

Using TDE Tablespace Encryption with Oracle E-Business Suite Release 12 [ID 828229.1]

Modified 28-MAR-2011 Type WHITE PAPER Status PUBLISHED

This document describes using the Transparent Data Encryption Tablespace Encryption option of Oracle Database 11g Enterprise Edition with Oracle E-Business Suite Release 12. You should read and understand the content before you begin your installation.

The most current version of this document can be obtained on My Oracle Support Knowledge document <[828229.1](#)>.

There is a [change log](#) at the end of this document.

[Section 1: Introduction](#)

[Section 2: Performance](#)

[Section 3: Outline of Tasks](#)

[Section 4: Prepare the Source System](#)

[Section 5: Prepare a target Applications Release 12 database instance](#)

[Section 6: Export the source Applications Release 12 database instance](#)

[Section 7: Import the Applications Release 12 database instance](#)

[Section 8: Update the imported Applications Release 12 database instance](#)

Section 1: Introduction

The TDE Tablespace Encryption option can be used to encrypt the tablespaces that store the content of your E-Business Suite application. As the encryption is transparent to the application, code does not have to be rewritten, and existing SQL statements work as they are. Transparent also means that any authorized database session can read the encrypted data without any problem: the encryption only applies to data-at-rest, i.e. the database files and any backups of these files.

The SYSTEM and SYSAUX tablespaces cannot be encrypted as these are created during the Create Database command rather than with a Create Tablespace command. It is the Create Tablespace command that can be appended with the 'ENCRYPTION ...' command.

Undo and Temp tablespaces cannot be encrypted, but the content from encrypted tablespaces that is stored there temporarily is stored encrypted.

For details of how Oracle Dataguard and Oracle Streams function with TDE Tablespace Encryption, see the [Transparent Data Encryption \(TDE\) Frequently Asked Questions document](#).


There is also a TDE Column Encryption option that enables you to encrypt only specific columns that you have determined contain sensitive data.

If you are unsure whether to use TDE Column Encryption or Tablespace Encryption, review the information in the [Transparent Data Encryption \(TDE\) Frequently Asked Questions document](#).

If you determine that it is TDE Column Encryption that you wish to use, please refer to My Oracle Support Knowledge document <[732764.1](#)>.

If you wish to migrate from TDE Column Encryption to TDE Tablespace Encryption, then it is recommended that you decrypt any columns that have been encrypted before performing the tasks listed in this document.

This document describes the process of implementing TDE Tablespace Encryption for an existing Oracle E-Business Suite Release 12 database instance using the export and import utilities.

 The [export/import process](#) described in this document is typically used to migrate a database from one platform to another. As implementing TDE tablespace encryption requires the encrypted tablespace to be created with a new ENCRYPT parameter, new tablespaces must be created and then the data must be inserted into the new tablespace.

The export and import utilities allow you to move existing data in Oracle format to and from Oracle databases. This document assumes that you are already familiar with export and import.

Attention: If you intend to export or import an Oracle E-Business Suite database in which Database Vault has already been enabled, please see My Oracle Support Knowledge document <[822048.1](#)> for additional steps.

Section 2: Performance

In TDE Tablespace Encryption, the limitation of TDE Column Encryption in terms of supported data and index types no longer applies.

Using TDE Tablespace Encryption with Oracle E-Business Suite may incur a decrease in performance by up to approximately 10%. Refer to the [Advanced Security Administrator's Guide](#) for more information regarding performance.

Section 3: Outline of Tasks

There are special considerations when exporting or importing an Oracle E-Business Suite 12 database instance. This process consists of five discrete steps. Each step is covered in a separate section in this document.

- [Section 4: Prepare the Source System](#)
Tasks that must be performed to prepare the source system for the database export
- [Section 5: Prepare a target Applications Release 12 Database Instance](#)
Tasks for creating an empty database instance in preparation for import
- [Section 6: Export the Source Applications Release 12 Database Instance](#)
Tasks that must be performed to produce a valid export of an Applications Release 12 database instance
- [Section 7: Import the Applications Release 12 Database Instance](#)
Tasks for running the import utility
- [Section 8: Update the Applications Imported Release 12 Database Instance](#)
Tasks that must be performed to restore the imported Applications Release 12 Database Instance to a fully functional state

The source (export from) ORACLE_HOME directories must be Oracle Database 10g Release 2 (10.2.0) or Oracle Database 11g (11.x). The target (import to) ORACLE_HOME directories must be Oracle Database 11g (11.x).

The export/import process requires the use of the datapump utilities (expdp/impdp). Datapump offers many advantages compared to traditional export/import (exp/imp). It is significantly faster, and it incorporates new features such as restarting from the point of failure and parallel processing. For more information, read [Oracle Database Utilities 11g Release 1 \(11.1\)](#) or [Oracle Database Utilities 11g Release 2 \(11.2\)](#).

Attention: This document uses UNIX/Linux syntax when describing directory structures. However, it applies to Windows servers as well. Where there is a significant difference in tasks for Windows, specific instructions are given.

Some of the tasks in this document affect the APPL_TOP of one or more application server tiers. Those tasks require that the Applications file system environment be enabled by running the APPSORA.env file (for UNIX or Linux) or the envshell.cmd file (for Windows) prior to performing the tasks. Other tasks affect the Applications database instance. Those tasks require that the Oracle 11g environment be enabled by running the .env/cmd file under the Oracle 11g Oracle home on the database server node prior to performing the tasks. In addition, you may have more than one Oracle home installed on the database server node, so it is important that you run the correct .env/cmd file before performing tasks that affect the database instance. Read the instructions carefully to determine which environment should be enabled for each step.

Attention: This document assumes that the source and target application server tiers are the same. To create new application server tiers for the target environment and to migrate the database server tier through export/import at the same time, perform the steps in [Platform Migration with Oracle Applications Release 12](#) either before starting or after completing all the steps in this document. Then, update and run AutoConfig for the source database and application server tiers to enable the source environment.

Section 4: Prepare the source system

This section describes how to ensure that you have the required patches, create your export file, and capture important information that is required to import your database.

1. Perform prerequisite steps

Perform the steps in the "Before the Database Installation" subsection of Section 1 of the [Oracle E-Business Suite Release 12 with Oracle Database 11g Release 1 \(11.1.0\) Interoperability Notes](#), [Oracle E-Business Suite Release 12.1 with Oracle Database 11g Release 1 \(11.1.0\) Interoperability Notes](#), or [Oracle E-Business Suite Release 12 with Oracle Database 11g Release 2 \(11.2.0\) Interoperability Notes](#). Do not export the OLAP analytical workspaces.

Attention: Verify that you have at least 1.5 GB of free SYSTEM tablespace.

2. Apply latest AutoConfig patches

Perform step 3 in Section 2 and steps 1 and 2 in Section 6 of the [Using AutoConfig to Manage System Configurations in Oracle E-Business Suite Release 12](#) document.

3. Apply the Applications consolidated export/import utility patch

Apply Patch [7120092](#) to the source administration server node. This patch provides several SQL scripts that facilitate exporting and importing an Applications database instance.

4. Create a working directory

Create a working directory named expimp in the source system that will contain all generated files and scripts required to complete this section. As an example,

```
$ mkdir /u01/expimp
```

5. Generate target database instance creation script aucrdb.sql

The target database instance must be created with the same tablespace structure as the source database instance. The export/import patch provides the auclondb.sql script that generates the aucrdb.sql script, which you use to create the target database instance with the appropriate tablespace and file structure.

On the Source Administration Server node, use SQL*Plus to connect to the database as SYSTEM and run the \$AU_TOP/patch/115/sql/auclondb.sql script. It creates aucrdb.sql in the current directory.

```
$ cd <working_directory>
sqlplus system/ <system password>
```

6. Record Advanced Queue settings

Advanced Queue settings are not propagated in the target database instance during the export/import process. Therefore, you must record them beforehand and enable them in the target database instance afterwards. The export/import patch contains auque1.sql, which generates a script called auque2.sql. You can use auque2.sql to enable the settings in the target database instance.

Copy the auque1.sql script from the \$AU_TOP/patch/115/sql directory on the source administration server node to the working directory in the source database server node. Then, on the source database server node, as the owner of the source database server file system and database instance, use SQL*Plus to connect to the source database as sysdba and run the auque1.sql script. It generates auque2.sql.

```
$ sqlplus /nolog
SQL> connect / as sysdba;
SQL> @auque1.sql
```

7. Create parameter file for tables with long columns (conditional)

If the source database is Oracle Database 10g Release 2, tables with long columns may not propagate properly in datapump. Therefore, they have to be migrated separately using the traditional export/import utilities.

Copy the aulong.sql script from the \$AU_TOP/patch/115/sql directory on the source administration server node to the working directory in the source database server node. Then, on the source database server node, as the owner of the source database server file system and database instance, use SQL*Plus to connect to the source database as sysdba and run the aulong.sql script. It generates aulongexp.dat.

```
$ sqlplus /nolog
SQL> connect system/<system password>;
SQL> @aulong.sql
```

8. Remove rebuild index parameter in spatial indexes

Ensure that you do not have the rebuild index parameter in the spatial indexes. To see if you have any rebuild index parameters, on the source database server node, as the owner of the source database server file system and database instance, use SQL*Plus to connect to the source database as sysdba and run the following command:

```
SQL> select owner, index_name, parameters from dba_indexes where index_type='DOMAIN' and upper(parameters) like '%REBUILD%';
```

To remove the rebuild index parameter, use SQL*Plus to connect to the source database as the owner of the index and run the following command:

```
SQL> alter index <index name> rebuild parameters <parameters> ;
```

where <parameters> is the original parameter set without the rebuild_index parameter.

For example:

If the results from the query above are:

OWNER	INDEX_NAME	PARAMETERS
CSF	CSF_MD_ADM_BNDS_N1	rebuild_index=MDRT_32B8A\$ sdo_indx_dims=2 sdo_rtr_pctfree=10 tablespace=APPS_TS_TX_IDX

The command to remove the rebuild index would be:

(After connecting to SQL*Plus as the CST user)

```
SQL> alter index CSF_MD_ADM_BNDS_N1 rebuild parameters ('sdo_indx_dims=2 sdo_rtr_pctfree=10 tablespace=APPS_TS_TX_IDX');
```

9. Deregister the current database server (conditional)

If you plan to change the database port, host, SID, or database name parameter on the database server, deregister the current database server node by running the following command as the owner of the Oracle RDBMS file system and current database instance:

```
$ perl $ORACLE_HOME/appsutil/bin/adgentns.pl appspass=apps contextfile=$CONTEXT_FILE -removeserver
```

Section 5: Prepare a target Applications Release 12 database instance

This section describes how to create the empty target database and populate it with all of the required system objects prior to running import.

The Oracle home of the target database instance can be the same Oracle home that the source database instance uses, or it can be different (on another machine running a different operating system, for example), as long as it uses Oracle Database 11g Release 1 Enterprise Edition.

1. Create target Oracle 11g Oracle home (conditional)

If you want the target Oracle 11g Oracle home to be separate from the source Oracle home, you must create it now. Perform the steps in the "Database Installation" subsection of Section 1 of the [Oracle E-Business Suite Release 12 with Oracle Database 11g Release 1 \(11.1.0\) Interoperability Notes](#), [Oracle E-Business Suite Release 12.1 with Oracle Database 11g Release 1 \(11.1.0\) Interoperability Notes](#), or [Oracle E-Business Suite Release 12 with Oracle Database 11g Release 2 \(11.2.0\) Interoperability Notes](#).

2. Enable Oracle Advanced Security TDE Tablespace Encryption

With Oracle 11g 11.1.0.7 and above, The master key can either be stored in an encrypted Oracle Wallet, or a Hardware Security Module (HSM).

Using an Encrypted Oracle Wallet:

The Oracle Wallet is an encrypted file outside of the database. Please read the [Transparent Data Encryption Best Practices Document](#) to familiarize you with TDE. Make sure the directory given below exists, and has the proper access rights for 'oracle' to read and write. Add the following lines to the sqlnet_ifile.ora file in \$ORACLE_HOME/network/admin/ <SID>_<HOST>

(If the sqlnet_ifile.ora does not exist, create one with the same permissions as the sqlnet.ora file and ensure that \$ORACLE_HOME/network/admin/<SID>_<HOST>/sqlnet.ora correctly references sqlnet_ifile.ora by adding the following line to sqlnet.ora:

IFILE=<full path of sqlnet_ifile.ora> :

Add this entry to the sqlnet_ifile.ora in \$ORACLE_HOME/network/admin/<SID>_<HOST>:

```
ENCRYPTION_WALLET_LOCATION =
(SOURCE = (METHOD = FILE)
(METHOD_DATA =
(DIRECTORY = /etc/ORACLE/WALLETS/oracle)))
```

Using a Hardware Security Module (HSM):

If HSM is to be used, follow the instructions of the HSM vendor in terms of HSM configuration and software installation. Again, the [Transparent Data Encryption Best Practices Document](#) contains many hints and details about master encryption key management with Oracle Wallet and/or HSM devices. For HSM mode, sqlnet_ifile.ora in \$ORACLE_HOME/network/admin/<SID>_<HOST> needs the following entry:

(If the sqlnet_ifile.ora does not exist, create one with the same permissions as the sqlnet.ora file in the \$ORACLE_HOME/network/admin/<SID>_<HOST> directory and ensure that the \$ORACLE_HOME/network/admin/<SID>_<HOST>/sqlnet.ora file correctly references sqlnet_ifile.ora, by adding the following line to sqlnet.ora:

IFILE=<full path of sqlnet_ifile.ora>

Add this entry to the sqlnet_ifile.ora in \$ORACLE_HOME/network/admin/<SID>_<HOST>:

```
ENCRYPTION_WALLET_LOCATION =
(SOURCE = (METHOD = HSM))
```

When starting with a fresh installation of Oracle 11g 11.1.0.7, it is mandatory not to create a software wallet. Instead you should first set sqlnet_ifile.ora to 'METHOD=HSM' and then create the master encryption key (in the HSM).

Please see the [Transparent Data Encryption Best Practices Document](#) for further details.

Attention: You must apply Patch [7563307](#) when using HSM mode.
For Windows customers, Rollup Patch 13 for 11.1.0.7 includes this patch.

3. Modify sqlnet.ora file (Windows only)

If the target database server node is running Windows, add the following line to the sqlnet.ora file in the %ORACLE_HOME%\network\admin\ directory, if it does not already exist:

```
SQLNET.AUTHENTICATION_SERVICES=(NTS)
```

4. Create the target initialization parameter file and CBO parameter file

The initialization parameter file (init.ora) and cost-based optimizer (CBO) parameter file (ifilecbo.ora) are located in the \$ORACLE_HOME/dbs directory on the source database server node. Copy both files to the Oracle 11g \$ORACLE_HOME/dbs directory on the target database server node.

Refer to [Database Initialization Parameters for Oracle Applications Release 12](#) and update both the init.ora and ifilecbo.ora files with any necessary changes. You may also need to update initialization parameters involving the db_name, control_files, and directory structures.

Ensure that the undo_tablespace parameter in the initialization parameter file of the target database instance matches with the default undo tablespace set in the aucrdb.sql script.

Ignore the initialization parameters that pertain to the native compilation of PL/SQL code. You will be instructed to add them later, if necessary.

5. Create a working directory

Create a working directory named expimp in the target system that will contain all generated files and scripts required to complete this section. As an example,

```
$ mkdir /u01/expimp
```

6. Modify the aucrdb.sql script

Copy the aucrdb.sql script, generated in Section 4, from the source administration server node to the working directory in the target database server node. Then update the script on the target database server node with any necessary changes to the directory structures for the log file(s), data file(s), or tablespaces, reflecting the layout of the target database server node. If the target database server node is running Windows, update the directory structure from UNIX/Linux format to Windows format.

Append the following to each 'CREATE TABLESPACE' command in aucrdb.sql:

```
ENCRYPTION [USING '<enc. algorithm>'] DEFAULT STORAGE (ENCRYPT)
```

No other parameters need to be modified, not even the 'size' parameter. All tablespaces with this additional command will store all content encrypted, protecting your application data on disk and backup tape through encryption. For example:

```
ENCRYPTION DEFAULT STORAGE (ENCRYPT)
to apply the default encryption algorithm (AES128) to the tablespace
or
ENCRYPTION USING 'AES256' DEFAULT STORAGE (ENCRYPT)
to apply another encryption algorithm to the tablespace
(AES256 in this example, possible choices are: 3DES168, AES128 (default if none specified), AES192, and AES256).
```

Finally split the edited copy of aucrdb.sql into two separate .sql files, the first will contain the CREATE DATABASE statement and the second will contain the CREATE TABLESPACE statements.

1. Save the lines from the top of aucrdb.sql to the line "REM Create Tablespaces" as aucrdb1.sql
2. Save the lines from the line "REM Create Tablespaces" to the end of aucrdb.sql as aucrdb2.sql.

The shell commands below will perform the above split:

```
$ sed -n '1,/^REM Create Tablespaces/ p' aucrdb.sql > aucrdb1.sql
$ sed -n '/^REM Create Tablespaces/, $ p' aucrdb.sql > aucrdb2.sql
```

7. Create the target database instance

Make sure that the environment of your session on the target database server node is set up properly for the target database instance, especially the ORACLE_HOME, ORACLE_SID, and ORA_NLS10 environment settings. (ORACLE_SID must be set to the same value as the db_name parameter in the init.ora file.) Then, use the following commands to run aucrdb.sql and create the target database instance:

```
$ sqlplus /nolog
```

```
SQL> connect / as sysdba;
SQL> spool aucrdb1.log;
```

For UNIX or Linux:

```
SQL> startup nomount;
```

For Windows:

```
SQL> startup nomount pfile=%ORACLE_HOME%\dbs\init%ORACLE_SID%.ora
```

For any platform create the database by running aucrdb1.sql

```
SQL> @aucrdb1.sql
SQL> spool off
```

After the database is created the encryption key must be created prior to the creation of the encrypted tablespaces. This step is performed by the "key custodian" which may be different from the DBA performing the other steps of this procedure. This depends on your company's key management procedures.

For encrypted tablespaces and encryption key management based on Oracle Wallet:

```
SQL> alter system set encryption key identified by "<strong_password>" ;
```

For encrypted tablespaces and encryption key management based on HSM:

```
SQL> alter system set encryption key identified by "<HSM_username>:<HSM_password>" ;
```

Create the tablespaces by running aucrdb2.sql

```
SQL> spool aucrdb2.log
SQL> @aucrdb2.sql
SQL> exit;
```

If PL/SQL of the source database was natively compiled, see the "Compiling PL/SQL Code for Native Execution" section of Chapter 11 of [Oracle Database PL/SQL User's Guide and Reference 11g Release 1 \(11.1\)](#) or Chapter 12 of [Oracle Database PL/SQL Language Reference 11g Release 2 \(11.2\)](#) for instructions on how to natively compile PL/SQL in the target database. Add the parameters that pertain to the native compilation where specified. Do not use the natively compiled code generated by the source database. Oracle does not support switching the PL/SQL compilation mode from interpreted to native (and vice-versa) for an export/import. Exporting/importing using native mode takes significantly more time than interpreted mode.

When the target database instance has been created, restart the database instance.

When encrypted tablespaces are used, the wallet needs to be opened every time the database starts. A user with the 'ALTER SYSTEM' privilege and knowledge of the wallet credential must open the wallet. The 'alter system set encryption wallet open' statement must be used between "startup mount" and "alter database open" statements of the startup sequence.

If an Oracle Wallet is used:

```
SQL> alter system set encryption wallet open identified by "<strong_password>" ;
```

or the HSM username/password, if an HSM is used:

```
SQL> alter system set encryption wallet open identified by "<HSM_username>:<HSM_password>" ;
```

Attention: After enabling Tablespace Encryption the wallet needs to be opened every time the database is started, i.e.:

```
SQL> startup mount;
SQL> alter system set encryption wallet open identified by "<strong_password>" ; (for Wallet mode encryption)
or
SQL> alter system set encryption wallet open identified by "<HSM_username>:<HSM_password>" ; (for HSM mode encryption)
SQL> alter database open;
```

An alternative is to create an auto-open wallet, if this satisfies your security requirements. Please see the [Transparent Data Encryption Best Practices Document](#) for further details.

Note: You may modify the sizes of the database files in aucrdb1.sql and aucrdb2.sql to ensure enough tablespace. Querying the dba_free_space and dba_data_files tables in the source database can give you an idea of how much tablespace is required. Power limits are guaranteed, but the space requirements may change depending on the extent sizes used by each object. Not having enough tablespace will cause failures as well as major performance degradation in the import run.

The export/import patch that you applied to the source administration server node in Section 4 contains four scripts that are needed on the target database server node. Copy the following files from the \$AU_TOP/patch/115/sql directory of the source administration server node to the working directory in the target database server node:

- audb1110.sql/audb1120.sql, ausy1110.sql/ausy1120.sql, aujv1110.sql/aujv1120.sql, and aumsc1110.sql/aumsc1120.sql (UNIX or Linux)
- or
- audb1110_nt.sql/audb1120_nt.sql, ausy1110_nt.sql/ausy1120_nt.sql, aujv1110_nt.sql/aujv1120_nt.sql, and aumsc1110_nt.sql/aumsc1120_nt.sql (Windows).

As you run each of the next four steps, note the following:

- a. The remarks section at the beginning of each script contains additional information.
- b. Each script creates a log file in the working directory.

9. Set up the SYS schema

The audb1110.sql, audb1120.sql, audb1110_nt.sql, or audb1120_nt.sql script sets up the SYS schema for use with the Applications. On the target database server node, use SQL*Plus to connect to the target database instance as SYSDBA and run audb1110.sql/audb1120.sql (UNIX/Linux) or audb1110_nt.sql/audb1120_nt.sql (Windows).

Here is an example on UNIX or Linux for 11.2.0:

```
$ sqlplus "/ as sysdba" @/u01/expimp/audb1120.sql
```

10. Set up the SYSTEM schema

The ausy1110.sql, ausy1120.sql, ausy1110_nt.sql, or ausy1120_nt.sql script sets up the SYSTEM schema for use with the Applications. On the target database server node, use SQL*Plus to connect to the target database instance as SYSTEM and run ausy1110.sql/ausy1120.sql (UNIX/Linux) or ausy1110_nt.sql/ausy1120_nt.sql (Windows).

Here is an example on UNIX or Linux for 11.2.0:

```
$ sqlplus system/<system password> @/u01/expimp/ausy1120.sql
```

11. Install Java Virtual Machine

The aujv1110.sql, aujv1120.sql, aujv1110_nt.sql, or aujv1120_nt.sql script installs the Java Virtual Machine (JVM) in the database. On the target database server node, use SQL*Plus to connect to the target database instance as SYSTEM and run aujv1110.sql/aujv1120.sql (UNIX/Linux) or aujv1110_nt.sql/aujv1120_nt.sql (Windows).

Here is an example on UNIX or Linux for 11.2.0:

```
$ sqlplus system/<system password> /u01/expimp/@aujv1120.sql
```

Attention: This script can be run only once in a given database instance, because the scripts that it calls are not rerunnable.

12. Install other required components

The aumsc1110.sql, aumsc1120.sql, aumsc1110_nt.sql, or aumsc1120_nt.sql script installs the following required components in the database: ORD, Spatial, XDB, OLAP, Data Mining, *interMedia*, and ConText. On the target database server node, use SQL*Plus to connect to the target database instance as SYSTEM and run aumsc1110.sql/aumsc1120.sql (UNIX/Linux) or aumsc1110_nt.sql/aumsc1120_nt.sql (Windows). You must pass the following arguments to the script, in the order specified:

Argument	Value
remove context?	FALSE
SYSAUX tablespace	SYSAUX
temporary tablespace	TEMP

Here is an example on UNIX or Linux for 11.2.0:

```
SQL> $ sqlplus system/<system password> @aumsc1120.sql FALSE SYSAUX TEMP
```

Attention: All of the components are created in the SYSAUX tablespace regardless of where it was installed in the source database.

13. Set CTXSYS parameter (conditional)

If your target database is 11g Release 2 (11.2.0), use SQL*Plus to connect to the database as SYSDBA and run the following command:

```
$ sqlplus "/ as sysdba"
SQL> exec ctxsys.ctx_adm.set_parameter('file_access_role', 'public');
```

14. Disable automatic gathering of statistics

Copy \$APPL_TOP/admin/adstats.sql from the administration server node to the target database server node. Use SQL*Plus to connect to the database as SYSDBA and use the following commands to restart the database in restricted mode and run adstats.sql

```
$ sqlplus "/ as sysdba"
SQL> shutdown normal;
SQL> startup restrict;
SQL> @adstats.sql
SQL> exit;
```

15. Back up the target database instance

The target database instance is now prepared for an import of the Applications data. You should perform a backup before starting the import.

Before backing up the .dbf files, shutdown the database:

```
SQL> shutdown normal;
SQL> exit;
```

Backup the .dbf files

Startup the database normally, ensuring that the wallet is open:

```
$ sqlplus "/ as sysdba"
SQL> startup mount;
SQL> alter system set encryption wallet open identified by "<strong_password>" ;
SQL> alter database open;
SQL> exit;
```

Section 6: Export the source Applications Release 12 database instance

This section describes how to ensure that you have the required patches, create your export file, and capture important information that is required to import your database.

1. Create the export parameter file

A template for the export parameter file has been included as part of the export/import patch. Copy \$AU_TOP/patch/115/import/auexpdp.dat from the source administration server node to the working directory in the source database server node. Use a text editor to modify the file to reflect the source environment and other customized parameters.

The customizable parameters are:

Parameter	Description	Template Value
directory	directory where the export dump files will be created	dmpdir
dumpfile	export dump file name(s)	aexp%U.dmp
filesize	export dump file size	1GB
log	log file name	expdpapps.log

interMedia, OLAP, and Data Mining schemas are not exported. The aumsc1110.sql/aumsc1120.sql script creates these schemas in the target database. Ensure that the schema names in the exclude parameters reflect those in your database.

If your source database is Oracle Database 11g (11.x), add the following line to the parameter file:

```
QUERY=appls.wf_item_attribute_values:'where item_type!='WFERROR' and name != 'EVENT_MESSAGE'
```

Create a directory in the system schema that corresponds to the directory specified in the template. Here is an example of how to create a directory named dmpdir:

```
$ sqlplus system/<system password>
SQL> create directory dmpdir as '/u01/expimp';
```

The export process uses as many of the listed file names as necessary to hold the exported data. You must ensure that the number of dump files specified, as well as the size of each dump file, is sufficient to contain all the data in your source database instance.

2. Shut down Applications server processes

Shut down all Applications server processes except the database and the database listener. Users cannot use the Applications until the import is completed.

3. Grant privilege to source system schema

Grant the exempt access policy privilege to system by using SQL*Plus to connect to the database as SYSDBA and run the following command:

```
SQL> grant EXEMPT ACCESS POLICY to system;
```

4. Export the Applications database instance

Start an export session on the source database server node using the customized export parameter file.

If the source database is 10.2.0 or 11.1.0, use the following command:

```
$ expdp system/<system password> parfile=<export parameter file name>
```

If the source database is 11.2.0, use the following command:

```
$ expdp '/' as sysdba" parfile=<export parameter file name>
```

Typically, the export runs for several hours.

5. Export tables with long columns (conditional)

If the source database is Oracle Database 10g Release 2 (10.2.0), start an export session on the source database server node using the customized aulongexp.dat file generated in [Section 4](#). Use the following command:

```
$ exp parfile=aulongexp.dat
```

6. Export tables with XML type columns (conditional)

If the source database is Oracle Database 10g Release 2 (10.2.0), copy \$AU_TOP/patch/115/import/auxmlexp.dat from the source administration server to the working directory in the source database server node. Start an export session on the source database server node using the following command:

```
$ exp parfile=auxmlexp.dat
```

7. Revoke privilege from source system schema

Revoke the exempt access policy privilege from system by using SQL*Plus to connect to the database as SYSDBA and run the following command:

```
SQL> revoke EXEMPT ACCESS POLICY from system;
```

Section 7: Import the Applications Release 12 database instance

This section describes how to use the import utility to load the Oracle Applications data into the target database.

1. Create the import parameter files

Copy aimpdp.dat, afullimp.dat, and aimpusr.dat from the \$AU_TOP/patch/115/import directory in the source administration server node to the working directory in the target database server node. Make sure that the directory, dumpfile, and logfile parameters in aimpdp.dat and aimpusr.dat are set properly.

Create a directory in the system schema with the name set to the directory specified in the template and the path set to where the export dump files will reside. Here is an example of how to create a directory named dmpdir:

```
$ sqlplus system/<system password>
SQL> create directory dmpdir as '/u01/expimp';
```

2. Copy the export dump files

Copy the export dump files from the source database server node to the working directory in the target database server node (or create a network mount).

3. Import the users into the target database (conditional)

If the source database is Oracle Database 10g Release 2 (10.2.0), start an import session on the target database server node using the customized import parameter file. Use the following command:

```
$ impdp system/<system password> parfile=aimpusr.dat
```

4. Import tables with long columns into the target database (conditional)

If the source database is Oracle Database 10g Release 2 (10.2.0), modify the afullimp.dat file with the following:

- Set userid to "sys/ as sysdba".
- Set file to the dump file containing the long tables (longexp by default).
- Set the log file appropriately.
- Leave the ignore parameter commented out.

Import the tables using the following command:

```
$ imp parfile=aufullimp.dat
```

Attention: You will get failures for the triggers as the dependent tables have not yet been imported.

5. Import the Applications database instance

If your source database is Oracle Database 11g (11.x), remove or comment out all the exclude parameters in the auimpdp.dat parameter file. If your source database is Oracle Database 10g Release 2 (10.2.0), leave the parameter file as is. Start an import session on the target database server node using the auimpdp.dat parameter file. Start an import session on the target database server node using the auimpdp.dat parameter file.

If the target database is 11.1.0, use the following command:

```
$ impdp system/<system password> parfile=auimpdp.dat
```

If the target database is 11.2.0, use the following command:

```
$ impdp '/' as sysdba" parfile=auimpdp.dat
```

Typically, import runs for several hours.

6. Import triggers into the target database (conditional)

If the source database is Oracle Database 10g Release 2 (10.2.0), modify the afullimp.dat file with the following:

- Set userid to "sys/ as sysdba".
- Set file to the dump file containing the long tables (longexp by default).
- Change the log file name.
- Uncomment the ignore parameter.
- Add a line with the parameter "rows=n".

Start an import session on the target database server node using the customized import parameter file. Use the following command:

```
$ imp parfile=aufullimp.dat
```

Once the import is complete, you can delete the export dump files, as well as the export and import parameter files, from the source and target database server nodes.

Section 8: Update the imported Applications Release 12 database instance

This section describes how to recreate the database objects and relationships that are not handled by the export and import utilities.

1. Reset Advanced Queues

Copy the auque2.sql script that was generated in [Section 4](#) from the working directory in the source database server node to the working directory in the target database server node. Then, on the target database server node, as the owner of the Oracle 11g file system and database instance, use SQL*Plus to connect to the target database as SYSDBA and run the auque2.sql script to enable the Advanced Queue settings that were lost during the export/import process. The script creates a log file in the current directory.

```
$ sqlplus /nolog
SQL> connect / as sysdba;
SQL> @auque2.sql
```

2. Start the new database listener (conditional)

If the Oracle Net listener for the database instance in the new Oracle home has not been started, you must start it now. Since AutoConfig has not yet been implemented, start the listener with the lsnrctl executable (UNIX/Linux) or Services (Windows). See the [Oracle Database Net Services Administrator's Guide, 11g Release 1 \(11.1\)](#) or [Oracle Database Net Services Administrator's Guide, 11g Release 2 \(11.2\)](#) for more information.

Attention: Set the TNS_ADMIN environment variable to the directory where you created your listener.ora and tnsnames.ora files.

3. Run adgrants.sql

Copy \$APPL_TOP/admin/adgrants.sql (adgrants_nt.sql for Windows) from the administration server node to the working directory in the database server node. Use SQL*Plus to connect to the database as SYSDBA and run the script using the following command:

```
$ sqlplus "/ as sysdba" @adgrants.sql (or adgrants_nt.sql) <APPS Schema Name>
```

4. Grant create procedure privilege on CTXSYS

Copy \$AD_TOP/patch/115/sql/adctxprv.sql from the administration server node to the database server node. Use SQL*Plus to connect to the database as APPS and run the script using the following command:

```
$ sqlplus apps/<APPS password>
```

5. Apply patch 6494466 (conditional)

If the target database is Windows and the source is not, apply <Patch 6494466> on the target database tier. Create the appsutil directory if needed.

6. Implement and run AutoConfig

Implement and run AutoConfig in the new Oracle home on the database server node. If the database listener of the new Oracle home is defined differently than the old Oracle home, you must also run AutoConfig on each application tier server node to update the system with the new listener.

See [Using AutoConfig to Manage System Configurations in Oracle E-Business Suite Release 12](#) on My Oracle Support , especially sections 3 and 6, for instructions on how to implement and run AutoConfig.

Shut down all processes, including the database and the listener, and restart them to load the new environment settings.

7. Gather statistics for SYS schema

Use SQL*Plus to connect to the database as SYSDBA and use the following commands to restart the database in restricted mode, run adstats.sql, and restart the database in normal mode:

```
$ sqlplus "/ as sysdba"
SQL> alter system enable restricted session;
SQL> @adstats.sql
(the script completes successfully and exits from sql*plus)

$ sqlplus "/ as sysdba"
SQL> alter system disable restricted session;
SQL> exit;
```

Attention: Make sure that you have at least 1.5 GB of free default temporary tablespace.

8. Re-create custom database links (conditional)

If the Oracle Net listener in the 11g Oracle home is defined differently than the one used by the old Oracle home, you must re-create any custom self-referential database links that exist in the Applications database instance. To check for the existence of database links, use SQL*Plus on the database server node to connect to the Applications database instance as APPS and run the following query:

```
$ sqlplus apps/<apps password>
SQL> select db_link from dba_db_links;
```

The EDW_APPS_TO_WH and APPS_TO_APPS database links, if they exist, should have been updated with the new port number by AutoConfig in the previous step.

If you have custom self-referential database links in the database instance, use the following commands to drop and re-create them:

```
$ sqlplus apps/<apps password>

SQL> drop database link <custom database link>;
SQL> create database link <custom database link> connect to <user> identified by <password> using '(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<hostname>) (PORT=<port number>)))(CONNECT_DATA=(SID=<ORACLE_SID>)))';
```

where <custom database link>, <user>, <password>, <hostname>, <port number>, and <ORACLE_SID> reflect the new Oracle Net listener for the database instance.

9. Create ConText and AZ objects

Certain ConText objects and the AZ objects dependent on the tables with XML type columns are not preserved by the import process. The consolidated export/import utility patch that you applied to the source administration server node in [Section 4](#) contains a perl script, dpost_imp.pl, that you can run to generate an AutoPatch driver file. You use this driver file to call the scripts that create these objects. Run the following command:

```
$ perl $AU_TOP/patch/115/bin/dpost_imp.pl <driver file> <source database version>
```

Where

<driver file> is any name you choose for the driver

<source database version> is set to 10 if the source database is 10.2 or to 11 if the source database is 11g.

Once the driver file has been generated, use AutoPatch to apply it on the target administration server node.

10. Import tables with XML type columns into the target database (conditional)

If the source database is Oracle Database 10g Release 2, modify the aufullimp.dat file with the following:

- Set userid to "az/<az password>"
- Set file to the dump file containing the tables with XML types (xmlexp by default).
- Change the log file name.
- Comment out the ignore parameter.

Start an import session on the target database server node using the customized import parameter file. Use the following command:

```
$ imp parfile=aufullimp.dat
```

Once the import is complete, you can delete the export dump files, as well as the export and import parameter files, from the source and target database server nodes.

11. Populate CTXSYS.DR\$SQE table

To populate the CTXSYS.DR\$SQE table, use SQL*Plus on the database server node to connect to the Applications database instance as APPS and run the following command:

```
$ sqlplus apps/<apps password>
SQL> exec icx_cat_sqe_pvt.sync_sqes_for_all_zones;
```

12. Compile invalid objects

On the target database server node, as the owner of the Oracle 10g file system and database instance, use SQL*Plus to connect to the target database as SYS and run the \$ORACLE_HOME/rdbms/admin/utlirp.sql script to compile invalid objects.

```
$ sqlplus "/ as sysdba" @$ORACLE_HOME/rdbms/admin/utlirp.sql
```

13. Maintain Applications database objects

Run AD Administration on the target administration server node. From the Maintain Applications Database Objects menu, perform the following tasks:

- Compile flexfield data in AOL tables
- Recreate grants and synonyms for APPS schema

14. Start Applications server processes

Start all the server processes on the target Applications system. You can allow users to access the system at this time.

15. Create DQM indexes

Create DQM indexes by following these steps:

- Log on to Oracle Applications with the "Trading Community Manager" responsibility
- Click Control > Request > Run
- Select "Single Request" option
- Enter "DQM Staging Program" name
- Enter the following parameters:
 - Number of Parallel Staging Workers: 4
 - Staging Command: CREATE_INDEXES

■ Continue Previous Execution: NO

■ Index Creation: SERIAL

f. Click "Submit"

References

- [Database Security - Transparent Data Encryption FAQ](#) from OTN
- [Oracle Advanced Security - Transparent Data Encryption Best Practices](#) from OTN
- [Oracle Database Advanced Security Administrator's Guide 10g Release 2 \(10.2\)](#) (Part No. B14268-02), Chapter 3
- [Oracle Database Error Messages 10g Release 2 \(10.2\)](#) (Part No. B14219-01)
- [Oracle Database Advanced Security Administrator's Guide 11g Release 1 \(11.1\)](#) (Part No. B28530-03), Chapter 3
- [Oracle Database Error Messages 11g Release 1 \(11.1\)](#) (Part No. B28278-02)
- [Oracle Database Advanced Security Administrator's Guide 11g Release 2 \(11.2\)](#) (Part Number E10746-02), Chapter 3
- [Oracle Database Error Messages 11g Release 2 \(11.2\)](#) (Part Number E10880-02)
- [Using TDE Column Encryption with Oracle E-Business Suite Release 12.](#)
- [Oracle E-Business Suite Release 12.0 with Oracle Database 11g Release 1 \(11.1.0\) Interoperability Notes.](#)
- [Oracle E-Business Suite Release 12.1 with Oracle Database 11g Release 1 \(11.1.0\) Interoperability Notes.](#)
- [Oracle E-Business Suite Release 12 with Oracle Database 11g Release 2 \(11.2.0\) Interoperability Notes.](#)
- [Database Initialization Parameters \(init.ora settings\) in Oracle Applications Release 12](#)
- [Oracle Database PL/SQL User's Guide and Reference 11g Release 1 \(11.1\)](#)
- [Oracle Database PL/SQL User's Guide and Reference 11g Release 2 \(11.2\)](#)
- [Oracle Database Utilities 11g Release 1 \(11.1\)](#)
- [Oracle Database Utilities 11g Release 2 \(11.2\)](#)

Change Log

Date	Description
May 21, 2009	• New document created, based on Export/Import Note for Release 12 (741818.1) and TDE Tablespace Encryption Note (828223.1) for Release 11i.
May 27, 2009	• Added attention box for Database Vault.
May 28, 2009	• Modified URLs for patches and notes.
June 2, 2009	• Modified attention box for Database Vault to point to My Oracle Support Knowledge Document 822048.1
June 23, 2009	• Removed references to 11.1.0.6.
May 18 , 2010	• Modified document to include 11.2.0 steps • Changed export/import patch to 7120092

My Oracle Support Knowledge Document <828229.1> by Oracle E-Business Suite Development
Copyright ? 2008, 2009 Oracle.

[Back to top](#)

Copyright (c) 2007, 2010, Oracle. All rights reserved. Legal Notices and Terms of Use | Privacy Statement

This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.