO	OWNER	GRANTEE
65	CRPCTL	PUBLIC
3333	CRPDTA	PUBLIC
7	DBSNMP	OEM_MONITOR
12	DD812	PUBLIC
48	DV812	PUBLIC
13	OL812	PUBLIC
5	ORDPLUGINS	PUBLIC
3	OUTLN	SELECT_CATALOG_ROLE
47	PD812	PUBLIC
65	PRODCTL	PUBLIC
3333	PRODDTA	PUBLIC
TABLE_NO	OWNER	GRANTEE
47	PS812	PUBLIC
77	PS812CTL	PUBLIC
3333	PS812DTA	PUBLIC
47	PY812	PUBLIC
7	SVM812	PUBLIC
294	SY812	PUBLIC
65	TESTCTL	PUBLIC
3333	TESTDTA	PUBLIC
187	XDB	PUBLIC
1	XDB	XDBADMIN

21 rows selected.

SQL>

Step One: Create Database Roles for EnterpriseOne

Our first step will create database roles. These roles will be associated with JDEdwards EnterpriseOne system users. It is possible to create multiple roles as a means of delineating database access by specified database role. For example, one could delineate access to production versus non-production based tables. One could also introduce a role for the purpose of controlling access to HRM and Payroll tables.

Table 2: Database Role Segmentation

Database Role	EnterpriseOne Segmentation	EnterpriseOne Proxy User
JDE_PROD	Production Environment	JDE, JDEAEC, JDECNC, JDECRM, JDEDST, JDEFIS, JDEFIN, JDEMFG, JDESCM, JDEUSR, etc.
JDE_NONPRO D	Non-Production Environment	JDE, JDECNC, TSTAEC, TSTCRM, TSTDST, TSTFIN, TSTHRM, TSTMFG, TSTSCM, TSTUSR
JDE_HRM	JDE HRM-Specific Tables	JDE, JDECNC, JDEHRM, JDEPAY
JDE_SPECIAL	JDE SPECIAL ACCESS	JDE, JDECNC, JDESPCL

Note: In all cases JDE (or PSFT) must maintain ALL access to ALL EnterpriseOne tables. This ensures that upgrades and/or updates are successful. For instance a table conversion will require complete rights to said table so as to ensure that the table is properly updated. As such, some customers will opt to disable database user JDE and enable it only for maintenance purposes, e.g. during Upgrades, Updates, Package Builds, etc.

Replacement of Substitution Variables

Our first step is to edit the enclosed 01_set_run_CreateRolesE1.bat file. The template batch file will contain substitution variables which must be replaced with valid values. In the template file shown below we will substitute the *password* (for user account system), *connect string*, and *role names* as needed for the EnterpriseOne Oracle database. Additionally, we have added entries for two generic roles for illustration purposes only.

Figure 10: Replacement of Substitution Variables (01_set_run_CreateRolesE1.bat)

Before:

```
...
:
: call <script>.bat <system_owner> <system_pswd> <connect_string> <role_name>
:
:*/
call CreateRoles.bat system MANAGER CONNECT_STRING JDE_ROLE
call CreateRoles.bat system MANAGER CONNECT_STRING JDE_NONPROD
```

After:

```
...
:*/
call CreateRoles.bat system oracle1 mlsan221 JDE_ROLE
call CreateRoles.bat system oracle1 mlsan221 JDE_NONPROD
call CreateRoles.bat system oracle1 mlsan221 JDE_<other>
call CreateRoles.bat system oracle1 mlsan221 JDE_<other2>
...
```

Once the substitution variables are updated the batch file is ready for use.

Under the Covers: 01_set_run_CreateRolesE1.bat

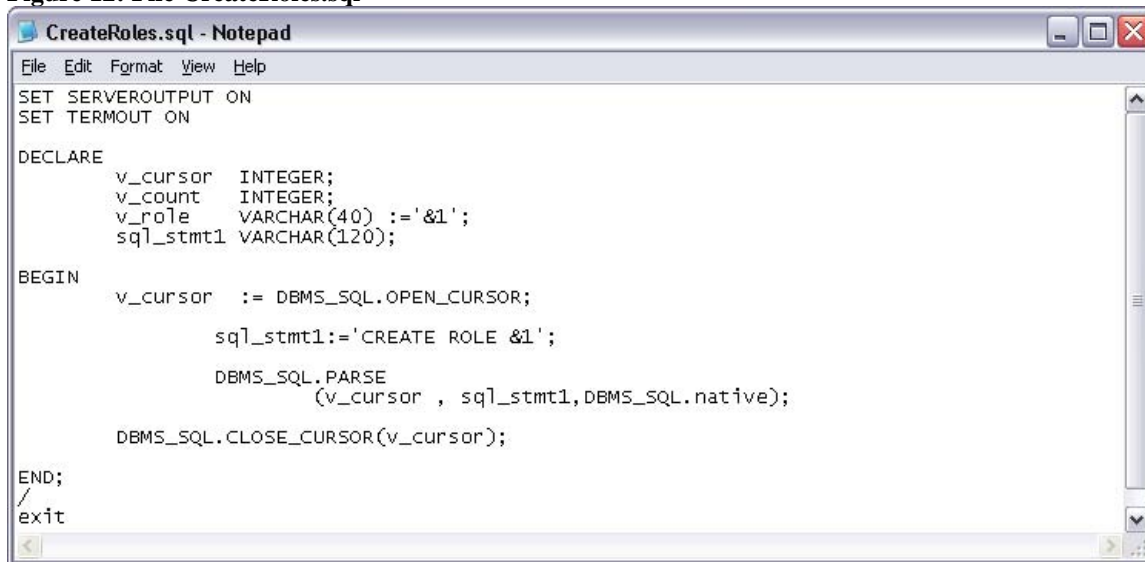
Under the covers the batch file 01_set_run_CreateRolesE1.bat will call CreateRoles.bat. This batch file will use a series of passed parameters to login to the EnterpriseOne Oracle database.

Figure 11: File CreateRoles.bat



The CreateRoles.bat file will in turn call the CreateRoles.sql file passing the database role name that is to be created.

Figure 12: File CreateRoles.sql



```
File Edit Format View Help
SET SERVEROUTPUT ON
SET TERMOUT ON

DECLARE
    v_cursor    INTEGER;
    v_count     INTEGER;
    v_role      VARCHAR(40) := '&1';
    sql_stmt1   VARCHAR(120);

BEGIN
    v_cursor := DBMS_SQL.OPEN_CURSOR;

    sql_stmt1:='CREATE ROLE &1';

    DBMS_SQL.PARSE
        (v_cursor , sql_stmt1,DBMS_SQL.native);

    DBMS_SQL.CLOSE_CURSOR(v_cursor);

END;
/
exit
```

Prior to running this procedure one can check their existing database roles using the following SQL statement: `SELECT * FROM DBA_ROLES ORDER BY 1;`. The SQL query requires DBA privileges; consider using login account 'system'. A sample SQL query and result set is shown directly below. This SQL query requires DBA privileges; consider using login account 'system'.

Figure 13: Checking Baseline Database Roles

Query:	
SELECT * FROM DBA_ROLES ORDER BY 1;	
Sample Result Set:	
SQL> SELECT * FROM DBA_ROLES ORDER BY 1;	
ROLE	PASSWORD
-----	-----
AQ_ADMINISTRATOR_ROLE	NO
AQ_USER_ROLE	NO
AUTHENTICATEDUSER	NO
CONNECT	NO
CTXAPP	NO
DBA	NO
DELETE_CATALOG_ROLE	NO
EJBCLIENT	NO
EXECUTE_CATALOG_ROLE	NO
EXP_FULL_DATABASE	NO
GATHER_SYSTEM_STATISTICS	NO
ROLE	PASSWORD
-----	-----
GLOBAL_AQ_USER_ROLE	GLOBAL
HS_ADMIN_ROLE	NO
IMP_FULL_DATABASE	NO
JAVADEBUGPRIV	NO
JAVAIDPRIV	NO
JAVASYSPRIV	NO
JAVAUSERPRIV	NO
JAVA_ADMIN	NO
JAVA_DEPLOY	NO

```

JDE_DUMMY                NO
LOGSTDBY_ADMINISTRATOR   NO

ROLE                      PASSWORD
-----
MGMT_USER                NO
OEM_ADVISOR              NO
OEM_MONITOR              NO
OLAP_DBA                 NO
OLAP_USER                NO
RECOVERY_CATALOG_OWNER   NO
RESOURCE                 NO
SCHEDULER_ADMIN          NO
SELECT_CATALOG_ROLE      NO
WM_ADMIN_ROLE            NO
XDBADMIN                 NO

ROLE                      PASSWORD
-----
XDBWEBSERVICES           NO

34 rows selected.

SQL>

```

Running 01_set_run_CreateRolesE1.bat file:

This script should be run from an open command window, e.g. Start | Run | cmd. Change the directory to the location of the script folder, e.g. x:\ORACLE_PUBLIC_SHUTDOWN. Command 01_set_run_CreateRolesE1.bat can be run from here. This procedure permits the user to validate success of the script.

Figure 14: Running 01_set_run_CreateRolesE1.bat

```

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\JDE>cd C:\ORACLE_PUBLIC_SHUTDOWN

C:\ORACLE_PUBLIC_SHUTDOWN>01_set_run_CreateRolesE1.bat

C:\ORACLE_PUBLIC_SHUTDOWN>call CreateRoles.bat system oracle1 mlsan221 J
DE_ROLE

C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus system/oracle1@mlsan221 @CreateRoles.sql JDE_R
OLE

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 15:30:16 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

old 4:          v_role    VARCHAR(40) :='&1';
new 4:          v_role    VARCHAR(40) :='JDE_ROLE';
old 10:         sql_stmt1:='CREATE ROLE &1';
new 10:         sql_stmt1:='CREATE ROLE JDE_ROLE';

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr
oduction
With the Partitioning, OLAP and Data Mining options

```

```

C:\ORACLE_PUBLIC_SHUTDOWN>call CreateRoles.bat system oracle1 mlsan221 J
DE_NONPROD

C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus system/oracle1@mlsan221 @CreateRoles.sql JDE_N
ONPROD

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 15:30:18 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

old 4:          v_role    VARCHAR(40) :='&1';
new 4:          v_role    VARCHAR(40) :='JDE_NONPROD';
old 10:         sql_stmt1:='CREATE ROLE &1';
new 10:         sql_stmt1:='CREATE ROLE JDE_NONPROD';

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr
oduction
With the Partitioning, OLAP and Data Mining options

C:\ORACLE_PUBLIC_SHUTDOWN>

```

To check the result one can again query table DBA_ROLES. Again, the SQL query requires DBA privileges; consider using login account 'system'. A sample SQL query and result set is shown directly below. In our example the WHERE clause uses a LIKE 'JDE%' query qualification.

Figure 15: Verifying Database Role Creation

```

Query:
SELECT * FROM DBA_ROLES WHERE ROLE LIKE 'JDE%' ORDER BY 1;

Sample Result Set:
SQL> SELECT * FROM DBA_ROLES WHERE ROLE LIKE 'JDE%' ORDER BY 1;

ROLE                                PASSWORD
-----
JDE_NONPROD                         NO
JDE_ROLE                           NO

SQL>

```

Step Two: Grant Table Access to 'Role' / Revoke Table Access from 'public'

Our next step will grant ALL privileges to defined EnterpriseOne database roles on specified EnterpriseOne database tables. With each table grant, the public role will be revoked from said table. This template script provides that the production users could access ALL EnterpriseOne tables while non-production users may access ONLY non-production tables (and the shared environment EnterpriseOne tables). This script can be further modified for your own specific business needs. **However, as before JDE (or PSFT) must maintain ALL access to ALL EnterpriseOne tables. This ensures that upgrades and/or updates are successful.**

Replacement of Substitution Variables

We begin by querying the EnterpriseOne Data Source Master table (F98611). With this information we can update the enclosed 02_set_run_GrantRevokeE1.bat file. The template batch file contains substitution variables which must be replaced with valid values. To properly update this batch file we must first ascertain valid EnterpriseOne schema owners found in Data Source Master.

Figure 16: Determining EnterpriseOne Database Schema Owners

Query:

```
SELECT OMDATP DTA_SRC, OMDATB CONNECT_STRNG, OMOOWN OWNER
FROM SY812.F98611 WHERE OMDSTP = 'O' ORDER BY 1;
```

Sample Result Set:

```
SQL> SELECT OMDATP DTA_SRC, OMDATB CONNECT_STRNG, OMOOWN OWNER
      2 FROM SY812.F98611 WHERE OMDSTP = 'O' ORDER BY 1;
```

DTA_SRC	CONNECT_STRNG	OWNER
Business Data - CRP	mlsan221	CRPDTA
Business Data - PROD	mlsan221	PRODDTA
Business Data - PS812	mlsan221	PS812DTA
Business Data - TEST	mlsan221	TESTDTA
Business Data - TEST - CIS	mlsan221	TESTDTA
Central Objects - DV812	mlsan221	DV812
Central Objects - PD812	mlsan221	PD812
Central Objects - PS812	mlsan221	PS812
Central Objects - PY812	mlsan221	PY812
Control Tables - CRP	mlsan221	CRPCTL
Control Tables - PS812	mlsan221	PS812CTL

DTA_SRC	CONNECT_STRNG	OWNER
Control Tables - Prod	mlsan221	PRODCTL
Control Tables - Test	mlsan221	TESTCTL
DENMLSAN221 - 812 Server Map	mlsan221	SVM812
Data Dictionary - 812	mlsan221	DD812
Data Dictionary - PS812	mlsan221	PS812CTL
Object Librarian - 812	mlsan221	OL812
System - 812	mlsan221	SY812
Versions - DV812	mlsan221	DV812
Versions - PD812	mlsan221	PD812
Versions - PS812	mlsan221	PS812
Versions - PY812	mlsan221	PY812

22 rows selected.

SQL>

In the template file shown below we will substitute a valid schema owner *account*, schema owner *password*, *connect string*, and *role name(s)*. These values are derived from the prior query run against the Data Source Master (F98611) table *and* from the database roles which were created during [Running 01_set_run_CreateRolesE1.bat](#).

Figure 17: Replacement of Substitution Variables (02_set_run_GrantRevokeE1.bat)

Excerpt of batch file:

```
...
: call <script>.bat <schema_owner> <schema_owner_pswd> <connect_string> <role>
:
: */
:
:   Production Schemas
call GrantRevokePROD.bat PD812 PD812 mlsan221 JDE_ROLE
call GrantRevokePROD.bat PRODDTA PRODDTA mlsan221 JDE_ROLE
call GrantRevokePROD.bat PRODCCTL PRODCCTL mlsan221 JDE_ROLE
:
:   System Common Schemas
call GrntRvokNONPROD.bat SY812 SY812 mlsan221 JDE_ROLE JDE_NONPROD
call GrntRvokNONPROD.bat SVM812 SVM812 mlsan221 JDE_ROLE JDE_NONPROD
call GrntRvokNONPROD.bat DD812 DD812 mlsan221 JDE_ROLE JDE_NONPROD
```

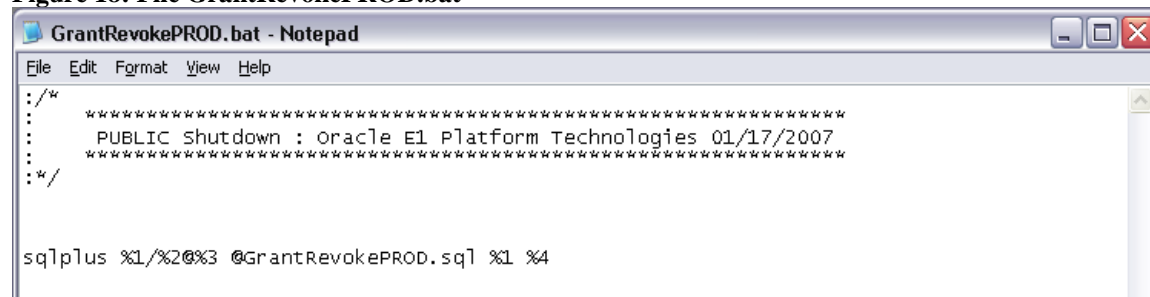
call GrntRvokNONPROD.bat	OL812	OL812	mlsan221	JDE_ROLE	JDE_NONPROD
:					
:	Prototype Schemas				
call GrntRvokNONPROD.bat	PY812	PY812	mlsan221	JDE_ROLE	JDE_NONPROD
call GrntRvokNONPROD.bat	CRPDTA	CRPDTA	mlsan221	JDE_ROLE	JDE_NONPROD
call GrntRvokNONPROD.bat	CRPCTL	CRPCTL	mlsan221	JDE_ROLE	JDE_NONPROD
:					
:	Development/Test Schemas				
call GrntRvokNONPROD.bat	DV812	DV812	mlsan221	JDE_ROLE	JDE_NONPROD
call GrntRvokNONPROD.bat	TESTDTA	TESTDTA	mlsan221	JDE_ROLE	JDE_NONPROD
call GrntRvokNONPROD.bat	TESTCTL	TESTCTL	mlsan221	JDE_ROLE	JDE_NONPROD
:					
:	Pristine Schemas				
call GrntRvokNONPROD.bat	PS812	PS812	mlsan221	JDE_ROLE	JDE_NONPROD
call GrntRvokNONPROD.bat	PS812DTA	PS812DTA	mlsan221	JDE_ROLE	JDE_NONPROD
call GrntRvokNONPROD.bat	PS812CTL	PS812CTL	mlsan221	JDE_ROLE	JDE_NONPROD

Once the substitution variables are updated the batch file is ready for use.

Under the Covers: 02_set_run_GrantRevokeE1.bat

Under the covers the batch file 02_set_run_GrantRevokeE1.bat will call one of two batch files, GrantRevokePROD.bat or GrntRvokNONPROD.bat. The only difference between the two files is the number of arguments passed to each, e.g. one database role versus two. These batch files in turn will use a series of passed parameters to login to the EnterpriseOne Oracle database and execute the PL/SQL Dynamic SQL logic.

Figure 18: File GrantRevokePROD.bat



```

GrantRevokePROD.bat - Notepad
File Edit Format View Help
:/*
:
: *****
: PUBLIC Shutdown : Oracle E1 Platform Technologies 01/17/2007
: *****
: */
:
sqlplus %1/%2@%3 @GrantRevokePROD.sql %1 %4

```

Figure 19: File GrntRvokNONPROD.bat



```

GrntRvokNONPROD.bat - Notepad
File Edit Format View Help
:/*
:
: *****
: PUBLIC Shutdown : Oracle E1 Platform Technologies 01/17/2007
: *****
: */
:
sqlplus %1/%2@%3 @GrntRvokNONPROD.sql %1 %4 %5

```

The GrantRevokePROD.bat and GrntRvokNONPROD.bat files will in turn call the GrantRevokePROD.sql and GrntRvokNONPROD.sql files respectively. These batch files will use a series of passed parameters to login to the EnterpriseOne Oracle database as a specific database schema owner. A portion of these parameters will be passed to the SQL file so the PL/SQL logic can construct dynamic SQL and perform the logic.

For this step, a cursor is described and opened listing all of the objects found in the schema that was used for login. Two additional SQL queries are constructed by the SQL file. The first grants ALL privileges to the roles

passed as parameters on each table (object) listed in the cursor. The second revokes the *public* role from the same table. This logic is repeated for each table (object) listed in the cursor, one by one until the cursor list is exhausted.

Figure 20: File GrntRvokNONPROD.sql

```
SET SERVEROUTPUT ON
SET TERMOUT ON

DECLARE
    v_cursor    INTEGER;
    v_owner     VARCHAR(40) := '&1';
    v_role1     VARCHAR(40) := '&2';
    v_role2     VARCHAR(40) := '&3';
    sql_stmt1   VARCHAR(120);
    sql_stmt2   VARCHAR(120);

    CURSOR object_list
    IS
        SELECT object_name
           FROM USER_OBJECTS
          WHERE OBJECT_TYPE = 'TABLE'
             OR OBJECT_TYPE = 'VIEW';

    v_object_name object_list%ROWTYPE;
BEGIN
    v_cursor := DBMS_SQL.OPEN_CURSOR;
    OPEN object_list;

    FETCH object_list INTO v_object_name;

    WHILE object_list%FOUND
    LOOP

        sql_stmt1:='GRANT ALL PRIVILEGES ON ' || v_owner || '.' ||
            v_object_name.object_name || ' TO &2, &3' ;

        sql_stmt2:='REVOKE ALL PRIVILEGES ON ' || v_owner || '.' ||
            v_object_name.object_name || ' FROM PUBLIC';

        DBMS_SQL.PARSE
            (v_cursor , sql_stmt1,DBMS_SQL.V7);

        DBMS_SQL.PARSE
            (v_cursor , sql_stmt2,DBMS_SQL.V7);

        FETCH object_list INTO v_object_name;
    END LOOP;

    CLOSE object_list;
END;
/
exit
```

Prior to running this procedure one can check their existing database roles using the following SQL statement: `SELECT * FROM DBA_ROLES ORDER BY 1;` The SQL query requires DBA privileges; consider using login account 'system'. A sample SQL query and result set is shown directly below.

Figure 21: Listing Database Roles

Query:

```
SELECT * FROM DBA_ROLES ORDER BY 1;
```

Sample Result Set:

```
SQL> SELECT * FROM DBA_ROLES ORDER BY 1;
```

ROLE	PASSWORD
-----	-----
AQ_ADMINISTRATOR_ROLE	NO
AQ_USER_ROLE	NO
AUTHENTICATEDUSER	NO
CONNECT	NO
CTXAPP	NO
DBA	NO
DELETE_CATALOG_ROLE	NO
EJBCLIENT	NO
EXECUTE_CATALOG_ROLE	NO
EXP_FULL_DATABASE	NO
GATHER_SYSTEM_STATISTICS	NO
ROLE	PASSWORD
-----	-----
GLOBAL_AQ_USER_ROLE	GLOBAL
HS_ADMIN_ROLE	NO
IMP_FULL_DATABASE	NO
JAVADEBUGPRIV	NO
JAVAIDPRIV	NO
JAVASYSPRIV	NO
JAVAUSERPRIV	NO
JAVA_ADMIN	NO
JAVA_DEPLOY	NO
JDE_DUMMY	NO
LOGSTDBY_ADMINISTRATOR	NO
ROLE	PASSWORD
-----	-----
MGMT_USER	NO
OEM_ADVISOR	NO
OEM_MONITOR	NO
OLAP_DBA	NO
OLAP_USER	NO
RECOVERY_CATALOG_OWNER	NO
RESOURCE	NO
SCHEDULER_ADMIN	NO
SELECT_CATALOG_ROLE	NO
WM_ADMIN_ROLE	NO
XDBADMIN	NO
ROLE	PASSWORD
-----	-----
XDBWEBSERVICES	NO

34 rows selected.

```
SQL>
```

Running 02_set_run_GrantRevokeE1.bat file:

See [Appendix B](#) for information about clearing the DBA_RECYCLEBIN prior to running this script. Failure to do this may result in portions of the script failing for particular schema owners.

This script should be run from an open command window, e.g. Start | Run | cmd. Change the directory to the location of the script folder, e.g. x:\ORACLE_PUBLIC_SHUTDOWN. Command 02_set_run_GrantRevokeE1.bat can be run from here. This procedure permits the user to validate success of the script.

Figure 22: Running 02_set_run_GrantRevokeE1.bat

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\JDE>cd C:\ORACLE_PUBLIC_SHUTDOWN

C:\ORACLE_PUBLIC_SHUTDOWN>02_set_run_GrantRevokeE1.bat

C:\ORACLE_PUBLIC_SHUTDOWN>call GrantRevokePROD.bat PD812 PD812 mlsan
221 JDE_ROLE

C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus PD812/PD812@mlsan221 @GrantRevokePROD.sql PD81
2 JDE_ROLE

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 16:14:20 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

old 3:          v_owner    VARCHAR(40) :='&1';
new 3:          v_owner    VARCHAR(40) :='PD812';
old 4:          v_role1    VARCHAR(40) :='&2';
new 4:          v_role1    VARCHAR(40) :='JDE_ROLE';
old 28:         v_object_name.object_name || ' TO &2' ;
new 28:         v_object_name.object_name || ' TO JDE_ROLE' ;

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr
oduction
With the Partitioning, OLAP and Data Mining options

C:\ORACLE_PUBLIC_SHUTDOWN>call GrantRevokePROD.bat PRODDTA PRODDTA mlsan
221 JDE_ROLE

C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus PRODDTA/PRODDTA@mlsan221 @GrantRevokePROD.sql
PRODDTA JDE_ROLE

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 16:14:24 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

old 3:          v_owner    VARCHAR(40) :='&1';
new 3:          v_owner    VARCHAR(40) :='PRODDTA';
old 4:          v_role1    VARCHAR(40) :='&2';
new 4:          v_role1    VARCHAR(40) :='JDE_ROLE';
old 28:         v_object_name.object_name || ' TO &2' ;
new 28:         v_object_name.object_name || ' TO JDE_ROLE' ;

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr
oduction
```

With the Partitioning, OLAP and Data Mining options

... <repeated as iterative call for each schema owner>

C:\ORACLE_PUBLIC_SHUTDOWN>

The entire result set of command 02_set_run_GrantRevokeE1.bat can be found in [Appendix A](#).

Step Three: Grant Role to EnterpriseOne Proxy Users

Our next step will grant the previously created database role(s) to defined EnterpriseOne system (proxy) users. This association will allow the EnterpriseOne system (proxy) user associated with a particular database role access to the database tables available to said role. Again, this script can be further modified for your own specific business needs. **As before database user JDE (or PSFT) must maintain ALL access to ALL EnterpriseOne tables.**

Replacement of Substitution Variables

Our first step is to edit the enclosed 03_set_run_GrantRoleE1.bat file. The template batch file will contain substitution variables which must be replaced with valid values. In the template file shown below we will substitute the *password* (for user account system), *connect string*, database *user*, and *role names* as needed for access to the EnterpriseOne Oracle database.

We begin by querying the EnterpriseOne OneWorld Security Master table (F98OWSEC). With this information we can update the enclosed 03_set_run_GrantRoleE1.bat file. The template batch file contains substitution variables which must be replaced with valid values. To properly update this batch file we must first ascertain a list of valid EnterpriseOne system (proxy) users.

Figure 23: Listing Valid EnterpriseOne System (Proxy) Users

Query:

```
SELECT DISTINCT(T0.SCSECUSR)
FROM SY812.F98OWSEC T0, SY812.F98OWPU T1
WHERE T0.SCSECUSR = T1.PUSECUSR;
```

Sample Result Set:

```
SQL> SELECT DISTINCT(T0.SCSECUSR)
      2 FROM SY812.F98OWSEC T0, SY812.F98OWPU T1
      3 WHERE T0.SCSECUSR = T1.PUSECUSR;
```

```
SCSECUSR
-----
JDE
JDEUSR1
JDEUSR2
JDEUSR3
JDEUSR4
KJUDSON
KPOND
MOCONNER
...
SQL>
```

Substitute the users found in the template 03_set_run_GrantRoleE1.bat file with the list obtained from the previous query (above). Associate the database role with the appropriate user account. For instance, production users are associated with role JDE_ROLE where development users are associated with role JDE_NONPROD.

Figure 24: Replacement of Substitution Variables (03_set_run_GrantRoleE1.bat)

```
Before:
...
:
: call <script>.bat <system_owner> <system_pswd> <connect_string> <user> <role>
:
:*/
: Production User Accounts
call GrantRoles.bat system oracle1 mlsan221 JDE JDE_ROLE
call GrantRoles.bat system oracle1 mlsan221 JDEAEC JDE_ROLE
call GrantRoles.bat system oracle1 mlsan221 JDECNC JDE_ROLE
call GrantRoles.bat system oracle1 mlsan221 JDECRM JDE_ROLE
call GrantRoles.bat system oracle1 mlsan221 JDEDST JDE_ROLE
call GrantRoles.bat system oracle1 mlsan221 JDEFIN JDE_ROLE
call GrantRoles.bat system oracle1 mlsan221 JDEFIS JDE_ROLE
call GrantRoles.bat system oracle1 mlsan221 JDEHRM JDE_ROLE
call GrantRoles.bat system oracle1 mlsan221 JDEMFG JDE_ROLE
call GrantRoles.bat system oracle1 mlsan221 JDESCM JDE_ROLE
:
: Non-production User Accounts (development)
call GrantRoles.bat system oracle1 mlsan221 DEVAEC JDE_NONPROD
call GrantRoles.bat system oracle1 mlsan221 DEVCRM JDE_NONPROD
call GrantRoles.bat system oracle1 mlsan221 DEV DST JDE_NONPROD
call GrantRoles.bat system oracle1 mlsan221 DEVFIN JDE_NONPROD
call GrantRoles.bat system oracle1 mlsan221 DEVFIS JDE_NONPROD
call GrantRoles.bat system oracle1 mlsan221 DEVHRM JDE_NONPROD
call GrantRoles.bat system oracle1 mlsan221 DEV MFG JDE_NONPROD
call GrantRoles.bat system oracle1 mlsan221 DEVSCM JDE_NONPROD
...

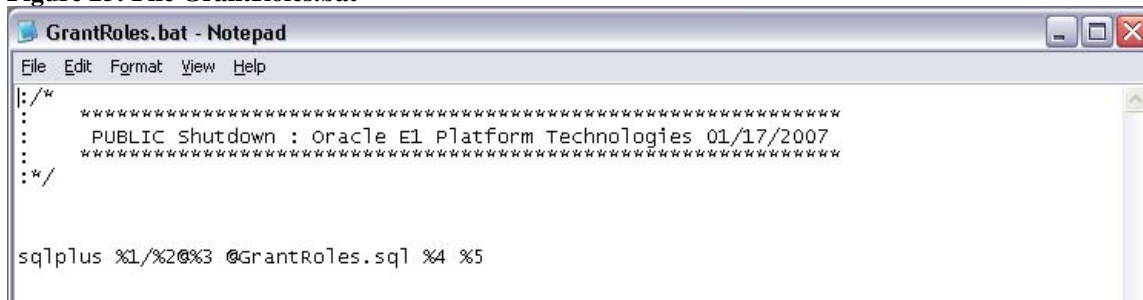
After:
...
:
: call <script>.bat <system_owner> <system_pswd> <connect_string> <user> <role>
:
:*/
: Production User Accounts
call GrantRoles.bat system oracle1 mlsan221 JDE JDE_ROLE
call GrantRoles.bat system oracle1 mlsan221 JDEUSR1 JDE_ROLE
call GrantRoles.bat system oracle1 mlsan221 JDEUSR2 JDE_ROLE
call GrantRoles.bat system oracle1 mlsan221 JDEUSR3 JDE_ROLE
call GrantRoles.bat system oracle1 mlsan221 JDEUSR4 JDE_ROLE
call GrantRoles.bat system oracle1 mlsan221 KJUDSON JDE_ROLE
call GrantRoles.bat system oracle1 mlsan221 KPOND JDE_ROLE
call GrantRoles.bat system oracle1 mlsan221 MOCONNER JDE_ROLE
...
```

Once the substitution variables are updated the batch file is ready for use.

Under the Covers: 03_set_run_GrantRoleE1.bat

Under the covers the batch file 03_set_run_GrantRoleE1.bat will call GrantRoles.bat. This batch file will use a series of passed parameters to login to the EnterpriseOne Oracle database.

Figure 25: File GrantRoles.bat

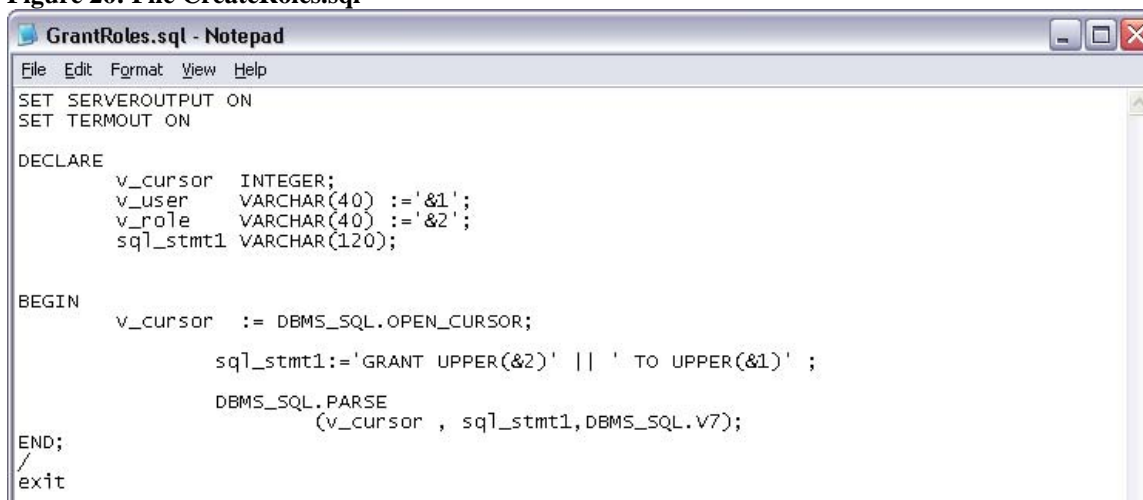


```
/*
PUBLIC Shutdown : Oracle E1 Platform Technologies 01/17/2007
*/

sqlplus %1/%2@%3 @GrantRoles.sql %4 %5
```

The GrantRoles.bat file will in turn call the GrantRoles.sql file passing the user and the database role name to be granted to the defined user.

Figure 26: File CreateRoles.sql



```
SET SERVEROUTPUT ON
SET TERMOUT ON

DECLARE
    v_cursor    INTEGER;
    v_user       VARCHAR(40) := '&1';
    v_role       VARCHAR(40) := '&2';
    sql_stmt1    VARCHAR(120);

BEGIN
    v_cursor := DBMS_SQL.OPEN_CURSOR;

    sql_stmt1 := 'GRANT UPPER(&2) || ' TO UPPER(&1)';

    DBMS_SQL.PARSE
        (v_cursor , sql_stmt1,DBMS_SQL.V7);

END;
/
exit
```

Prior to running this procedure one can check their existing database roles to user relationship using the following SQL statement: `SELECT GRANTEE, GRANTED_ROLE FROM DBA_ROLE_PRIVS WHERE GRANTED_ROLE LIKE 'JDE%' ORDER BY 2, 1;`. The SQL query requires DBA privileges; consider using login account 'system'. A sample SQL query and result set is shown directly below.

Figure 27: Checking Baseline Database Roles

Query:

```
SELECT GRANTEE, GRANTED_ROLE FROM DBA_ROLE_PRIVS
WHERE GRANTED_ROLE LIKE 'JDE%' ORDER BY 2, 1;
```

Sample Result Set:

```
SQL> SELECT GRANTEE, GRANTED_ROLE FROM DBA_ROLE_PRIVS
      2  WHERE GRANTED_ROLE LIKE 'JDE%' ORDER BY 2, 1;
```

GRANTEE	GRANTED_ROLE
SYSTEM	JDE_NONPROD
SYSTEM	JDE_ROLE

SQL>

Running 03_set_run_GrantRoleE1.bat file:

This script should be run from an open command window, e.g. Start | Run | cmd. Change the directory to the location of the script folder, e.g. x:\ORACLE_PUBLIC_SHUTDOWN. Command 03_set_run_GrantRoleE1.bat can be run from here. This procedure permits the user to validate success of the script.

Figure 28: Running 01_set_run_CreateRolesE1.bat

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\JDE>cd C:\ORACLE_PUBLIC_SHUTDOWN

C:\ORACLE_PUBLIC_SHUTDOWN>03_set_run_GrantRoleE1.bat

C:\ORACLE_PUBLIC_SHUTDOWN>call GrantRoles.bat system oracle1 mlsan221 JD
E JDE_ROLE

C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus system/oracle1@mlsan221 @GrantRoles.sql JDE JD
E_ROLE

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 20:22:08 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

old 3:          v_user      VARCHAR(40) := '&1';
new 3:          v_user      VARCHAR(40) := 'JDE';
old 4:          v_role      VARCHAR(40) := '&2';
new 4:          v_role      VARCHAR(40) := 'JDE_ROLE';
old 11:         sql_stmt1:= 'GRANT &2 TO &1' ;
new 11:         sql_stmt1:= 'GRANT JDE_ROLE TO JDE' ;

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr
oduction
With the Partitioning, OLAP and Data Mining options

... <runs multiple times for each database user and database role combination>

C:\ORACLE_PUBLIC_SHUTDOWN>
```

To check the result one can again query table DBA_ROLES. Again, the SQL query requires DBA privileges; consider using login account 'system'. A sample SQL query and result set is shown directly below. In our example the WHERE clause uses a LIKE 'JDE%' query qualification.

Figure 29: Verifying Database Role Creation

```
Query:
SELECT GRANTEE, GRANTED_ROLE FROM DBA_ROLE_PRIVS
WHERE GRANTED_ROLE LIKE 'JDE%' ORDER BY 2, 1;

Sample Result Set:
SQL> SELECT GRANTEE, GRANTED_ROLE FROM DBA_ROLE_PRIVS
2 WHERE GRANTED_ROLE LIKE 'JDE%' ORDER BY 2, 1;

GRANTEE                                GRANTED_ROLE
-----                                -
JDEUSR4                                JDE_NONPROD
JDEUSR5                                JDE_NONPROD
```

```

SYSTEM                JDE_NONPROD
JDE                    JDE_ROLE
JDEUSR1                JDE_ROLE
JDEUSR2                JDE_ROLE
JDEUSR3                JDE_ROLE
KJUDSON                JDE_ROLE
KPOND                  JDE_ROLE
MOCONNER               JDE_ROLE
SYSTEM                 JDE_ROLE
...
SQL>

```

Step Four: Evaluate & Compare Table Privileges by Role and ‘public’

Our next step will permit you to evaluate privileges found on EnterpriseOne database tables. Prior to starting we ran a query to obtain EnterpriseOne [baseline table privileges](#). If we ran the same query at this time we would find the schema owner privileges have changed whereby a reference to *public* no longer exists. Additionally the database tables would now have the database role association we designated when running script 02_set_run_GrantRevokeE1.bat.

Figure 30: Checking Table Privileges

Query:

```

SELECT COUNT(DISTINCT(TABLE_NAME)) TABLE_NO, OWNER, GRANTEE
FROM DBA_TAB_PRIVS WHERE OWNER NOT LIKE '%SYS%'
GROUP BY OWNER, GRANTEE ORDER BY 2;

```

Sample Result Set:

```

SQL> SELECT COUNT(DISTINCT(TABLE_NAME)) TABLE_NO, OWNER, GRANTEE
2   FROM DBA_TAB_PRIVS WHERE OWNER NOT LIKE '%SYS%'
3   GROUP BY OWNER, GRANTEE ORDER BY 2;

```

TABLE_NO	OWNER	GRANTEE
65	CRPCTL	JDE_NONPROD
65	CRPCTL	JDE_ROLE
3333	CRPDTA	JDE_NONPROD
3333	CRPDTA	JDE_ROLE
7	DBSNMP	OEM_MONITOR
12	DD812	JDE_NONPROD
12	DD812	JDE_ROLE
47	DV812	JDE_NONPROD
47	DV812	JDE_ROLE
13	OL812	JDE_NONPROD
13	OL812	JDE_ROLE

TABLE_NO	OWNER	GRANTEE
5	ORDPLUGINS	PUBLIC
3	OUTLN	SELECT_CATALOG_ROLE
47	PD812	JDE_ROLE
65	PRODCCTL	JDE_ROLE
3333	PRODDTA	JDE_ROLE
47	PS812	JDE_NONPROD
47	PS812	JDE_ROLE
77	PS812CTL	JDE_NONPROD
77	PS812CTL	JDE_ROLE
3333	PS812DTA	JDE_NONPROD
3333	PS812DTA	JDE_ROLE

TABLE_NO	OWNER	GRANTEE

47	PY812	JDE_NONPROD
47	PY812	JDE_ROLE
7	SVM812	JDE_NONPROD
7	SVM812	JDE_ROLE
295	SY812	JDE_NONPROD
295	SY812	JDE_ROLE
65	TESTCTL	JDE_NONPROD
65	TESTCTL	JDE_ROLE
3333	TESTDTA	JDE_NONPROD
3333	TESTDTA	JDE_ROLE
187	XDB	PUBLIC
TABLE_NO OWNER		GRANTEE
-----		-----
1	XDB	XDBADMIN

34 rows selected.

SQL>

Script 04_set_run_CompareTablePrivs.bat has been created to yield similar results for ALL schema owners and for individual schema owners as defined by the script.

Replacement of Substitution Variables

We begin by querying the EnterpriseOne Data Source Master table (F98611). With this information we can update the enclosed 04_set_run_CompareTablePrivs.bat file. The template batch file contains substitution variables which must be replaced with valid values. To properly update this batch file we must first ascertain valid EnterpriseOne schema owners found in Data Source Master.

Figure 31: Determining EnterpriseOne Database Schema Owners

Query:		
SELECT OMDATP DTA_SRC, OMDATB CONNECT_STRNG, OMOOWN OWNER		
FROM SY812.F98611 WHERE OMDSTP = 'O' ORDER BY 1;		
Sample Result Set:		
SQL> SELECT OMDATP DTA_SRC, OMDATB CONNECT_STRNG, OMOOWN OWNER		
2 FROM SY812.F98611 WHERE OMDSTP = 'O' ORDER BY 1;		
DTA_SRC	CONNECT_STRNG	OWNER
-----	-----	-----
Business Data - CRP	mlsan221	CRPDTA
Business Data - PROD	mlsan221	PRODDTA
Business Data - PS812	mlsan221	PS812DTA
Business Data - TEST	mlsan221	TESTDTA
Business Data - TEST - CIS	mlsan221	TESTDTA
Central Objects - DV812	mlsan221	DV812
Central Objects - PD812	mlsan221	PD812
Central Objects - PS812	mlsan221	PS812
Central Objects - PY812	mlsan221	PY812
Control Tables - CRP	mlsan221	CRPCTL
Control Tables - PS812	mlsan221	PS812CTL
DTA_SRC	CONNECT_STRNG	OWNER
-----	-----	-----
Control Tables - Prod	mlsan221	PRODCTL
Control Tables - Test	mlsan221	TESTCTL
DENMLSAN221 - 812 Server Map	mlsan221	SVM812
Data Dictionary - 812	mlsan221	DD812
Data Dictionary - PS812	mlsan221	PS812CTL
Object Librarian - 812	mlsan221	OL812

```

System - 812          mlsan221          SY812
Versions - DV812      mlsan221          DV812
Versions - PD812      mlsan221          PD812
Versions - PS812      mlsan221          PS812
Versions - PY812      mlsan221          PY812

22 rows selected.

SQL>

```

In the template file shown below we will substitute a valid schema owner *account*, schema owner *password*, *connect string*, *role name(s)*, and *log file name*. These values are derived from the prior query run against the Data Source Master (F98611) table *and* from the database roles which were created during [Running 01_set_run_CreateRolesE1.bat](#).

Figure 32: Replacement of Substitution Variables (04_set_run_CompareTablePrivs.bat)

Excerpt of batch file:

```

...
:
:call <script>.bat <schema_owner> <schema_own_pswd> <connect_string> <PUBLIC> <role2> <role3> <log_file>
:
:*/
:
call CompSysTablePrivs.bat system oracle1 mlsan221 PUBLIC JDE_ROLE JDE_NONPROD SYSTEM.log
:
: Production Schemas
call CompareTablePrivs.bat PD812 PD812 mlsan221 PUBLIC JDE_ROLE BLANK_ROLE PD812.log
call CompareTablePrivs.bat PRODDTA PRODDTA mlsan221 PUBLIC JDE_ROLE BLANK_ROLE PRODDTA.log
call CompareTablePrivs.bat PRODCtl PRODCtl mlsan221 PUBLIC JDE_ROLE BLANK_ROLE PRODCtl.log
:
: System Common Schemas
call CompareTablePrivs.bat SY812 SY812 mlsan221 PUBLIC JDE_ROLE JDE_NONPROD SY812.log
call CompareTablePrivs.bat SVM812 SVM812 mlsan221 PUBLIC JDE_ROLE JDE_NONPROD SVM812.log
call CompareTablePrivs.bat DD812 DD812 mlsan221 PUBLIC JDE_ROLE JDE_NONPROD DD812.log
call CompareTablePrivs.bat OL812 OL812 mlsan221 PUBLIC JDE_ROLE JDE_NONPROD OL812.log
:
: Prototype Schemas
call CompareTablePrivs.bat PY812 PY812 mlsan221 PUBLIC JDE_ROLE JDE_NONPROD PY812.log
call CompareTablePrivs.bat CRPDTA CRPDTA mlsan221 PUBLIC JDE_ROLE JDE_NONPROD CRPDTA.log
call CompareTablePrivs.bat CRPCTL CRPCTL mlsan221 PUBLIC JDE_ROLE JDE_NONPROD CRPCTL.log
:
: Development/Test Schemas
call CompareTablePrivs.bat DV812 DV812 mlsan221 PUBLIC JDE_ROLE JDE_NONPROD DDV812.log
call CompareTablePrivs.bat TESTDTA TESTDTA mlsan221 PUBLIC JDE_ROLE JDE_NONPROD TESTDTA.log
call CompareTablePrivs.bat TESTCTL TESTCTL mlsan221 PUBLIC JDE_ROLE JDE_NONPROD TESTCTL.log
:
: Pristine Schemas
call CompareTablePrivs.bat PS812 PS812 mlsan221 PUBLIC JDE_ROLE JDE_NONPROD PS812.log
call CompareTablePrivs.bat PS812DTA PS812DTA mlsan221 PUBLIC JDE_ROLE JDE_NONPROD PS812DTA.log
call CompareTablePrivs.bat PS812CTL PS812CTL mlsan221 PUBLIC JDE_ROLE JDE_NONPROD PS812CTL.log

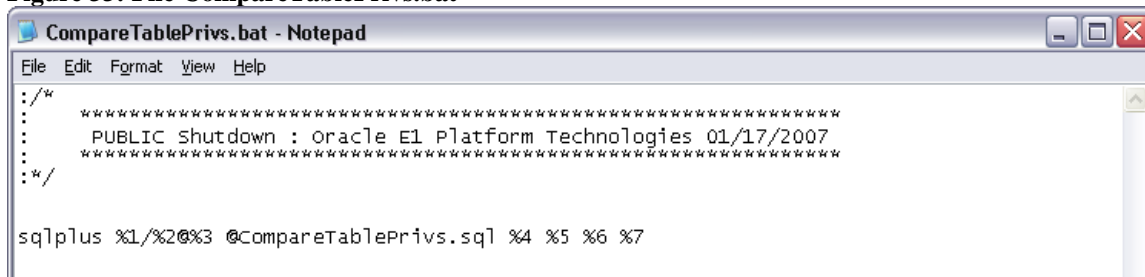
```

Once the substitution variables are updated the batch file is ready for use.

Under the Covers: 04_set_run_CompareTablePrivs.bat

Under the covers the batch file 04_set_run_CompareTablePrivs.bat will call one of two batch files, CompSysTablePrivs.bat or CompareTablePrivs.bat. The only difference between the two files is the use of DBA versus USER table lookups for privileges. These batch files in turn will use a series of passed parameters to login to the EnterpriseOne Oracle database and execute the Dynamic SQL logic.

Figure 33: File CompareTablePrivs.bat

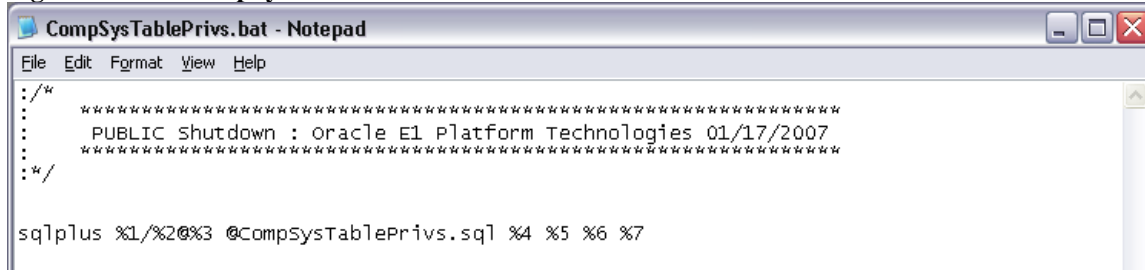


```
CompareTablePrivs.bat - Notepad
File Edit Format View Help

:/*
*****
PUBLIC Shutdown : Oracle E1 Platform Technologies 01/17/2007
*****
*/

sqlplus %1/%2@%3 @CompareTablePrivs.sql %4 %5 %6 %7
```

Figure 34: File CompSysTablePrivs.bat



```
CompSysTablePrivs.bat - Notepad
File Edit Format View Help

:/*
*****
PUBLIC Shutdown : Oracle E1 Platform Technologies 01/17/2007
*****
*/

sqlplus %1/%2@%3 @CompSysTablePrivs.sql %4 %5 %6 %7
```

The CompSysTablePrivs.bat and CompareTablePrivs.bat files will in turn call the CompSysTablePrivs.sql and CompareTablePrivs.sql files respectively. These batch files will use a series of passed parameters to login to the EnterpriseOne Oracle database as a specific database schema owner. A portion of these parameters will be passed to the SQL file to construct dynamic SQL queries.

For this step, a query will output its results to a log file for reference.

Figure 35: File CompSysTablePrivs.sql

```
set head off
set long 9000
set pages 0
set linesize 132
set colsep " "
set pagesize 0
set newpage none
set feedback off
set termout off
set recsep off
set heading off
set echo off

DEFINE v_Role=&1
DEFINE v_Role1=&2
DEFINE v_Role2=&3
DEFINE LogFile=&4
spool &LogFile
SELECT '-----' from dual;
SELECT 'COUNT(DISTINCT(TABLE_NAME)) OWNER GRANTEE' from dual;
SELECT '-----' from dual;
SELECT COUNT(DISTINCT(TABLE_NAME)), ' ', OWNER, ' ', GRANTEE FROM DBA_TAB_PRIVS WHERE GRANTEE = '&&v_Role' OR
GRANTEE = '&&v_Role1' OR GRANTEE = '&&v_Role2' GROUP BY OWNER, GRANTEE ORDER BY 2;
spool off;
exit
```

Running 04_set_run_CompareTablePrivs.bat:

This script should be run from an open command window, e.g. Start | Run | cmd. Change the directory to the location of the script folder, e.g. x:\ORACLE_PUBLIC_SHUTDOWN. Command 04_set_run_CompareTablePrivs.bat can be run from here. This procedure permits the user to validate success of the script.

Figure 36: Running 04_set_run_CompareTablePrivs.bat

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\JDE>cd C:\ORACLE_PUBLIC_SHUTDOWN

C:\ORACLE_PUBLIC_SHUTDOWN>04_set_run_CompareTablePrivs.bat

C:\ORACLE_PUBLIC_SHUTDOWN>call CompSysTablePrivs.bat system oracle1 mls
an221 PUBLIC JDE_ROLE JDE_NONPROD SYSTEM.log

C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus system/oracle1@mlsan221 @CompSysTablePrivs.sql
PUBLIC JDE_ROLE JDE_NONPROD SYSTEM.log

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 21:35:38 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr
oduction
With the Partitioning, OLAP and Data Mining options

... <runs multiple times for each call statement>

C:\ORACLE_PUBLIC_SHUTDOWN>
```

The output for each call statement will be written to a log file. The log file will be created in the scripts folder C:\ORACLE_PUBLIC_SHUTDOWN.

Figure 37: Sample SYSTEM.log File

```
-----
COUNT(DISTINCT(TABLE_NAME)) OWNER GRANTEE
-----
old 1: SELECT COUNT(DISTINCT(TABLE_NAME)), ' ', OWNER, ' ', GRANTEE FROM DBA_TAB_PRIVS WHERE
GRANTEE = '&&v_Role' OR GRANTEE = '&&v_Role1' OR GRANTEE = '&&v_Role2' GROUP BY OWNER, GRANTEE
ORDER BY 2
new 1: SELECT COUNT(DISTINCT(TABLE_NAME)), ' ', OWNER, ' ', GRANTEE FROM DBA_TAB_PRIVS WHERE
GRANTEE = 'PUBLIC' OR GRANTEE = 'JDE_ROLE' OR GRANTEE = 'JDE_NONPROD' GROUP BY OWNER, GRANTEE
ORDER BY 2

17599 SYS PUBLIC
187 XDB PUBLIC
12 DD812 JDE_ROLE
12 DD812 JDE_NONPROD
114 DMSYS PUBLIC
47 DV812 JDE_ROLE
47 DV812 JDE_NONPROD
651 MDSYS PUBLIC
13 OL812 JDE_ROLE
13 OL812 JDE_NONPROD
47 PD812 JDE_ROLE
47 PS812 JDE_ROLE
47 PS812 JDE_NONPROD
47 PY812 JDE_ROLE
47 PY812 JDE_NONPROD
```

295	SY812	JDE_ROLE
295	SY812	JDE_NONPROD
100	WMSYS	PUBLIC
65	CRPCTL	JDE_ROLE
65	CRPCTL	JDE_NONPROD
3333	CRPDTA	JDE_ROLE
3333	CRPDTA	JDE_NONPROD
82	CTXSYS	PUBLIC
72	EXFSYS	PUBLIC
1681	ORDSYS	PUBLIC
7	SVM812	JDE_ROLE
7	SVM812	JDE_NONPROD
9	SYSTEM	PUBLIC
239	OLAPSYS	PUBLIC
65	PRODCTL	JDE_ROLE
3333	PRODDTA	JDE_ROLE
65	TESTCTL	JDE_ROLE
65	TESTCTL	JDE_NONPROD
3333	TESTDTA	JDE_ROLE
3333	TESTDTA	JDE_NONPROD
77	PS812CTL	JDE_ROLE
77	PS812CTL	JDE_NONPROD
3333	PS812DTA	JDE_ROLE
3333	PS812DTA	JDE_NONPROD
5	ORDPLUGINS	PUBLIC

Step Five: Update the Schema Owner User Passwords

The customer should change all of the schema owner account passwords at the database account level. Our next set of scripts will accomplish changing ALL EnterpriseOne schema owner passwords. As delivered each EnterpriseOne schema owner has a corresponding database user account. The database account password is the same as the database account user name.

Replacement of Substitution Variables

We begin by querying the EnterpriseOne Data Source Master table (F98611). With this information we can update the enclosed 05_set_run_UpdSchemaPswdE1.bat file. The template batch file contains substitution variables which must be replaced with valid values. To properly update this batch file we must first ascertain valid EnterpriseOne schema owners found in Data Source Master.

Figure 38: Identifying EnterpriseOne Database Schema Owners

Query:
SELECT DISTINCT(OMOOWN) OWNER FROM SY812.F98611 WHERE OMDSTP = 'O' ORDER BY 1;

Sample Result Set:
SQL> SELECT DISTINCT(OMOOWN) OWNER FROM SY812.F98611 WHERE OMDSTP = 'O' ORDER BY 1;

OWNER

CRPCTL
CRPDTA
DD812
DV812
OL812
PD812
PRODCTL
PRODDTA
PS812
PS812CTL
PS812DTA

OWNER

```

PY812
SVM812
SY812
TESTCTL
TESTDTA

16 rows selected.

SQL>

```

In the template file shown below we will substitute a valid schema owner *account*, schema owner *password*, *connect string*, and *role name(s)*. Some of these values are derived from the prior query run against the Data Source Master (F98611) table.

Figure 39: Replacement of Substitution Variables (05_set_run_UpdSchemaPswdE1.bat)

Excerpt of batch file:

```

:
: call <script>.bat <system> <sys_pswd> <connect_string> <schema_owner> <schema_own_pswd>
:
: */
:
call UpdSchemaPswd.bat system oracle1 mlsan221 CRPCTL COMPLEX_PSWD
call UpdSchemaPswd.bat system oracle1 mlsan221 CRPDTA COMPLEX_PSWD
call UpdSchemaPswd.bat system oracle1 mlsan221 DD812 COMPLEX_PSWD
call UpdSchemaPswd.bat system oracle1 mlsan221 DV812 COMPLEX_PSWD
call UpdSchemaPswd.bat system oracle1 mlsan221 OL812 COMPLEX_PSWD
call UpdSchemaPswd.bat system oracle1 mlsan221 PD812 COMPLEX_PSWD
call UpdSchemaPswd.bat system oracle1 mlsan221 PRODCTL COMPLEX_PSWD
call UpdSchemaPswd.bat system oracle1 mlsan221 PRODDTA COMPLEX_PSWD
call UpdSchemaPswd.bat system oracle1 mlsan221 PS812 COMPLEX_PSWD
call UpdSchemaPswd.bat system oracle1 mlsan221 PS812CTL COMPLEX_PSWD
call UpdSchemaPswd.bat system oracle1 mlsan221 PS812DTA COMPLEX_PSWD
call UpdSchemaPswd.bat system oracle1 mlsan221 PY812 COMPLEX_PSWD
call UpdSchemaPswd.bat system oracle1 mlsan221 SVM812 COMPLEX_PSWD
call UpdSchemaPswd.bat system oracle1 mlsan221 SY812 COMPLEX_PSWD
call UpdSchemaPswd.bat system oracle1 mlsan221 TESTCTL COMPLEX_PSWD
call UpdSchemaPswd.bat system oracle1 mlsan221 TESTDTA COMPLEX_PSWD

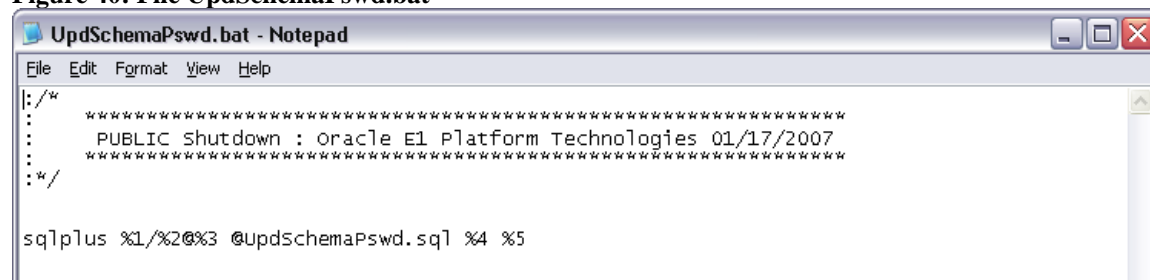
```

Once the substitution variables are updated the batch file is ready for use.

Under the Covers: 05_set_run_UpdSchemaPswdE1.bat

Under the covers the batch file 05_set_run_UpdSchemaPswdE1.bat will call batch file UpdSchemaPswd.bat.

Figure 40: File UpdSchemaPswd.bat



The UpdSchemaPswd.bat file will in turn call the UpdSchemaPswd.sql file. These batch files will use a series of passed parameters to login to the EnterpriseOne Oracle database as a user *system*. A portion of these

parameters will be passed to the SQL file so the PL/SQL logic can construct dynamic SQL and perform the database schema owner password update.

For this procedure two parameters are passed to the SQL file. A SQL statement will be constructed to change the password using the user and complex password information provided. This procedure will be called multiple times, e.g. once for each call statement found in the 05_set_run_UpdSchemaPswdE1.bat file.

Figure 41: File UpdSchemaPswd.sql

```
SET SERVEROUTPUT ON
SET TERMOUT ON

DECLARE
    v_cursor    INTEGER;
    v_user       VARCHAR(40) := '&1';
    v_pswd       VARCHAR(40) := '&2';
    sql_stmt1    VARCHAR(120);
    sql_stmt2    VARCHAR(120);

BEGIN
    v_cursor := DBMS_SQL.OPEN_CURSOR;

    sql_stmt1:='ALTER USER ' || v_user || ' IDENTIFIED BY &2' ;

    DBMS_SQL.PARSE
        (v_cursor , sql_stmt1,DBMS_SQL.V7);

END;
/
exit
```

Running 05_set_run_UpdSchemaPswdE1.bat file:

This script should be run from an open command window, e.g. Start | Run | cmd. Change the directory to the location of the script folder, e.g. x:\ORACLE_PUBLIC_SHUTDOWN. Command 05_set_run_UpdSchemaPswdE1.bat can be run from here. This procedure permits the user to validate success of the script.

Figure 42: Running 05_set_run_UpdSchemaPswdE1.bat

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\JDE>cd C:\ORACLE_PUBLIC_SHUTDOWN

C:\ORACLE_PUBLIC_SHUTDOWN>05_set_run_UpdSchemaPswdE1.bat

C:\ORACLE_PUBLIC_SHUTDOWN>call UpdSchemaPswd.bat system oracle1 mlsan221
CRPCTL COMPLEX_PSWD

C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus system/oracle1@mlsan221 @UpdSchemaPswd.sql CRP
CTL COMPLEX_PSWD

SQL*Plus: Release 9.2.0.6.0 - Production on Mon Nov 5 11:51:30 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

old 3:          v_user       VARCHAR(40) := '&1';
```

```

new 3:      v_user      VARCHAR(40) := 'CRPCTL';
old 4:      v_psswd     VARCHAR(40) := '&2';
new 4:      v_psswd     VARCHAR(40) := 'COMPLEX_PSWD';
old 11:      sql_stmt1:='ALTER USER ' || v_user || ' IDENTIFIED BY &
2' ;
new 11:      sql_stmt1:='ALTER USER ' || v_user || ' IDENTIFIED BY C
OMPLEX_PSWD' ;

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr
oduction
With the Partitioning, OLAP and Data Mining options

C:\ORACLE_PUBLIC_SHUTDOWN>call UpdSchemaPswd.bat system oracle1 mlsan221
CRPDTA COMPLEX_PSWD

C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus system/oracle1@mlsan221 @UpdSchemaPswd.sql CRP
DTA COMPLEX_PSWD

SQL*Plus: Release 9.2.0.6.0 - Production on Mon Nov 5 11:51:31 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

old 3:      v_user      VARCHAR(40) := '&1';
new 3:      v_user      VARCHAR(40) := 'CRPDTA';
old 4:      v_psswd     VARCHAR(40) := '&2';
new 4:      v_psswd     VARCHAR(40) := 'COMPLEX_PSWD';
old 11:      sql_stmt1:='ALTER USER ' || v_user || ' IDENTIFIED BY &
2' ;
new 11:      sql_stmt1:='ALTER USER ' || v_user || ' IDENTIFIED BY C
OMPLEX_PSWD' ;

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr
oduction
With the Partitioning, OLAP and Data Mining options

... <runs multiple times for each call statement>

C:\ORACLE_PUBLIC_SHUTDOWN>

```

The JDEdwards OneWorld/EnterpriseOne database schema owner accounts are used by the software. This occurs when running table conversions (TCs). Table conversions can occur as a consequence of applying ESUs, Updates, and Upgrades. Generally TCs are run from the deployment server.

Thus, after you change your database passwords update the JDE.INI found on the deployment server with the new password. For good measure you may want to only update these passwords with the correct value ONLY when you are running TCs. Consider either securing the file and/or only making these changes when using the Deployment Server for maintenance purposes. When you are not running TCs the values contained in this stanza are of no consequence to anything else.

The JDE.INI [DSPWD] stanza updates will act as overrides ultimately providing the underlying TC job to properly connect to the database as a designated schema owner using a valid password. A sample JDE.INI file [DSPWD] stanza is shown below.

Figure 43: JDE.INI File [DSPWD] Stanza

```
[DSPWD]
;Format is datasource OWNER=PASSWORD
SY811=0n3w0rld
SVM811=0n3w0rld
OL811=0n3w0rld
DD811=0n3w0rld
DV811=0n3w0rld
TESTDTA=0n3w0rld
TESTCTL=0n3w0rld
PD811=0n3w0rld
PRODDTA=0n3w0rld
PRODCTL=0n3w0rld
PS811=0n3w0rld
PS811DTA=0n3w0rld
PS811CTL=0n3w0rld
PY811=0n3w0rld
CRPDTA=0n3w0rld
CRPCTL=0n3w0rld
```

UNDO :: Granting *public* to EnterpriseOne Tables (Undo Prior Work)

A script has been created which will permit you to ‘undo’ the public shutdown. This is accomplished by granting public access to the EnterpriseOne tables. This script can be further modified for your own specific business needs.

Replacement of Substitution Variables

We begin by querying the EnterpriseOne Data Source Master table (F98611). With this information we can update the enclosed 99_set_UNDO_GrntPublicE1.bat file. The template batch file contains substitution variables which must be replaced with valid values. To properly update this batch file we must first ascertain valid EnterpriseOne schema owners found in Data Source Master.

Figure 44: Determining EnterpriseOne Database Schema Owners

Query:

```
SELECT OMDATP DTA_SRC, OMDATB CONNECT_STRNG, OMOOWN OWNER
FROM SY812.F98611 WHERE OMDSTP = 'O' ORDER BY 1;
```

Sample Result Set:

```
SQL> SELECT OMDATP DTA_SRC, OMDATB CONNECT_STRNG, OMOOWN OWNER
      2 FROM SY812.F98611 WHERE OMDSTP = 'O' ORDER BY 1;
```

DTA_SRC	CONNECT_STRNG	OWNER
Business Data - CRP	mlsan221	CRPDTA
Business Data - PROD	mlsan221	PRODDTA
Business Data - PS812	mlsan221	PS812DTA
Business Data - TEST	mlsan221	TESTDTA
Business Data - TEST - CIS	mlsan221	TESTDTA
Central Objects - DV812	mlsan221	DV812
Central Objects - PD812	mlsan221	PD812
Central Objects - PS812	mlsan221	PS812
Central Objects - PY812	mlsan221	PY812
Control Tables - CRP	mlsan221	CRPCTL
Control Tables - PS812	mlsan221	PS812CTL
DTA_SRC	CONNECT_STRNG	OWNER
Control Tables - Prod	mlsan221	PRODCTL
Control Tables - Test	mlsan221	TESTCTL
DENMLSAN221 - 812 Server Map	mlsan221	SVM812

Data Dictionary - 812	mlsan221	DD812
Data Dictionary - PS812	mlsan221	PS812CTL
Object Librarian - 812	mlsan221	OL812
System - 812	mlsan221	SY812
Versions - DV812	mlsan221	DV812
Versions - PD812	mlsan221	PD812
Versions - PS812	mlsan221	PS812
Versions - PY812	mlsan221	PY812

22 rows selected.

SQL>

In the template file shown below we will substitute a valid schema owner *account*, schema owner *password*, and *connect string*. Some of these values are derived from the prior query run against the Data Source Master (F98611) table.

Figure 45: Replacement of Substitution Variables (99_set_UNDO_GrntPublicE1.bat)

Excerpt of batch file:

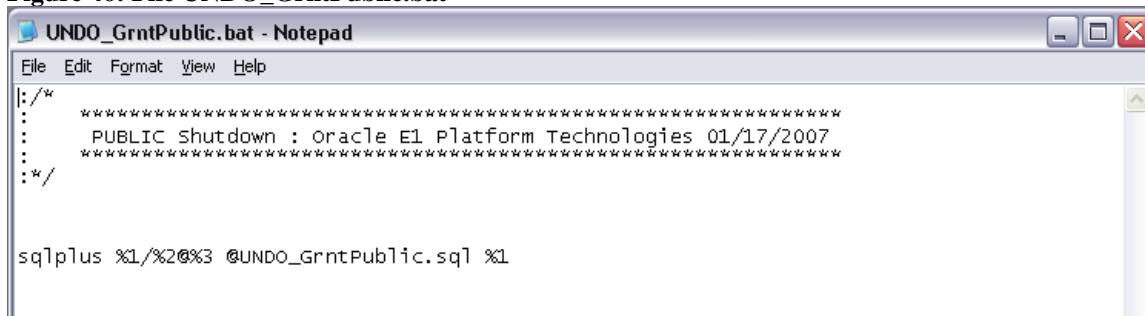
```
...
:
: call <script>.bat <schema_owner> <schema_owner_pswd> <connect_string>
:
:*/
:
:   Production Schemas
call UNDO_GrntPublic.bat PD812      COMPLEX_PSWD mlsan221
call UNDO_GrntPublic.bat PRODDTA    COMPLEX_PSWD mlsan221
call UNDO_GrntPublic.bat PRODCTL    COMPLEX_PSWD mlsan221
:
:   System Common Schemas
call UNDO_GrntPublic.bat SY812      COMPLEX_PSWD mlsan221
call UNDO_GrntPublic.bat SVM812     COMPLEX_PSWD mlsan221
call UNDO_GrntPublic.bat DD812      COMPLEX_PSWD mlsan221
call UNDO_GrntPublic.bat OL812      COMPLEX_PSWD mlsan221
:
:   Prototype Schemas
call UNDO_GrntPublic.bat PY812      COMPLEX_PSWD mlsan221
call UNDO_GrntPublic.bat CRPDTA     COMPLEX_PSWD mlsan221
call UNDO_GrntPublic.bat CRPCTL     COMPLEX_PSWD mlsan221
:
:   Development/Test Schemas
call UNDO_GrntPublic.bat DV812      COMPLEX_PSWD mlsan221
call UNDO_GrntPublic.bat TESTDTA    COMPLEX_PSWD mlsan221
call UNDO_GrntPublic.bat TESTCTL    COMPLEX_PSWD mlsan221
:
:   Pristine Schemas
call UNDO_GrntPublic.bat PS812      COMPLEX_PSWD mlsan221
call UNDO_GrntPublic.bat PS812DTA   COMPLEX_PSWD mlsan221
call UNDO_GrntPublic.bat PS812CTL   COMPLEX_PSWD mlsan221
```

Once the substitution variables are updated the batch file is ready for use.

Under the Covers: 99_set_UNDO_GrntPublicE1.bat

Under the covers the batch file 99_set_UNDO_GrntPublicE1.bat will call a second batch file named UNDO_GrntPublic.bat.

Figure 46: File UNDO_GrntPublic.bat



```
UNDO_GrntPublic.bat - Notepad
File Edit Format View Help
|:/*
:  ****
:  PUBLIC Shutdown : Oracle E1 Platform Technologies 01/17/2007
:  ****
:*/

sqlplus %1/%2@%3 @UNDO_GrntPublic.sql %1
```

The UNDO_GrntPublic.bat file will call the UNDO_GrntPublic.sql file. This batch file will use passed parameters to login to the EnterpriseOne Oracle database as a specific database schema owner. The SQL file will construct PL/SQL logic to grant public to database tables owned by the schema owner used during login.

Figure 47: File UNDO_GrntPublic.sql

```
SET SERVEROUTPUT ON
SET TERMOUT ON

DECLARE
    v_cursor    INTEGER;
    v_owner      VARCHAR(40) := '&1';
    sql_stmt1    VARCHAR(120);

    CURSOR object_list
    IS
        SELECT object_name
            FROM USER_OBJECTS
           WHERE OBJECT_TYPE = 'TABLE'
              OR OBJECT_TYPE = 'VIEW';

    v_object_name  object_list%ROWTYPE;

BEGIN
    v_cursor := DBMS_SQL.OPEN_CURSOR;
    OPEN object_list;

    FETCH object_list INTO v_object_name;

    WHILE object_list%FOUND
    LOOP

        sql_stmt1:='GRANT ALL PRIVILEGES ON ' || v_owner || '.' ||
                  v_object_name.object_name || ' TO PUBLIC' ;

        DBMS_SQL.PARSE
            (v_cursor , sql_stmt1,DBMS_SQL.V7);

        FETCH object_list INTO v_object_name;
    END LOOP;

    CLOSE object_list;

END;
/
exit
```

Running 99_set_UNDO_GrntPublicE1.bat file:

See [Appendix B](#) for information about clearing the DBA_RECYCLEBIN prior to running this script. Failure to do this may result in portions of the script failing for particular schema owners.

This script should be run from an open command window, e.g. Start | Run | cmd. Change the directory to the location of the script folder, e.g. x:\ORACLE_PUBLIC_SHUTDOWN. Command 99_set_UNDO_GrntPublicE1.bat can be run from here. This procedure permits the user to validate success of the script.

Figure 48: Running 99_set_UNDO_GrntPublicE1.bat

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\JDE>cd C:\ORACLE_PUBLIC_SHUTDOWN

C:\ORACLE_PUBLIC_SHUTDOWN> 99_set_UNDO_GrntPublicE1.bat

C:\ORACLE_PUBLIC_SHUTDOWN>call UNDO_GrntPublic.bat PD812 COMPLEX_PSWD
mlsan221

C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus PD812/COMPLEX_PSWD@mlsan221 @UNDO_GrntPublic.s
ql PD812

SQL*Plus: Release 9.2.0.6.0 - Production on Mon Nov 5 14:43:07 2007
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

old 3:          v_owner  VARCHAR(40) := '&1';
new 3:          v_owner  VARCHAR(40) := 'PD812';

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr
oduction
With the Partitioning, OLAP and Data Mining options

... <repeated as iterative call for each schema owner>

C:\ORACLE_PUBLIC_SHUTDOWN>
```

Evaluate & Compare Table Privileges by Role and 'public'

We can now evaluate privileges found on EnterpriseOne database tables. We should find that public has been granted back to tables owned by each schema owner. This SQL query requires DBA privileges; consider using login account 'system'.

Figure 49: Checking Table Privileges

```
Query:
SELECT COUNT(DISTINCT(TABLE_NAME)) TABLE_NO, OWNER, GRANTEE
FROM DBA_TAB_PRIVS WHERE OWNER NOT LIKE '%SYS%'
GROUP BY OWNER, GRANTEE ORDER BY 2;

Sample Result Set:
SQL> SELECT COUNT(DISTINCT(TABLE_NAME)) TABLE_NO, OWNER, GRANTEE
2 FROM DBA_TAB_PRIVS WHERE OWNER NOT LIKE '%SYS%'
3 GROUP BY OWNER, GRANTEE ORDER BY 2;

TABLE_NO OWNER GRANTEE
```

65	CRPCTL	JDE_NONPROD
65	CRPCTL	JDE_ROLE
65	CRPCTL	PUBLIC
3333	CRPDTA	JDE_NONPROD
3333	CRPDTA	JDE_ROLE
3333	CRPDTA	PUBLIC
7	DBSNMP	OEM_MONITOR
12	DD812	JDE_NONPROD
12	DD812	JDE_ROLE
12	DD812	PUBLIC
47	DV812	JDE_NONPROD
TABLE_NO	OWNER	GRANTEE
47	DV812	JDE_ROLE
47	DV812	PUBLIC
13	OL812	JDE_NONPROD
13	OL812	JDE_ROLE
13	OL812	PUBLIC
5	ORDPLUGINS	PUBLIC
3	OUTLN	SELECT_CATALOG_ROLE
47	PD812	JDE_ROLE
47	PD812	PUBLIC
65	PRODCTL	JDE_ROLE
65	PRODCTL	PUBLIC
TABLE_NO	OWNER	GRANTEE
3333	PRODDTA	JDE_ROLE
3333	PRODDTA	PUBLIC
47	PS812	JDE_NONPROD
47	PS812	JDE_ROLE
47	PS812	PUBLIC
77	PS812CTL	JDE_NONPROD
77	PS812CTL	JDE_ROLE
77	PS812CTL	PUBLIC
3333	PS812DTA	JDE_NONPROD
3333	PS812DTA	JDE_ROLE
3333	PS812DTA	PUBLIC
TABLE_NO	OWNER	GRANTEE
47	PY812	JDE_NONPROD
47	PY812	JDE_ROLE
47	PY812	PUBLIC
7	SVM812	JDE_NONPROD
7	SVM812	JDE_ROLE
7	SVM812	PUBLIC
295	SY812	JDE_NONPROD
295	SY812	JDE_ROLE
295	SY812	PUBLIC
65	TESTCTL	JDE_NONPROD
65	TESTCTL	JDE_ROLE
TABLE_NO	OWNER	GRANTEE
65	TESTCTL	PUBLIC
3333	TESTDTA	JDE_NONPROD
3333	TESTDTA	JDE_ROLE
3333	TESTDTA	PUBLIC
187	XDB	PUBLIC
1	XDB	XDBADMIN

```
50 rows selected.
```

```
SQL>
```

As for the database roles we previously created and their associated privileges you can use SQL command(s) `DROP ROLE JDE_ROLE;` and `DROP ROLE JDE_NONPROD;` to eliminate these roles and their associated privileges. These SQL queries require DBA privileges; as before consider using login account 'system'.

Appendix A: Result 02_set_run_GrantRevokeE1.bat

The iterative and complete result of running batch file 02_set_run_GrantRevokeE1.bat from command window is shown directly below.

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\JDE>cd C:\ORACLE_PUBLIC_SHUTDOWN

C:\ORACLE_PUBLIC_SHUTDOWN>02_set_run_GrantRevokeE1.bat

C:\ORACLE_PUBLIC_SHUTDOWN>call GrantRevokePROD.bat PD812 PD812 mlsan
221 JDE_ROLE

C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus PD812/PD812@mlsan221 @GrantRevokePROD.sql PD81
2 JDE_ROLE

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 16:14:20 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

old 3:          v_owner  VARCHAR(40) :='&1';
new 3:          v_owner  VARCHAR(40) :='PD812';
old 4:          v_role1  VARCHAR(40) :='&2';
new 4:          v_role1  VARCHAR(40) :='JDE_ROLE';
old 28:          v_object_name.object_name || ' TO &2' ;
new 28:          v_object_name.object_name || ' TO JDE_ROLE' ;

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr
oduction
With the Partitioning, OLAP and Data Mining options

C:\ORACLE_PUBLIC_SHUTDOWN>call GrantRevokePROD.bat PRODDTA PRODDTA mlsan
221 JDE_ROLE

C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus PRODDTA/PRODDTA@mlsan221 @GrantRevokePROD.sql
PRODDTA JDE_ROLE

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 16:14:24 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

old 3:          v_owner  VARCHAR(40) :='&1';
new 3:          v_owner  VARCHAR(40) :='PRODDTA';
old 4:          v_role1  VARCHAR(40) :='&2';
new 4:          v_role1  VARCHAR(40) :='JDE_ROLE';
old 28:          v_object_name.object_name || ' TO &2' ;
new 28:          v_object_name.object_name || ' TO JDE_ROLE' ;

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr
oduction
With the Partitioning, OLAP and Data Mining options
```

```
C:\ORACLE_PUBLIC_SHUTDOWN>call GrantRevokePROD.bat  PRODC TL      PRODC TL      mlsan
221  JDE_ROLE
```

```
C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus  PRODC TL/PRODC TL@mlsan221  @GrantRevokePROD.sql
PRODC TL JDE_ROLE
```

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 16:16:14 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:

Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

```
old  3:          v_owner    VARCHAR(40) :='&1';
new  3:          v_owner    VARCHAR(40) :='PRODC TL';
old  4:          v_role1    VARCHAR(40) :='&2';
new  4:          v_role1    VARCHAR(40) :='JDE_ROLE';
old 28:          v_object_name.object_name || ' TO &2' ;
new 28:          v_object_name.object_name || ' TO JDE_ROLE' ;
```

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr
oduction
With the Partitioning, OLAP and Data Mining options

```
C:\ORACLE_PUBLIC_SHUTDOWN>call GrntRvokNONPROD.bat  SY812      SY812      mlsan
221  JDE_ROLE  JDE_NONPROD
```

```
C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus  SY812/SY812@mlsan221  @GrntRvokNONPROD.sql SY81
2 JDE_ROLE JDE_NONPROD
```

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 16:16:18 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:

Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

```
old  3:          v_owner    VARCHAR(40) :='&1';
new  3:          v_owner    VARCHAR(40) :='SY812';
old  4:          v_role1    VARCHAR(40) :='&2';
new  4:          v_role1    VARCHAR(40) :='JDE_ROLE';
old  5:          v_role2    VARCHAR(40) :='&3';
new  5:          v_role2    VARCHAR(40) :='JDE_NONPROD';
old 29:          v_object_name.object_name || ' TO &2, &3' ;
new 29:          v_object_name.object_name || ' TO JDE_ROLE, JDE
_NONPROD' ;
```

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr
oduction
With the Partitioning, OLAP and Data Mining options

```
C:\ORACLE_PUBLIC_SHUTDOWN>call GrntRvokNONPROD.bat  SVM812      SVM812      mlsan
221  JDE_ROLE  JDE_NONPROD
```

```
C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus  SVM812/SVM812@mlsan221  @GrntRvokNONPROD.sql SV
M812 JDE_ROLE JDE_NONPROD
```

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 16:16:21 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:

Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

```
old 3:      v_owner  VARCHAR(40) := '&1';
new 3:      v_owner  VARCHAR(40) := 'SVM812';
old 4:      v_role1  VARCHAR(40) := '&2';
new 4:      v_role1  VARCHAR(40) := 'JDE_ROLE';
old 5:      v_role2  VARCHAR(40) := '&3';
new 5:      v_role2  VARCHAR(40) := 'JDE_NONPROD';
old 29:          v_object_name.object_name || ' TO &2, &3' ;
new 29:          v_object_name.object_name || ' TO JDE_ROLE, JDE
_NONPROD' ;
```

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr
oduction
With the Partitioning, OLAP and Data Mining options

C:\ORACLE_PUBLIC_SHUTDOWN>call GrntRvokNONPROD.bat DD812 DD812 mlsan
221 JDE_ROLE JDE_NONPROD

C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus DD812/DD812@mlsan221 @GrntRvokNONPROD.sql DD81
2 JDE_ROLE JDE_NONPROD

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 16:16:23 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:

Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

```
old 3:      v_owner  VARCHAR(40) := '&1';
new 3:      v_owner  VARCHAR(40) := 'DD812';
old 4:      v_role1  VARCHAR(40) := '&2';
new 4:      v_role1  VARCHAR(40) := 'JDE_ROLE';
old 5:      v_role2  VARCHAR(40) := '&3';
new 5:      v_role2  VARCHAR(40) := 'JDE_NONPROD';
old 29:          v_object_name.object_name || ' TO &2, &3' ;
new 29:          v_object_name.object_name || ' TO JDE_ROLE, JDE
_NONPROD' ;
```

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr
oduction
With the Partitioning, OLAP and Data Mining options

C:\ORACLE_PUBLIC_SHUTDOWN>call GrntRvokNONPROD.bat OL812 OL812 mlsan
221 JDE_ROLE JDE_NONPROD

C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus OL812/OL812@mlsan221 @GrntRvokNONPROD.sql OL81
2 JDE_ROLE JDE_NONPROD

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 16:16:25 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:

Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

```
old 3:      v_owner  VARCHAR(40) := '&1';
new 3:      v_owner  VARCHAR(40) := 'OL812';
old 4:      v_role1  VARCHAR(40) := '&2';
new 4:      v_role1  VARCHAR(40) := 'JDE_ROLE';
old 5:      v_role2  VARCHAR(40) := '&3';
new 5:      v_role2  VARCHAR(40) := 'JDE_NONPROD';
```

```
old 29:          v_object_name.object_name || ' TO &2, &3' ;
new 29:          v_object_name.object_name || ' TO JDE_ROLE, JDE
_NONPROD' ;
```

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

```
C:\ORACLE_PUBLIC_SHUTDOWN>call GrntRvokNONPROD.bat PY812 PY812 mlsan
221 JDE_ROLE JDE_NONPROD
```

```
C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus PY812/PY812@mlsan221 @GrntRvokNONPROD.sql PY81
2 JDE_ROLE JDE_NONPROD
```

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 16:16:28 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:

Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

```
old 3:          v_owner   VARCHAR(40) := '&1' ;
new 3:          v_owner   VARCHAR(40) := 'PY812' ;
old 4:          v_role1   VARCHAR(40) := '&2' ;
new 4:          v_role1   VARCHAR(40) := 'JDE_ROLE' ;
old 5:          v_role2   VARCHAR(40) := '&3' ;
new 5:          v_role2   VARCHAR(40) := 'JDE_NONPROD' ;
old 29:          v_object_name.object_name || ' TO &2, &3' ;
new 29:          v_object_name.object_name || ' TO JDE_ROLE, JDE
_NONPROD' ;
```

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

```
C:\ORACLE_PUBLIC_SHUTDOWN>call GrntRvokNONPROD.bat CRPDTA CRPDTA mlsan
221 JDE_ROLE JDE_NONPROD
```

```
C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus CRPDTA/CRPDTA@mlsan221 @GrntRvokNONPROD.sql CR
PD TA JDE_ROLE JDE_NONPROD
```

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 16:16:32 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:

Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

```
old 3:          v_owner   VARCHAR(40) := '&1' ;
new 3:          v_owner   VARCHAR(40) := 'CRPD TA' ;
old 4:          v_role1   VARCHAR(40) := '&2' ;
new 4:          v_role1   VARCHAR(40) := 'JDE_ROLE' ;
old 5:          v_role2   VARCHAR(40) := '&3' ;
new 5:          v_role2   VARCHAR(40) := 'JDE_NONPROD' ;
old 29:          v_object_name.object_name || ' TO &2, &3' ;
new 29:          v_object_name.object_name || ' TO JDE_ROLE, JDE
_NONPROD' ;
```

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

```
C:\ORACLE_PUBLIC_SHUTDOWN>call GrntRvokNONPROD.bat CRPCTL CRPCTL mlsan
221 JDE_ROLE JDE_NONPROD
```

```
C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus CRPCTL/CRPCTL@mlsan221 @GrntRvokNONPROD.sql CR
PTCL JDE_ROLE JDE_NONPROD
```

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 16:18:49 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:

Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

```
old 3:          v_owner  VARCHAR(40) :='&1';
new 3:          v_owner  VARCHAR(40) :='CRPCTL';
old 4:          v_role1  VARCHAR(40) :='&2';
new 4:          v_role1  VARCHAR(40) :='JDE_ROLE';
old 5:          v_role2  VARCHAR(40) :='&3';
new 5:          v_role2  VARCHAR(40) :='JDE_NONPROD';
old 29:                                     v_object_name.object_name || ' TO &2, &3' ;
new 29:                                     v_object_name.object_name || ' TO JDE_ROLE, JDE
_NONPROD' ;
```

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr
oduction
With the Partitioning, OLAP and Data Mining options

```
C:\ORACLE_PUBLIC_SHUTDOWN>call GrntRvokNONPROD.bat DV812 DV812 mlsan
221 JDE_ROLE JDE_NONPROD
```

```
C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus DV812/DV812@mlsan221 @GrntRvokNONPROD.sql DV81
2 JDE_ROLE JDE_NONPROD
```

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 16:18:53 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:

Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

```
old 3:          v_owner  VARCHAR(40) :='&1';
new 3:          v_owner  VARCHAR(40) :='DV812';
old 4:          v_role1  VARCHAR(40) :='&2';
new 4:          v_role1  VARCHAR(40) :='JDE_ROLE';
old 5:          v_role2  VARCHAR(40) :='&3';
new 5:          v_role2  VARCHAR(40) :='JDE_NONPROD';
old 29:                                     v_object_name.object_name || ' TO &2, &3' ;
new 29:                                     v_object_name.object_name || ' TO JDE_ROLE, JDE
_NONPROD' ;
```

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr
oduction
With the Partitioning, OLAP and Data Mining options

```
C:\ORACLE_PUBLIC_SHUTDOWN>call GrntRvokNONPROD.bat TESTDTA TESTDTA mlsan
221 JDE_ROLE JDE_NONPROD
```

```
C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus TESTDTA/TESTDTA@mlsan221 @GrntRvokNONPROD.sql
TESTDTA JDE_ROLE JDE_NONPROD
```

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 16:18:57 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:

Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

```
old 3:      v_owner  VARCHAR(40) := '&1';
new 3:      v_owner  VARCHAR(40) := 'TESTDTA';
old 4:      v_role1  VARCHAR(40) := '&2';
new 4:      v_role1  VARCHAR(40) := 'JDE_ROLE';
old 5:      v_role2  VARCHAR(40) := '&3';
new 5:      v_role2  VARCHAR(40) := 'JDE_NONPROD';
old 29:      v_object_name.object_name || ' TO &2, &3' ;
new 29:      v_object_name.object_name || ' TO JDE_ROLE, JDE
_NONPROD' ;
```

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr
oduction
With the Partitioning, OLAP and Data Mining options

```
C:\ORACLE_PUBLIC_SHUTDOWN>call GrntRvokNONPROD.bat TESTCTL TESTCTL mlsan
221 JDE_ROLE JDE_NONPROD
```

```
C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus TESTCTL/TESTCTL@mlsan221 @GrntRvokNONPROD.sql
TESTCTL JDE_ROLE JDE_NONPROD
```

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 16:21:12 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:

Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

```
old 3:      v_owner  VARCHAR(40) := '&1';
new 3:      v_owner  VARCHAR(40) := 'TESTCTL';
old 4:      v_role1  VARCHAR(40) := '&2';
new 4:      v_role1  VARCHAR(40) := 'JDE_ROLE';
old 5:      v_role2  VARCHAR(40) := '&3';
new 5:      v_role2  VARCHAR(40) := 'JDE_NONPROD';
old 29:      v_object_name.object_name || ' TO &2, &3' ;
new 29:      v_object_name.object_name || ' TO JDE_ROLE, JDE
_NONPROD' ;
```

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr
oduction
With the Partitioning, OLAP and Data Mining options

```
C:\ORACLE_PUBLIC_SHUTDOWN>call GrntRvokNONPROD.bat PS812 PS812 mlsan
221 JDE_ROLE JDE_NONPROD
```

```
C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus PS812/PS812@mlsan221 @GrntRvokNONPROD.sql PS81
2 JDE_ROLE JDE_NONPROD
```

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 16:21:16 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:

Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

```
old 3:      v_owner  VARCHAR(40) := '&1';
new 3:      v_owner  VARCHAR(40) := 'PS812';
```

```

old 4:      v_role1  VARCHAR(40) := '&2';
new 4:      v_role1  VARCHAR(40) := 'JDE_ROLE';
old 5:      v_role2  VARCHAR(40) := '&3';
new 5:      v_role2  VARCHAR(40) := 'JDE_NONPROD';
old 29:      v_object_name.object_name || ' TO &2, &3' ;
new 29:      v_object_name.object_name || ' TO JDE_ROLE, JDE
_NONPROD' ;

```

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

```

C:\ORACLE_PUBLIC_SHUTDOWN>call GrntRvokNONPROD.bat PS812DTA PS812DTA mlsan
221 JDE_ROLE JDE_NONPROD

```

```

C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus PS812DTA/PS812DTA@mlsan221 @GrntRvokNONPROD.sq
l PS812DTA JDE_ROLE JDE_NONPROD

```

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 16:21:20 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:

Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

```

old 3:      v_owner  VARCHAR(40) := '&1';
new 3:      v_owner  VARCHAR(40) := 'PS812DTA';
old 4:      v_role1  VARCHAR(40) := '&2';
new 4:      v_role1  VARCHAR(40) := 'JDE_ROLE';
old 5:      v_role2  VARCHAR(40) := '&3';
new 5:      v_role2  VARCHAR(40) := 'JDE_NONPROD';
old 29:      v_object_name.object_name || ' TO &2, &3' ;
new 29:      v_object_name.object_name || ' TO JDE_ROLE, JDE
_NONPROD' ;

```

PL/SQL procedure successfully completed.

Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

```

C:\ORACLE_PUBLIC_SHUTDOWN>call GrntRvokNONPROD.bat PS812CTL PS812CTL mlsan
221 JDE_ROLE JDE_NONPROD

```

```

C:\ORACLE_PUBLIC_SHUTDOWN>sqlplus PS812CTL/PS812CTL@mlsan221 @GrntRvokNONPROD.sq
l PS812CTL JDE_ROLE JDE_NONPROD

```

SQL*Plus: Release 10.2.0.2.0 - Production on Sun Nov 4 16:23:36 2007

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

Connected to:

Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Production
With the Partitioning, OLAP and Data Mining options

```

old 3:      v_owner  VARCHAR(40) := '&1';
new 3:      v_owner  VARCHAR(40) := 'PS812CTL';
old 4:      v_role1  VARCHAR(40) := '&2';
new 4:      v_role1  VARCHAR(40) := 'JDE_ROLE';
old 5:      v_role2  VARCHAR(40) := '&3';
new 5:      v_role2  VARCHAR(40) := 'JDE_NONPROD';
old 29:      v_object_name.object_name || ' TO &2, &3' ;
new 29:      v_object_name.object_name || ' TO JDE_ROLE, JDE
_NONPROD' ;

```

PL/SQL procedure successfully completed.

```
Disconnected from Oracle Database 10g Enterprise Edition Release 10.2.0.2.0 - Pr  
oduction  
With the Partitioning, OLAP and Data Mining options  
C:\ORACLE_PUBLIC_SHUTDOWN>
```

Appendix B: Purging the DBA_RECYCLEBIN

With Oracle 10g and higher it is may be necessary to purge the DBA_RECYCLEBIN in order to run the 02_set_run_GrantRevokeE1.bat or 99_set_UNDO_GrntPublicE1.bat scripts. These two scripts will otherwise fail if recycle bin tables exist for the schema being updated. One must login as user “sys / as SYSDBA” to purge the DBA_RECYCLEBIN as shown below.

```
SQL> CONNECT SYSTEM/ORACLE1@MLSAN221
Connected.
SQL> SELECT OWNER, OBJECT_NAME, ORIGINAL_NAME FROM DBA_RECYCLEBIN;

OWNER                                OBJECT_NAME                          ORIGINAL_NAME
-----                                -
SY812                                BIN$8nQtX/dsQvW5pReX5iBT4A==$0      OWNER
DV812                                BIN$hb0csS3EQD6/VkyUvBUycw==$0      F983051_2
DV812                                BIN$R/pWgtRyQ9m4sGLrGxVpjA==$0      F983051_3
DV812                                SYS_IL0000085266C00021$$            SYS_IL0000085266C00021$$
DV812                                BIN$Nlg7kKtoRvaRNm8Tem5+dA==$0      F983051_0
DV812                                SYS_LOB0000085266C00021$$            SYS_LOB0000085266C00021$$
DV812                                BIN$OlKuSBDXTDSwp9sxvfwf4Q==$0      F983051

7 rows selected.

SQL> SELECT OBJECT_NAME FROM DBA_RECYCLEBIN;

OBJECT_NAME
-----
BIN$8nQtX/dsQvW5pReX5iBT4A==$0
BIN$hb0csS3EQD6/VkyUvBUycw==$0
BIN$R/pWgtRyQ9m4sGLrGxVpjA==$0
SYS_IL0000085266C00021$$
BIN$Nlg7kKtoRvaRNm8Tem5+dA==$0
SYS_LOB0000085266C00021$$
BIN$OlKuSBDXTDSwp9sxvfwf4Q==$0

7 rows selected.

SQL>

SQL> CONNECT SYS/ORACLE1@MLSAN221 AS SYSDBA
Connected.
SQL> PURGE DBA_RECYCLEBIN;

DBA Recyclebin purged.

SQL>
```

Appendix C: Securing HRM and Payroll Tables

The techniques and principals used herein could be applied more granularly. For instance, one could create a JDE_HRMPAY database role and associate EnterpriseOne system (proxy) users who require HRM and Payroll access to the JDE_HRMPAY database role. The JDE_HRMPAY database role could then be granted ALL access to HRM & Payroll tables and the both public and JDE_ROLE (example) could have their privileges to these same tables revoked. **Note: In all cases JDE (or PSFT) must maintain ALL access to ALL EnterpriseOne tables. This ensures that upgrades and/or updates are successful.**

For example, in the figure shown below we find table F07490 (Payroll - Accounts Payable Detail) is only accessible by database role JDE_PAYHRM. This means only user logins who are members of the JDE_PAYHRM database role can access this table.

Figure 50: Payroll Table Secured by Special Database Role JDE_PAYHRM

Query: SELECT TABLE_NAME, OWNER, GRANTEE, PRIVILEGE FROM DBA_TAB_PRIVS WHERE OWNER = 'PRODDTA' AND TABLE_NAME = 'F07490' AND OWNER NOT LIKE '%SYS%' ORDER BY OWNER, GRANTEE;			
Sample Result Set:			
SQL> SELECT TABLE_NAME, OWNER, GRANTEE, PRIVILEGE 2 FROM DBA_TAB_PRIVS WHERE OWNER = 'PRODDTA' 3 AND TABLE_NAME = 'F07490' AND OWNER NOT LIKE '%SYS%' 4 ORDER BY OWNER, GRANTEE;			
TABLE_NAME	OWNER	GRANTEE	PRIVILEGE
-----	-----	-----	-----
F07490	PRODDTA	JDE_PAYHRM	DEBUG
F07490	PRODDTA	JDE_PAYHRM	QUERY REWRITE
F07490	PRODDTA	JDE_PAYHRM	ON COMMIT REFRESH
F07490	PRODDTA	JDE_PAYHRM	UPDATE
F07490	PRODDTA	JDE_PAYHRM	FLASHBACK
F07490	PRODDTA	JDE_PAYHRM	INSERT
F07490	PRODDTA	JDE_PAYHRM	DELETE
F07490	PRODDTA	JDE_PAYHRM	ALTER
F07490	PRODDTA	JDE_PAYHRM	SELECT
9 rows selected.			
SQL>			

The JDE_PAYHRM database role members include the user login JDE (*primary EnterpriseOne login*), JDE_HRM, and JDE_PAY user logins. Both the JDE_HRM and JDE_PAY user logins are defined as EnterpriseOne system (proxy) user accounts. Thus EnterpriseOne user accounts requiring access to this Payroll table(s) must be assigned to the database (proxy) user account of JDE_PAY or JDE_HRM.

Figure 51: Payroll Table Secured by Special Database Role JDE_PAYHRM

Query: SELECT GRANTEE, GRANTED_ROLE FROM DBA_ROLE_PRIVS
WHERE GRANTED_ROLE = 'JDE_PAYHRM' ORDER BY 1;

Sample Result Set

SQL> SELECT GRANTEE, GRANTED_ROLE FROM DBA_ROLE_PRIVS
2 WHERE GRANTED_ROLE = 'JDE_PAYHRM' ORDER BY 1;

GRANTEE	GRANTED_ROLE
JDE	JDE_PAYHRM
JDE_HRM	JDE_PAYHRM
JDE_PAY	JDE_PAYHRM
SYSTEM	JDE_PAYHRM

SQL>

Please recall the relationship between JDEdwards EnterpriseOne ERP users and proxy users is maintained in table F98OWSEC (application P98OWSEC for OneWorld Security). In releases 8.9 and higher table F98OWPU was introduced to further segment the system (proxy) user credential information. The F98OWPU table is accessed via application P980001 for *Work with OneWorld System Users*. In the sample query shown below the first two columns illustrate how the ERP user and proxy user are cross-referenced. The third column maps to the related system/database (proxy) user table.

Figure 52: Security Server Tables (F98OWSEC & F98OWPU)

Query: SELECT T0.SCUSER, T0.SCSECUSR, T1.PUSECUSR
FROM SY812.F98OWSEC T0, SY812.F98OWPU T1
WHERE T0.SCSECUSR = T1.PUSECUSR;

Sample Result Set:

SCUSER	SCSECUSR	PUSECUSR
JDE	JDE	JDE
JDEUSR1	JDEUSR1	JDEUSR1
JDEUSR2	JDEUSR2	JDEUSR2
JDEUSR3	JDEFIN	JDEFIN
JDEUSR4	JDEHRM	JDEHRM
KJUDSON	JDEPAY	JDEPAY
KPOND	JDEHRM	JDEHRM
BWILSON	JDEFIN	JDEFIN
MOCONNER	JDESCM	JDESCM
JKOGER	JDEMFG	JDEMFG
JMAKUSKY	JDECNC	JDECNC
...		

From the sample result set above EnterpriseOne users JDEUSR4, KJUDSON, and KPOND would have access to the F07490 (Payroll - Accounts Payable Detail) table.

Appendix D: Maintaining a Secured Database

Consider that each time you upgrade, update, or apply add-ons to EnterpriseOne whereby tables were changed or added there may be need to reevaluate privileges on your database table(s). One can do this by re-running batch script 04_set_run_CompareTablePrivs.bat.

Also by default triggers and shadow tables created for EnterpriseOne 21 CFR Part 11 will be owned by the database schema owner. This ownership retention ensures access by those triggers to the schema specific database tables.

Figure 53: 21 CFR Part 11 (Code of Federal Regulation) Database Triggers & Shadow Tables

The screenshot shows the Oracle Enterprise Manager 10g Database Control interface. The browser address bar shows the URL: `http://denoshp11:1158/em/console/database/databaseObjectsSearch?target=orclhp&type=oracle_database&otype=schema`. The page title is "Oracle Enterprise Manager (SYS) - Triggers - Microsoft Internet Explorer". The main content area is titled "Triggers" and shows the "Database Instance: orclhp" and "Logged in As SYS". The "Object Type" dropdown is set to "Trigger". The "Search" section includes fields for "Schema" (CRPDTA), "Object Name", and "Status" (All). A "Go" button is present. Below the search section, the "Selection Mode" is set to "Single". A table of triggers is displayed with the following columns: Select, Schema, Trigger Name, Type, Event, Base Object Type, Base Object Owner, Base Object Name, Status, and Enabled?. The table contains five rows of triggers for the CRPDTA schema.

Select	Schema	Trigger Name	Type	Event	Base Object Type	Base Object Owner	Base Object Name	Status	Enabled?
<input checked="" type="radio"/>	CRPDTA	F0101_CFRA_RDA	AFTER EACH ROW	DELETE	TABLE	CRPDTA	F0101_ADT	VALID	YES
<input type="radio"/>	CRPDTA	F0101_CFRA_RIA	BEFORE EACH ROW	INSERT	TABLE	CRPDTA	F0101_ADT	VALID	YES
<input type="radio"/>	CRPDTA	F0101_CFRA_RUA	BEFORE EACH ROW	UPDATE	TABLE	CRPDTA	F0101_ADT	VALID	YES
<input type="radio"/>	CRPDTA	F0101_CFRA_SUA	AFTER STATEMENT	UPDATE	TABLE	CRPDTA	F0101_ADT	VALID	YES
<input type="radio"/>	CRPDTA	F0101_CFRA_SUB	BEFORE STATEMENT	UPDATE	TABLE	CRPDTA	F0101_ADT	VALID	YES



Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Copyright © 2006, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This document is for informational purposes only and may not be incorporated into a contract or agreement.