

Migrating the Oracle E-Business
Suite Database to Oracle Exadata
Database Machine Using
Transportable Tablespaces

*Oracle Maximum Availability Architecture White Paper
February 2012*

Maximum Availability Architecture

Oracle Best Practices For High Availability

Executive Overview	2
Document Conventions	3
Introduction	4
Preparation.....	5
Using Transportable Tablespaces	6
Prerequisites	7
Known Issues & Limitations.....	7
Preparing the Source-system	8
Preparing the Target Database	12
Exporting the Source Database	20
Importing	26
Updating the Imported Database	30
Further Configuration.....	38
Set Up Parallel Concurrent Processing.....	38
Set Up Load Balancing	38
Conclusion	38
Appendix	39
Test Environment Details.....	39
Target Database Starting Parameter File	41
Initial listener.ora Files	42
Timestamp Script.....	43
Generated SQL Files	43
Data Pump Parameter Files	46
Other SQL Files and Scripts	51
References.....	54
Change Record	55

Executive Overview

Oracle Maximum Availability Architecture (MAA) [1] is Oracle's best practices blueprint for implementing Oracle high-availability technologies. Oracle Exadata Database Machine provides an optimal platform for all database workloads. This paper describes how to use Oracle cross-platform transportable tablespaces to migrate your Oracle E-Business Suite database to Oracle Exadata Database Machine.

There are several techniques for migrating the E-Business Suite database to Oracle Exadata Database Machine, as outlined in My Oracle Support (MOS) ID [1133355.1](#), “*A Roadmap for Migrating Oracle E-Business Suite to Oracle Exadata Database Machine*”. This paper illustrates the steps required to migrate an Oracle database 11g release 2 E-Business Suite database to Oracle Exadata Database Machine using Cross-platform Transportable Tablespaces (XTTS). Currently, a source system with an Oracle Database 11g release 1 (11.1.0.7.6) or 11g release 2 E-Business Suite database is supported with XTTS. Check back for updates.

This example was done with Oracle E-Business Suite release 12. The XTTS migration option to migrate an E-Business Suite database to Oracle Exadata Database Machine is typically used when the source system is using the big endian (byte ordering) format, and using Data Pump export/import does not meet the availability service level requirement. The more common big endian platforms are:

- AIX-Based Systems (64-bit)
- HP-UX (64-bit)
- HP-UX IA (64-bit)
- Solaris[tm] OE (32-bit and 64-bit)

Oracle Exadata Database Machine uses Linux x86-64 bit and is little endian format.

This paper is part of a series of Oracle Maximum Availability Architecture (MAA) [1] white papers. MAA is Oracle's best practices blueprint for implementing Oracle high-availability technologies.

Document Conventions

CONVENTION	MEANING
Application Tier	Machines running Forms, Web, Concurrent Processing and other servers. Sometimes called middle tier.
Database Tier	Machines running an Oracle E-Business Suite database.
oracle	User account that owns the database file system (database ORACLE_HOME and files).
CONTEXT_NAME	The CONTEXT_NAME variable specifies the name of the Applications context that is used by Autoconfig. The default is <SID>_<short hostname>.
CONTEXT_FILE	Full path to the Applications context file on the application tier or database tier. The default locations are as follows. Application tier context file: <INST_TOP>/appl/admin/CONTEXT_NAME.xml Database tier context file: <Database ORACLE_HOME>/appsutil/<CONTEXT_NAME>.xml
Monospace Text	Represents command line text. Type such a command exactly as shown, excluding prompts such as '% '.
< >	Text enclosed in angle brackets represents a variable. Substitute a value for the variable text. Do not type the angle brackets.
\	On UNIX, the backslash character can be entered at the end of a command line to indicate continuation of the command on the next line.

Introduction

As stated, Oracle XTTS is typically used when you are migrating from a platform with a different database endian (byte ordering) format and using Data Pump export/import does not meet the availability service level requirement.

Oracle Exadata Database Machine uses the Linux x86 64-bit operating system which is little endian format. XTTS can also be used to move from an older release of the database to a newer release starting with database release 10.2.0.5 on the source system. MOS ID [1133355.1](#) should be consulted for other migration options and any updates.

This paper illustrates the use of XTTS to move a Release 12 E-Business Suite database running Oracle Database 11g release 2 (11.2) from a Solaris 10 source system (big endian format). This example uses E-Business Suite release 12.1.3 with database release 11.2.0.2 on the source database and the target database machine. This example can be used for any big endian platform. Migrating the Oracle E-Business Suite Release 12 database to Oracle Exadata Database Machine from a source system using a database release prior to Oracle Database Release 1 (11.1) has not been tested. Currently, E-Business Suite releases certified on an Oracle Exadata Database Machine configuration of OEL5 with Oracle Enterprise Edition release 11.2 are:

- 11.5.10.2 (11i)
- 12.0
- 12.1

These references provide the foundational steps for using XTTS with the E-Business Suite to migrate to Oracle Exadata Database Machine:

- [Platform Migration Using Transportable Tablespaces: Oracle Database 11g](#)
- Using Transportable Tablespaces for EBS Release 12 Using Database 11gR2 [ID [1311487.1](#)]
- Using Transportable Tablespaces to Migrate Oracle E-Business Suite Release 11i Using Oracle Database 11g Release 2 Enterprise Edition (ID [1366265.1](#))
- Cross Platform Transportable Tablespaces on 11i with 10gR2 [ID [454574.1](#)]

In addition to the steps outlined in the above references, migrating to Oracle Exadata Database Machine involves additional steps to configure Oracle RAC and Oracle ASM, and implement Oracle Exadata Database Machine best practices. The additional steps are:

- Creating an 11.2 target database that uses Oracle ASM and Oracle RAC with a server parameter file (SPFILE)
- Verifying the required operating system packages

- Verifying that Oracle Exadata Database Machine best practices are in place with the Database Machine Health Check
- Configuring Linux huge pages
- Converting the source system datafiles on the target database machine rather than on the source as [454574.1](#) illustrates.
- Configuring the target system for Oracle E-Business Suite to work in an Oracle RAC environment. For further background see “Using Oracle 11g Release 2 Real Application Clusters with Oracle E-Business Suite Release 12” [ID [823587.1](#)].

The Oracle E-Business Suite Vision database that is part of the Oracle E-Business Suite Rapid Install was used for this example. **The application tier nodes remain the same as part of this example** and are re-directed to the new database on Oracle Exadata Database Machine using the Autoconfig tool in the E-Business Suite.

In this paper, a combination of the steps in MOS ID [454574.1](#) and the MAA paper, [Platform Migration Using Transportable Tablespaces: Oracle Database 11g](#), were followed to migrate a release 11.2 database for an Oracle E-Business Suite release 12 system running on Solaris 10, adding in the above steps where needed. To reiterate, this paper illustrates the steps required to migrate an Oracle database 11g release 2 E-Business Suite database to Oracle Exadata Database Machine using Cross-platform Transportable Tablespaces (XTTS).

Preparation

Preparing properly for the migration will give you the necessary knowledge and confidence to complete a successful migration. While this is a migration, upgrade resources are also referenced as much of their content applies to a migration as well. Create a test plan that includes the best practices outlined in the following MOS articles:

- “*Oracle Database 11g Release 2 Upgrade Companion*,” MOS ID [785351.1](#),
- “*Exadata Patching Overview and Patch Testing Guidelines*,” MOS ID [1262380.1](#)

and the following additional steps:

- Backup the source system and application tier.
- Preserve the source system performance baseline information. See [Upgrade to 11g Performance Best Practices](#) for details.
- Rehearse the migration and fallback procedures
- Stress test with real-life data and workloads

Using Transportable Tablespaces

Using XTTS makes extensive use of Oracle Data Pump. Complete details for Oracle Data Pump can be found in the “[Oracle Database Utilities, 11g Release 2 \(11.2\)](#)” guide. As detailed in [Platform Migration Using Transportable Tablespaces: Oracle Database 11g](#) and MOS ID [454574.1](#), the high-level steps for migrating the Oracle E-Business Suite database to Oracle Exadata Database Machine using XTTS are:

1. [Prerequisites](#)
2. [Known Issues & Limitations](#)
3. [Preparing the Source-system](#)

This step ensures that the required patches are installed, generates the target database creation script, records any advanced queue settings and some other general preparation that is detailed in this section.

4. [Preparing the Target Database Instance](#)

This section describes how to create the empty target database with Oracle RAC and Oracle ASM and populate it with all of the required system objects prior to running import. This section also discusses deciding on using a network file system or using the Oracle Database File System (DBFS) for the export dump files.

5. [Exporting the Source Database](#)

This section describes how to create your export files and capture important information that is required to import your database. This is the step where the outage begins.

6. [Importing](#)

This section describes how to use the Oracle Data Pump import utility to load the Oracle E-Business Suite data into the target database

7. [Updating the Imported Database](#)

This section describes how to re-create the database objects and relationships that are not handled by the Oracle Data Pump export (expdp) and import (impdp) utilities. This section also details re-configuring Oracle E-Business Suite to utilize load balancing for the application connections.

8. [Further Configuration](#)

This section describes how to set up load balancing parallel concurrent processing.

Prerequisites

- **Source Platform Must be Oracle Database 11g Release 2 (11.2.0) or Oracle Database 11g Release 1 (11.1.0.7.6)**

Oracle E-Business Suite systems using an older database release must be upgraded to either Oracle Database 11g Release 1 (11.1.0.7.6) or Oracle Database 11g Release 2 (11.2.0) before you begin the transportable tablespaces process.

- **Determine if the source database platform is supported**

Run this query on the source database:

```
SQL> SELECT D.PLATFORM_NAME, ENDIAN_FORMAT
FROM V$TRANSPORTABLE_PLATFORM TP, V$DATABASE D
WHERE TP.PLATFORM_NAME = D.PLATFORM_NAME;
```

PLATFORM_NAME	ENDIAN_FORMAT

Solaris[tm] OE (64-bit)	Big

- **Resolve User-Owned Tables Residing in the SYSTEM or SYSAUX Tablespaces**

XTTS does not move objects that reside in the SYSTEM or SYSAUX tablespaces of the source database. Run the following script (in the [appendix](#)) identify user objects that reside in SYSTEM or SYSAUX:

```
SQL> @tts_system_user_obj.sql
```

You must move the identified objects to a user tablespace prior to beginning the transport process so the objects can be transported by XTTS. Alternatively, you can move the objects separately with Data Pump or you can manually re-create the objects after performing the platform migration.

Known Issues & Limitations

- **Column Encryption Using TDE**

Transportable Tablespaces migration is not supported for tables that have columns encrypted using the Transparent Data Encryption (TDE) functionality (see MOS ID [828223.1](#) for E-Business Suite 11i and [828229.1](#) for E-Business Suite Release 12).

Tables that have encrypted columns must be de-encrypted prior to following the transportable tablespaces migration and then encrypted in the target database after migration.

Preparing the Source-system

1. Apply Prerequisite Patches

For this example an Oracle E-Business Suite release 12.1 Rapid Install with the Vision database was done. That installs a file-system based database with Oracle RDBMS Enterprise Edition 11.1.0.7. Prior to the export process the source system database was upgraded to RDBMS 11.2 and the steps in “*Interoperability Notes Oracle E-Business Suite Release 12 with Oracle Database 11g Release 2 (11.2.0)*”, ID [1058763.1](#) were followed for the database patch requirements.

a. 11.2 Source Database (conditional)

If your source database is 11.2 then MOS ID [1058763.1](#) should be consulted for any updates and it will also be used to set up the target system 11.2 database ORACLE_HOME.

b. 11.1.0.7 Source Database (conditional)

If your source database is 11.1.0.7 then you must first install patchset update (PSU) 11.1.0.7.6 and then apply patch 8353098. The import step will fail without this patch.

2. Obtain and apply the Applications consolidated export/import utility patch

Apply the export/import patch, patch [13377437](#), to the administration server node. Note that this patch is currently 'Controlled' and requires customers to file a Service Request (SR) with Oracle Support to request an appropriate password to download. While working with Oracle Support to obtain the password, you should also send an email to apps_relgrp_us@oracle.com with the SR number as a notification so that Development can monitor the SR.

3. Apply Object Grants Performance patch

Download and apply database generic patch 10195109 to the source system. This patch supersedes patch 10185319 and fixes an issue with the performance of importing grant objects. It is recommended on the source system so that grants are ordered by user on the export which will improve the import time. Patch 10185319 is also recommended on the target system as well and will make a substantial improvement to the import time. See MOS ID [1267951.1](#), “DataPump Import (impdp) is Very Slow at Object Grants.”

4. Convert Non-SYSTEM Dictionary Managed Tablespaces to Locally Managed

1. Verify if any dictionary managed non-system tablespaces exist

```
SELECT tablespace_name
FROM dba_tablespaces
where extent_management='DICTIONARY' and TABLESPACE_NAME <> 'SYSTEM';
```

2. If any rows are returned then convert those Tablespaces to locally managed Tablespaces. See MOS [735978.1](#) *How to change a dictionary managed tablespace to a locally managed tablespace?*.

5. Synchronize Oracle Text Indexes

Oracle Text is implemented with domain indexes. At the moment that a row is inserted/updated/deleted the row will be put in table `ctxsys.dr$pending` in order to be indexed. If these rows are not synchronized prior to export then they will substantially slow down the export during the “Processing object type `TRANSPORTABLE_EXPORT/DOMAIN_INDEX/INDEX`” step. These steps are taken from MOS [1302676.1](#).

1. Check if there are indexes in need of synchronization:

```
set pagesize 9999;
select pnd_index_owner,pnd_index_name,count(*)
from ctxsys.ctx_pending
group by pnd_index_owner,pnd_index_name
order by pnd_index_owner,pnd_index_name;
```

If Step 1 shows no rows you can continue with the TTS export

2. Check which indexes need to be synchronized before TTS export:

```
set pagesize 9999 header off flush off echo off
spool /tmp/ctx_indexes_to_synchronize.sql
select distinct
'exec ctx_ddl.sync_index(''||pnd_index_owner||'.||pnd_index_name||'');'
from ctxsys.ctx_pending;
spool off
```

3. Synchronize the indexes as shown at output of Step 2.

```
@/tmp/ctx_indexes_to_synchronize.sql
```

4. Repeat step 1 to confirm that all indexes have been synchronized:

```
=> if TRUE continue
=> if FALSE check why synchronize failed
```

6. Create a working directory

For this example a working directory named `/ebs/expimp` was created on network file system (NFS) storage. This working directory will contain all generated files and scripts required for the complete process. To use NFS you must ensure the correct mount options are used. For this example all 3 systems:

- a. the source database
- b. the target database
- c. the primary applications node

were set up with an NFS mount point as follows:

As root execute:

```
mkdir /ebs
chown oracle:dba /ebs
mount dscbbg03:/export2/ /ebs -o rsize=32768,wsiz=32768,hard,actimeo=0,nolock
```

As oracle user (the software owner):

```
mkdir /ebs/expimp
```

This mount point will be used on all three systems: the source database, the primary applications node, and the target database node. For details of the system environment the appendix at [Environment Details](#). For further details on NFS mounting Oracle files see MOS ID [359515.1](#).

An alternative to using NFS storage is to use the Oracle Database File System (DBFS) on the target Oracle Exadata Database Machine to hold the Oracle Data Pump exported dump files. This DBFS file system can also be mounted remotely. To set up and configure DBFS, see [Oracle Database SecureFiles and Large Objects Developer's Guide 11g Release 2 \(11.2\)](#) and MOS ID [1054431.1](#) for setting up DBFS on Oracle Exadata Database Machine.

7. Copy the export/import Patch Files to the Working Directory

Copy the files from the applications administration node directory to the working directory. If the working directory was not NFS mounted to the source and target systems and is a local directory then make sure to copy the files to both the source and target.

```
$ cp $AU_TOP/patch/115/sql/* /ebs/expimp
```

8. Create transport_set_violations table

Use SQL*Plus to connect to the source database as sysdba and run the following command:

```
SQL> exec dbms_tts.transport_set_check(' [TABLESPACE] ');
```

You may use any existing tablespace name for [TABLESPACE]. It does not matter if errors appear. What is needed is to create the SYS.transport_set_violations table for subsequent use by the [auxttspre.sql](#) script to ensure all object references from the transportable set are contained in the transportable set.

```
SQL> exec dbms_tts.transport_set_check('APPS_TS_SUMMARY');
```

```
PL/SQL procedure successfully completed.
```

9. Run the auxttspre.sql script

In addition to generating the files below, `auxttspre.sql` also does a self-containment check, invoked by running the procedure `DBMS_TTS.TRANSPORT_SET_CHECK`. This validates that all object references from the transportable set are contained in the transportable set. The transportable set is determined in `auxttspre.sql` by using the `tbpsset` cursor. See the `auxttspre.sql` file for details.

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 auxxtspre.sql script.

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

The following files are generated by auxxtspre.sql:

File	Description
aucrdb.sql	Script that creates the target database instance with the appropriate tablespace and file structure
auxttsread.sql	Script that converts the transportable tablespace set in the source database to read only
auxttswrite.sql	Script that converts the transportable tablespace set in the source database back to read write.
audeftbsp.sql	Script that converts the users to their corresponding default tablespaces.
auexpxtts.dat	Export parameter file for the transportable tablespace set.
auexpglb.dat	Export parameter file for the global temporary tables
auexpsys.dat	Export parameter file for the SYSTEM tablespace schemas.
auexpmsc.dat	Export parameter file for the other objects.
aurman.dat	Recovery manager parameter file to convert the endian format of data files.
auimpxtts.dat	Import parameter file for the transportable tablespace set.
auimpmsc1.dat	Import parameter file for the other objects
auimpmsc2.dat	Import parameter file for the other objects
auimpglb.dat	Import parameter file for the global temporary tables.
auimpsys.dat	Import parameter file for the SYSTEM tablespace schemas.
auexpcall.dat	Converts database callouts into data
auexpcallclean.dat	Converts callout data back to metadata

Table 1 auxxtspre.sql Generated Files

Review the auxxtspre.log file, fix any transport set violations it lists out, and repeat the instructions in this step before proceeding with the rest of this document. A successful run should show the following message in the auxxtspre.log file

```
Transportable set check completed successfully.
```

10. Record Advanced Queue settings

Advanced Queue settings are not propagated in the target database during the export/import process. Therefore, you must record them beforehand and enable them in the target database instance afterwards. The export/import patch, [10396457](#), contains `auque1.sql`, which generates a script called `auque2.sql`. You will use the generated `auque2.sql` to enable the settings in the target database instance after the import is complete.

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`. On the source Database system:

```
$ sqlplus "/ as sysdba" @auque1
```

11. Identify the Source System Time Zone Version

```
SQL> select name,value$ from props$ where name='DST_PRIMARY_TT_VERSION';
```

```
NAME
```

```
-----
```

```
VALUE$
```

```
-----
```

```
DST_PRIMARY_TT_VERSION
```

```
10
```

Preparing the Target Database

1. Oracle Exadata Database Machine Best Practices and Health Check

- Review Oracle Exadata Database Machine best practices in MOS ID [1187674.1](#), “*Master Note for Oracle Database Machine and Exadata Storage Server.*”
- If Oracle Exadata Database Machine health has not been verified yet then follow MOS ID [1070954.1](#), “Database Machine Health Check”.
- Review the MAA paper, “[Oracle E-Business Suite on Exadata](#)”

2. Verify Oracle E-Business Suite Required Packages

For the latest list of required packages also see MOS ID [761566.1](#), “*Oracle Applications Installation and Upgrade Notes Release 12 (12.1.1) for Linux x86-64.*”

3. Prepare the target database Oracle 11g Oracle home

a. Install a target database Oracle 11g Oracle home (conditional)

Creating a new `ORACLE_HOME` is optional since you can use the existing `ORACLE_HOME` that is installed with Oracle Exadata Database Machine.

The E-Business Suite requires exclusive use of its database ORACLE_HOME -- it cannot be shared with other applications, or even with other installations of the E-Business Suite.

In past releases, Oracle Database patch sets consisted of a specific set of files that replaced files in an existing ORACLE_HOME. Beginning with Oracle Database 11g Release 2 (11.2.0.2), patch sets are full installations of the database software. See “Important Changes to Oracle Database Patch Sets Starting With 11.2.0.2” [MOS ID [1189783.1](#)] for more details on this.

The software can be downloaded from My Oracle support: patch [10098816](#), making sure to select the Linux x86-64 platform.

Install the new 11.2.0.2 ORACLE_HOME set of software on the target Oracle Exadata Database Machine and make sure to choose the following options:

- Install Software Only
- Oracle Real Application Clusters database installation and select all Oracle RAC nodes

b. Apply Prerequisite Patches

- i. Follow MOS ID [888828.1](#), “Database Machine and Exadata Storage Server 11g Release 2 (11.2) Supported Versions” to ensure the latest Oracle Database machine patches are applied.
- ii. In addition to installing a new 11.2.0.2 ORACLE_HOME, MOS ID [1392527.1](#) and [1058763.1](#) must be followed. Since the application tier patches were already applied previously when the source system database was upgraded to 11.2, that section of [1058763.1](#) is not necessary now. If you did not previously apply the application tier patches as instructed under “[Apply Prerequisite Patches](#)” then you should do that now. Be sure to consult MOS ID [1392527.1](#) and [1058763.1](#) for an up to date list of E-Business related 11.2 database patches.
- iii. Check and apply the latest 11.2 Patchset Updates and Exadata patches, see “Oracle Recommended Patches -- Oracle Database” [ID [756671.1](#)]. Patchset Updates also include the latest Critical Patch Update (CPU). Note that some of the patches listed in MOS ID [1058763.1](#) may already be part of the latest PSU or CPU. Consult MOS ID [1392527.1](#) for which 11.2 database patches are needed.

c. Install the Database Examples

The examples CD install is needed because it contains all the localization files needed for Oracle Text.

- i. Download the software from http://download.oracle.com/otn/linux/oracle11g/R2/linux.x64_11gR2_examples.zip
- ii. Follow the installation steps in the “[Database Examples Installation Guide](#)” in chapter 3 to install in the Oracle Database 11g Release 2 ORACLE_HOME on each target database node.

d. Create the nls/data/9idata directory

On each of the target system database nodes 11.2 ORACLE_HOME's execute the following:

```
perl $ORACLE_HOME/nls/data/old/cr9idata.pl
```

This will create the \$ORACLE_HOME/nls/data/9idata directory which will be set in the default database environment file,

```
$ORACLE_HOME/<${ORACLE_SID}_short-hostname>.env, in the ORA_NLS10 variable.
```

4. Apply Object Grants Performance patch

Download and apply database generic patch 10195109 to the target system. This patch supersedes patch 10185319 and fixes an issue with the performance of importing grant objects. It is also recommended on the source system so that grants are ordered by user on the export which will improve the import time. Patch 10185319 will make a substantial improvement to the import time. See MOS ID [1267951.1](#), “DataPump Import (impdp) is Very Slow at Object Grants.”

5. Create and prepare the Target init.ora File

Refer to MOS article, “Database Initialization Parameters for Oracle Applications Release 12” [ID [396009.1](#)] and update the init.ora file 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.

- a. Copy the source system (bur407-25) init.ora file from the \$ORACLE_HOME/dbs directory to the target system.

On the target system (sclczdb01)

```
cd $ORACLE_HOME/dbs
scp bur407-25:/ebs/VIS/db/tech_st/11.2.0/dbs/initVIS.ora initSOLXTTS1.ora
```

- b. Edit the file to change file system paths with the new database name of SOLXTTS, the appropriate Oracle ASM disk group and add Oracle RAC parameters. The resulting file is included in the appendix at [Target Database Starting Parameter File](#). Note that the CLUSTER_DATABASE parameter must be unset at database creation.
- c. Note that later Linux hugepages will be configured. The Automatic Memory Management feature, enabled by the MEMORY_TARGET / MEMORY_MAX_TARGET database parameters, is incompatible with the use of HugePages, do not set these parameters. The Automatic Shared Memory Management feature, enabled by setting SGA_TARGET, IS compatible with HugePages.

For further details on database parameter settings see MOS ID [396009.1](#), “Database Initialization Parameters for Oracle Applications Release 12”

6. Create a working directory (conditional)

If not using an NFS mounted file system or DBFS as discussed previously under the “[Preparing the Source System](#)” section under “[Create a working directory](#)” then create a new directory here.

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

7. Edit the target database instance creation script aucrdb.sql

Edit the aucrdb.sql script created by [auxttspre.sql](#) and change it as listed in the bullet items below. You can also consolidate tablespace datafiles into a single or minimal number of files if desired. For example, if a tablespace has three datafiles of size 5G due to the fact that a default E-Business Suite install does not enable AUTOEXTEND, then you can consolidate those into a single 15G datafile.

A sample of a modified aucrdb.sql script is in the appendix at [Modified aucrdb.sql Script](#). The changes to the original are:

- Added AUTOEXTEND. Note that you can also add the ‘MAXSIZE <n>’ clause to the ‘AUTOEXTEND ON’ clause if you want to limit the extension size.
- Datafile paths were ‘?/dbf/<datafile-name>’. The target database will be SOLXTTS and use the already created Oracle ASM disk groups of +DATA and +RECO.
- The LOGFILE clause was modified to multiplex redo logs in both disk groups, +DATA and +RECO.
- The uniform extent size of 128K was removed in favor of the default extent management method, AUTOALLOCATE .
- DDL to create the second UNDO tablespace was added. An UNDO tablespace must be added for each instance. In this example there are two database instances on the target system.
- For further background see “Using Oracle 11g Release 2 Real Application Clusters with Oracle E-Business Suite Release 12” [ID [823587.1](#)]

8. Set up the Database Listeners

See MOS ID [823587.1](#) under section “3.3 Listener Configuration in 11gR2” for details of setting up a listener. E-Business Suite does support use of the Oracle RAC 11g database release 2 Single Client Access Name (SCAN), however it cannot make complete use of the SCAN since the Oracle E-Business Suite clients are pre 11.2 (see <http://www.oracle.com/technetwork/database/clustering/overview/scan-129069.pdf> for a

detailed explanation of this). This example uses a new listener. For an example of using the SCAN listener see the “[Installing Oracle E-Business Suite Release 12 with Exadata Database Machine](#)” paper.

- a. Set your environment to the new target database ORACLE_HOME. From [Environment Details](#) that would be as follows:

```
ORACLE_HOME /u01/app/oracle/product/11.2.0.2/dbhome_2
ORACLE_SID=SOLXITTS1
PATH=$ORACLE_HOME/bin:$PATH
```

- b. Run netca and create the SOLXITTS listener selecting the “Cluster Configuration”. For this example the listener was named LISTENER_SOLXITTS. Use an unused port for the listener port as well. For this example port 1524 was used. This will create a listener on each Oracle RAC node with the same name and add it to the Oracle RAC Oracle Cluster Registry (OCR). Optionally, you can use srvctl as shown in MOS ID [823587.1](#).
- c. Edit \$ORACLE_HOME/network/admin/listener.ora on each database node and add a static SID entry. See the example in the appendix at [Initial listener.ora Files](#).
- d. Reload the listener on all nodes by executing the following on each database node

```
lsnrctl reload listener_solxtts
```

- e. Status the listener to make sure it is running

```
lsnrctl status LISTENER_SOLXITTS
```

```
LSNRCTL for Linux: Version 11.2.0.2.0 - Production on 10-JUN-2010 09:52:04
```

```
Copyright (c) 1991, 2009, Oracle. All rights reserved.
```

```
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC) (KEY=LISTENER_SOLXITTS)))
STATUS of the LISTENER
```

```
-----
```

```
Alias                LISTENER_SOLXITTS
Version              TNSLSNR for Linux: Version 11.2.0.2.0 -
Production
Start Date           10-JUN-2010 09:51:30
Uptime               0 days 0 hr. 0 min. 34 sec
Trace Level          off
Security             ON: Local OS Authentication
SNMP                 OFF
Listener Parameter File
/u01/app/oracle/product/11.2.0.2/dbhome_2/network/admin/li
stener.ora
Listener Log File
/u01/app/oracle/diag/tnslsnr/sclczdb01/listener_vismi
g/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=LISTENER_SOLXITTS)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=10.204.74.168) (PORT=1524)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=10.204.77.181) (PORT=1524)))
The listener supports no services
The command completed successfully
```

9. Create the Target Database

- a. Ensure that the environment of your session on the target database system is set to the new ORACLE_SID of SOLXITTS1 and the ORACLE_HOME and ORA_NLS10 are also set to the target 11.2 location.

```
export ORACLE_HOME=/u01/app/oracle/product/11.2.0.2/dbhome_2
export ORACLE_SID=SOLXTTS1
export ORA_NLS10=$ORACLE_HOME/nls/data/9idata
export PATH=$ORACLE_HOME/bin:$PATH
```

- b. Ensure that the time zone to be used in this target database matches the [source database time zone](#). The source and target database time zone settings must match or the import will fail with:

```
ORA-39002: invalid operation
ORA-39322: Cannot use transportable tablespace with timestamp with
timezone columns and different timezone version.
```

You cannot downgrade the time zone version, only upgrade. To control what time zone version this target database will use set the `ORA_TZFILE` environment variable to the [version of the source database](#) prior to running the create database in [step c](#) below.

```
export ORA_TZFILE="$ORACLE_HOME/oracore/zoneinfo/timezlrq_10.dat"
```

- c. Using the edited script from [Modified aucrdb.sql Script](#) and the parameter file from [Target Database Starting Parameter File](#), create the database.

```
cd /ebs/expimp
sqlplus "/" as sysdba"
SQL> spool aucrdb.log;
SQL> startup nomount;
SQL> @aucrdb.sql
SQL> exit;
```

10. Complete the Target Database Preparation

The export/import patch, [10396457](#), that was applied to the source administration server node in “[Apply the Applications consolidated export/import utility patch](#)” contains four scripts that are needed on the target database server node: `audb1120.sql`, `ausy1120.sql`, `aujv1120.sql`, and `aumsc1120.sql`. As you run each of the next four steps, note the following:

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

a. Set up the SYS schema

```
sqlplus "/" as sysdba" @audb1120
```

b. Set up the SYSTEM schema

```
sqlplus system/welcome1 @ausy1120
```

c. Install Java Virtual Machine

```
sqlplus system/welcome1 @aujv1120
```

d. Install other required components

```
sqlplus system/welcome1 @aumsc1120.sql FALSE SYSAUX TEMP
```

11. Set CTXSYS parameter

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

12. Disable automatic gathering of statistics

Use `/ebs/expimp/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 put the database in restricted mode and run `adstats.sql`:

```
$ sqlplus "/ as sysdba"
SQL> alter system enable restricted session;
SQL> @adstats.sql
$ sqlplus "/ as sysdba"
SQL> alter system disable restricted session;
SQL> exit;
```

13. Set up the SPFILE on the Target Database Servers

These commands are done on the first database server node, `sclczdb01`.

1. Set `CLUSTER_DATABASE=TRUE` in the database parameter file, `$ORACLE_HOME/dbs/initSOLXTTS1.ora`

```
*.cluster_database=true
```
2. Create a backup copy of the current database parameter file to a `.pfile` file

```
cp $ORACLE_HOME/dbs/initSOLXTTS1.ora
   $ORACLE_HOME/dbs/initSOLXTTS1.pfile2
```
3. Create the SPFILE

```
SQL> create spfile='+DATA/solxtts/spfilesolxtts.ora' from
pfile='$ORACLE_HOME/dbs/initSOLXTTS1.pfile2';

File created.
```
4. Create database parameter file, `$ORACLE_HOME/dbs/initSOLXTTS1.spfile`, that points to the new SPFILE

```
spfile='+DATA/solxtts/spfilesolxtts.ora'
```
5. Copy the new `.spfile` to the database parameter file

```
cp $ORACLE_HOME/dbs/initSOLXTTS1.spfile
   $ORACLE_HOME/dbs/initSOLXTTS1.ora
```
6. Restart the database

```
SQL> shutdown immediate
SQL> startup
```
7. Copy the database parameter file to the other node(s) with the appropriate name. In this case study, `sclczdb02` is the second Oracle RAC node (see Environment Details for details).

```
scp initSOLXTTS1.ora sclczdb02:`pwd`/initSOLXTTS2.ora

initSOLXTTS1.ora          100%  39      0.0KB/s
00:00
```

14. Create Password File on Each node

```
cd $ORACLE_HOME/dbs
orapwd file=orapw${ORACLE_SID} password=welcome1
```

15. Set up Oracle RAC

- a. To support use of the database configuration assistant (dbca), Oracle Enterprise Manager discovery and monitoring, add the database and ORACLE_SID to /etc/oratab on each node

```
SOLXTTS:/u01/app/oracle/product/11.2.0.2/dbhome_2:N
On sclczdb01: SOLXTTS1:/u01/app/oracle/product/11.2.0.2/dbhome_2:N
On sclczdb02: SOLXTTS2:/u01/app/oracle/product/11.2.0.2/dbhome_2:N
```

- b. Ensure an undo tablespace exists for each database instance. There were created as part of executing [aucrdb.sql](#) above in the “[Create the Target Database](#)” step..

```
select tablespace_name from dba_tablespaces where contents='UNDO';
TABLESPACE_NAME
-----
UNDO_TBS1
UNDO_TBS2
```

If you need to create an undo tablespace then an example statement is:

```
CREATE UNDO TABLESPACE UNDO_TBS2
DATAFILE '+DATA' SIZE 26508M AUTOEXTEND ON
EXTENT MANAGEMENT LOCAL;
```

- c. Create an additional redo thread and redo logs for each database instance

- i. Get the redo log size and maximum group number


```
SQL> select distinct bytes/1048576 from v$log;
```

```
BYTES/1048576
-----
1024
SQL> select max(group#) from v$log;
```

```
MAX(GROUP#)
-----
3
```

- ii. Add redo logs for other thread or threads:

```
alter database
add logfile instance `SOLXTTS2'
group 4 ('+DATA', '+RECO') size 1024M,
group 5 ('+DATA', '+RECO') size 1024M,
group 6 ('+DATA', '+RECO') size 1024M;
```

- iii. Enable the new thread:


```
alter database enable thread 2;
```

- d. Add the database and instances to the Oracle Cluster Registry (OCR):

```
srvctl add database -d SOLXTTS -o /u01/app/oracle/product/11.2.0/dbhome_2
-a "DATA,RECO"
srvctl add instance -d SOLXTTS -i SOLXTTS1 -n sclczdb01
srvctl add instance -d SOLXTTS -i SOLXTTS2 -n sclczdb02
```

- e. Verify srvctl is working for the database:

```

srvctl stop database -d SOLXTTS
srvctl start database -d SOLXTTS
srvctl status database -d SOLXTTS
Instance SOLXTTS1 is running on node sclczdb01
Instance SOLXTTS2 is running on node sclczdb02

```

16. Configure HugePages on each database node:

HugePages are not configured by default on Oracle Exadata Database Machine.

Follow MOS ID [361468.1](#), “HugePages on Oracle Linux 64-bit”. Ensure that when running the recommended script provided with [Note 401749.1](#) that the database is started.

For RDBMS 11.2.0 there is an additional workaround required that is described in MOS ID [983715.1](#), “11gR2 Grid Infrastructure Does not Use ULIMIT Setting Appropriately”.

17. Back up the target database:

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

Exporting the Source Database

1. Modify the export parameter files

Use a text editor to modify the [auexpxtts.dat](#), [auexpmsc.dat](#), and [auexpsys.dat](#) Data Pump parameter files created in the “[Run the auxxtspre.sql script](#)” step in the “[Preparing the Source-system](#)” section 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)	xttsmeta%U.dmp, xttsmsc%U.dmp, or xttssys%U.dmp
filesize	export dump file size	1GB
logfile	log file name	expmeta.log, expmsc.log, or expsys.log

Table 2 Export Customizable Parameters

2. Create a directory in the system schema

Create a directory 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 '/ebs/expimp';

```

3. Modify the auexpglb.dat export parameter file

Use a text editor to modify the auexpglb.dat export parameter file created in Section 1 to reflect the source environment and other customized parameters.

The customizable parameters are:

Parameter	Description	Template Value
file	export dump file name(s)	gtab.dmp
filesize	export dump file size	1GB
log	log file name	expgtab.log

Table 3 auexpblg Export Parameter File Parameters

4. Shut down Applications server processes

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

5. Drop Large XLA Packages (recommended for an Oracle E-Business Suite release 12 source system)

There are a number of large packages (greater than 100,000 lines) associated with the Oracle E-Business Suite Sub-ledger Accounting (XLA) feature. Data Pump does not handle these performantly and these packages can add a substantial amount of time (15.5 hours in testing) to the XTTTS “[Import miscellaneous data](#)” step. Thus, it is recommended to drop these packages prior to the export and re-create them after the import. This saved 15.5 hours during the import because the drop and re-creation only take about 40 minutes total.

Use SQL*Plus to connect to the source database as sysdba and run the following command to execute the [genDropXLA.sql](#) script to drop the large XLA packages:

```
$ sqlplus "/ as sysdba" @genDropXLA.sql
```

6. Purge unused columns

Use SQL*Plus to connect to the source database as sysdba and run the following commands to find and purge any unused columns:

```
SQL> select * from sys.dba_unused_col_tabs;
SQL> alter table [table name] drop unused columns;
```

Example to generate the necessary SQL:

```
set head off pages 0 feedback off verify off echo off
spool dropUnusedCols.sql
select 'alter table ` || owner || `.' || table_name || ` drop unused
columns;`
from sys.dba_unused_col_tabs;
spool off
```

7. Purge recycle bins

Run the following commands to find and purge the recycle bin:

```
SQL> select owner, count(*) from dba_recyclebin group by owner;
SQL> connect [user]/[password];
SQL> purge recyclebin;
```

Example to generate the necessary SQL:

```
REM chr(13)=CR (carriage return), chr(10)=LF (linefeed)
set head off pages 0 feedback off verify off echo off
spool purgeRecyclebin.sql
select distinct('connect ' || owner || '/' || owner || chr(13) ||
chr(10) || 'purge recyclebin;')
from dba_recyclebin;
spool off
```

8. Back up AZ table data (conditional)

If you are using AZ.H, upgrade to AZ.H.DELTA.1. See [document 403092.1](#) on My Oracle Support for instructions.

9. Grant privilege to source system schema

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

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

10. Remove the PAY_ALL_PAYROLLS_F Public Synonym

To avoid the following import errors related to the PAY_ALL_PAYROLLS_F public synonym, drop the public synonym

```
ORA-39083: Object type VIEW failed to create with error:
ORA-04020: deadlock detected while trying to lock object APPS.PAY_PAYROLLS_F
ORA-01731: circular view definition encountered
```

```
SQL> drop public synonym PAY_ALL_PAYROLLS_F;
```

11. Execute the callout setup that creates the callout tables

This step does not actually export anything but sets up the Data Pump callout tables used during export.

```
expdp "'/ as sysdba'" parfile=auexpcall.dat
```

Note there is an expected set of errors for this step that can be ignored

```
ORA-39168: Object path PRE_SYSTEM/SYSTEM_CALLOUT was not found.
```

```
ORA-39168: Object path SCHEMA/SCHEMA_CALLOUT was not found.  
ORA-39168: Object path SCHEMA/TABLE/INSTANCE_CALLOUT was not found.  
ORA-31655: no data or metadata objects selected for job
```

12. Mark the tablespaces within the transportable set read only

Use SQL*Plus to connect to the database as SYSDBA and run `auxttsread.sql` to mark the tablespaces within the transportable set as read only:

```
$ sqlplus "/" as sysdba" @auxttsread.sql
```

13. Copy the read only data files to the target

Once the source system non-system tablespaces are made read-only you can copy them all to the target database server node.

This can be done in parallel to the succeeding export steps and must be completed while the datafiles are read-only.

In contrast to MOS ID [454574.1](#) instructions, we will utilize Oracle Exadata Database Machine to convert the datafiles, i.e. do the conversion on the target system rather than on the source system. There are multiple ways to copy the datafiles to the target system.

File system datafiles:

- Use FTP or SCP to move the datafiles directly from the source system to the target system. See [genSCPsh.sql](#) for an example scp generation script.
- NFS mount the file system containing the datafiles to the target system and copy the files to the target system.
- Reconfigure the SAN so that the storage devices can be mounted directly on the target system. Refer to your storage vendor for operating system specific details.

ASM datafiles:

- Reconfigure the SAN so that the storage devices can be mounted directly on the target system. Refer to your storage vendor for operating system specific details.
- Use the PL/SQL `DBMS_FILE_TRANSFER` package to transfer datafiles from the source instance to target instance. See the [Oracle Database Administrator's Guide](#) for details. For an example see the `DBMS_FILE_TRANSFER` section of MOS ID [371556.1](#), "How to Move Tablespaces Across Platforms Using Transportable Tablespaces With RMAN."
- Use XML DB FTP capability to ftp datafiles from the source instance to the target instance. See the Oracle [XML DB Developer's Guide](#) for details.
- Use RMAN to move datafiles to a staging area on the source system, use standard operating system tools to transfer datafiles to the target system, and then use RMAN to

move the files into Oracle ASM on the target system. See the Oracle Database [Backup and Recovery User's Guide](#) for details.

- The Oracle ASM 'asmcmd cp' command should also be an option, but at the time of this writing there was a known issue blocking use of this. Contact Oracle Support for updates on patch 9078442, "ORA-19762 from asmcmd cp copying file with different byte order from file system."

For this example, since our source system to target system connection was over a wide area network, we used scp to copy the datafiles to the target system. The scp commands were generated with the sample [genSCPsh.sql](#) script. Execute the generated script to copy the source system datafiles to the target system staging area.

14. Convert the endian format of the datafiles on the target system

This can be done in parallel to the succeeding export steps and must be completed while the datafiles are still read-only on the target.

If you are using this process then it is assumed that the source and target platforms have different endian formats. Thus, a conversion is necessary. Since we are doing the conversion on the target system the `aurman.dat` script from the [auxttspre.sql step](#) will *not* be used as it is for using when the conversion is done on the source system.

- Execute the SQL script, [genRMANconvert.sql](#), on the source system to generate a new [aurman.dat](#) file.
- Copy the new [aurman.dat](#) file to the target system working directory.
- Modify the generated [aurman.dat](#) script as needed. You should review the following for possible modification:
 - Datafile path
 - FROM PLATFORM setting
 - PARALLEISM setting
 - Paths in the DB_FILE_NAME_CONVERT line
 - While most systems are using automatic undo, make sure to remove any datafile lines that are associated with tablespaces that hold conventional rollback segments.
- Use the [RMAN Convert Script](#) shell script on the target system to execute the conversion on the target system.

The following steps can be run in parallel, but only after the tablespaces are put in read-only mode.

15. Monitoring Job Status

For details on monitoring Data Pump jobs see the "Oracle Database Utilities 11g Release 2 (11.2)" guide under Chapter 1, [Monitoring Job Status](#). To summarize, Data Pump jobs can be monitored through the following:

- The Data Pump log file
LOGFILE setting in the parameter file
- Attaching to the Data Pump job
See the Data Pump ATTACH option

- The `V$SESSION_LONGOPS` view
- the `DBA_DATAPUMP_JOBS`, `USER_DATAPUMP_JOBS`, or `DBA_DATAPUMP_SESSIONS` view

See the [Monitoring Job Status](#) documentation for details.

16. Capture Timings (conditional)

If you want to capture timestamp information for each output line in the export logs use the [Timestamp Script](#) in the appendix to prefix the export log output with a timestamp. An example of that would be:

```
tailTime expxtts.log > expxtts_time.log
```

The output in `expxtts_time.log` will then have the form:

```
YYMMDD hh:mi:ss <export log message>
```

e.g.

```
100602 11:42:07 Processing object type TRANSPORTABLE_EXPORT/TRIGGER
```

17. Export the transportable tablespace set

Start an export session on the source database server node using the [auexpxtts.dat](#) parameter file. Use the following command:

```
$ expdp "'/ as sysdba'" parfile=auexpxtts.dat
```

18. Export SYSTEM tablespace schemas

Export the SYSTEM tablespace schemas by starting an export session on the source database server node using the `auexpsys.dat` parameter file. Use the following command:

```
$ expdp "'/ as sysdba'" parfile=auexpsys.dat
```

19. Export global temporary tables

Global temporary tables are not exported as they do not reside in any tablespace. Start an export session on the source database server node using the `auexpglb.dat` parameter file to export the global temporary tables. Use the following command:

```
$ exp "'/ as sysdba'" parfile=auexpglb.dat
```

20. Export miscellaneous data

Start an export session on the source database server node using the `auexpmsc.dat` parameter file to export other objects. Use the following command:

```
$ expdp "'/ as sysdba'" parfile=auexpmsc.dat
```

21. Revoke privilege from source system schema

After all prior steps are complete, then 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;
```

22. Modify source tablespaces back to read/write mode (optional)

If the source database is still to be used, 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 auxttswrite.sql script.

```
$ sqlplus "/ as sysdba" @auxttswrite.sql
```

23. Convert callout data back to metadata (optional)

If the source database is still to be used, start an export session on the source database server node using the [auexpallclean.dat](#) parameter file. This converts callout data back into metadata. Use the following command:

```
$ expdp "'/ as sysdba'" parfile=auexpallclean.dat
```

24. Shut down the database

Use SQL*Plus to connect to the database as SYSDBA and use the following command to shut down the database:

```
$ sqlplus "/ as sysdba"
SQL> shutdown normal;
```

Importing

1. Copy the export dump files (conditional)

Copy the export dump files generated in the “[Exporting the Source Database](#)” section from the source database server node to the target database server node. In this example the dump files are in the NFS mounted working directory, /ebs/expimp, so this was not necessary.

2. Modify the import parameter files

Copy all the import parameter files generated in Section 1 from the source database server node to the target database server node. Use a text editor to modify the [auimpxtts.dat](#), [auimpmsc1.dat](#), and [auimpsys.dat](#) data pump parameter files to reflect the target environment and other customized parameters.

The customizable parameters are:

Parameter	Description	Template Value
directory	directory where the export dump files are located	dmpdir
dumpfile	export dump file name(s)	xttsmeta%U.dmp,

logfile	log file name	xttmsc%U.dmp, xttssys%U.dmp impmeta.log, impmsc.log, or impsys.log
transport_datafiles	location of every data file in the transportable tablespace set	?/dbf/[file name]

Table 4 Import Parameter Files Parameters

Additionally modifications are:

- In the generated [auimpmsc1.dat](#) add TABLESPACE to the EXCLUDE line. See [auimpmsc1.dat](#).
- In [auimpxtts.dat](#) modify the directory structures for transport_datafiles to reflect the location of the data files on the target database server node.
- Use a text editor to modify the [auimpglb.dat](#) export parameter file created in Section 1 to reflect the target environment and other customized parameters.

The customizable parameters are:

Parameter	Description	Template Value
file	import dump file name(s)	gtab.dmp
log	import log file	impgtab.log

Table 5 Global Temporary Table Import Parameters

3. Create a working directory

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 '/ebs/expimp';
```

4. Monitoring Job Status

For details on monitoring Data Pump jobs see the “Oracle Database Utilities 11g Release 2 (11.2)” guide under Chapter 1, [Monitoring Job Status](#). To summarize, Data Pump jobs can be monitored through the following:

- The Data Pump log file
LOGFILE setting in the parameter file
- Attaching to the Data Pump job
See the Data Pump ATTACH option
- The V\$SESSION_LONGOPS view
- the DBA_DATAPUMP_JOBS, USER_DATAPUMP_JOBS, or DBA_DATAPUMP_SESSIONS view

See the [Monitoring Job Status](#) documentation for details.

5. Capture Timings (conditional)

If you want to capture timestamp information for each output line in the import logs use the [Timestamp Script](#) in the appendix to prefix the export log output with a timestamp. An example of that would be:

```
tailTime impxtts.log > impxtts_time.log
```

The output in `impxtts_time.log` will then have the form:

```
YYMMDD hh:mi:ss <export log message>
```

e.g.

```
101214 18:51:53 Processing object type DATABASE_EXPORT/SCHEMA/EVENT/TRIGGER
```

6. Import miscellaneous data

The import steps must be done serially, i.e. not in parallel. Start the import session on the target database server node using the [auimpmsc1.dat](#) parameter file. Use the following command:

```
$ impdp ''' as sysdba''' parfile=auimpmsc1.dat
```

7. Take a database backup

The next step to import the transportable tablespace set is not reentrant/restartable and thus it's imperative to take a backup at this stage in case any issues are encountered in the “Import the transportable tablespace set” step.

8. Import the transportable tablespace set

Start the import session on the target database server node using the `auimpxtts.dat` parameter file. Use the following command:

```
$ impdp ''' as sysdba''' parfile=auimpxtts.dat
```

9. Verify all non-system tablespaces and datafiles are imported

a. Run the following query on the source database:

```
set head off pages 0 feedback off verify off echo off
spool source_tbs
select a.tablespace_name
from dba_tablespaces a, dba_data_files b
where a.tablespace_name = b.tablespace_name
and a.tablespace_name not in
(select unique (default_tablespace)
from dba_users where username in
('ORDSYS', 'MDSYS', 'CTXSYS', 'ORDPLUGINS', 'LBACSYS',
'XDB', 'SI_INFORMTN_SCHEMA', 'DIP', 'DMSYS',
'DBSNMP', 'EXFSYS', 'WMSYS','ORACLE_OCM'))
and a.tablespace_name not in ('SYSTEM','SYSAUX')
and a.contents <> 'UNDO'
order by 1;
spool off
```

b. Run the same query with a different output file name on the target database:

```
set head off pages 0 feedback off verify off echo off
spool taregt_tbs
```

```

select a.tablespace_name
from dba_tablespaces a, dba_data_files b
where a.tablespace_name = b.tablespace_name
and a.tablespace_name not in
  (select unique (default_tablespace)
   from dba_users where username in
    ('ORDSYS', 'MDSYS', 'CTXSYS', 'ORDPLUGINS', 'LBACSYS',
     'XDB', 'SI_INFORMTN_SCHEMA', 'DIP', 'DMSYS',
     'DBSNMP', 'EXFSYS', 'WMSYS', 'ORACLE_OCM'))
and a.tablespace_name not in ('SYSTEM', 'SYSAUX')
and a.contents <> 'UNDO'
order by 1;
spool off

```

- c. Ensure that the tablespace output files match

```
$ diff source_tbs.lst target_tbs.lst
```

10. Modify tablespaces back to read/write mode

Once you have verified that 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 auxttwrite.sql script. Then repeat this on the target database.

```
$ sqlplus "/ as sysdba" @auxttwrite.sql
```

11. Modify default tablespaces of users

As the owner of the target database server file system and database instance, use SQL*Plus to connect to the source database as sysdba and run the [audeftbsp.sql](#) script.

```
$ sqlplus "/ as sysdba" @audeftbsp.sql
```

Check quota on the target system by issuing the sqlplus command:

```
SQL> select username, count(*) from dba_ts_quotas group by
username order by username;
```

Manually compare this result with that on the source machine. If there are any discrepancies, issue the sqlplus command:

```
SQL> alter user [user] unlimited quota on [tablespace];
```

12. Import global temporary tables

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

```
$ imp "/ as sysdba" parfile=auimpglb.dat
```

13. Import procedural objects

Start the import session on the target database server node using the [auimpmsc2.dat](#) parameter file. Use the following command:

```
$ impdp "/ as sysdba" parfile=auimpmsc2.dat
```

14. Import SYSTEM tablespace schemas

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

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

15. Revoke privilege from target 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;
```

Updating the Imported Database

This section outlines the post-import steps required to restore the target database to a fully functional state.

1. Set the Database Environment

Example

1. on sclczdb01: source [SOLXTTS.env](#)
2. on sclczdb01: source [SOLXTTS.env](#)

2. Reset Advanced Queues

Copy the auque2.sql script that was generated in the “[Preparing the Source-system](#)” section to the target database server node. Then, on the target database server node, as the owner of the Oracle software 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 process.

The script creates a log file in the current directory.

```
$ sqlplus "/ as sysdba" @auque2.sql
```

3. Start the new database listeners

The listeners should already be running but if they are not then start them now:

```
srvctl start listener -l listener_solxtts
```

4. Run adgrants.sql

1. Copy \$APPL_TOP/admin/adgrants.sql from the administration server node to the working directory, /ebs/expimp, in the database server node.

```
$ cp $APPL_TOP/admin/adgrants.sql /ebs/expimp
```

2. Use SQL*Plus to connect to the database as SYSDBA and run the script using the following command:

```
sqlplus "/ as sysdba" @adgrants.sql APPS
```

5. Grant create procedure privilege on CTXSYS

1. Copy `$AD_TOP/patch/115/sql/adctxprv.sql` from the administration server node to the database server node.

```
$ cp $AD_TOP/patch/115/sql/adctxprv.sql /ebs/expimp
```

2. Use SQL*Plus to connect to the database as APPS and run the script using the following command:

```
sqlplus apps/apps @adctxprv.sql welcome1 CTXSYS
```

6. Implement and run AutoConfig on the new Database Nodes

On application administration node:

1. Generate the `appsutil.zip` for the database tier, run the following command:

```
perl $AD_TOP/bin/admkappsutil.pl
```

2. Copy the `appsutil.zip` file to each DB server node `ORACLE_HOME`

```
scp /ebs/VIS/inst/apps/VIS_haovm021/admin/out/appsutil.zip
sclczdb01:/u01/app/oracle/product/11.2.0.2/dbhome_2
scp /ebs/VIS/inst/apps/VIS_haovm021/admin/out/appsutil.zip
sclczdb02:/u01/app/oracle/product/11.2.0.2/dbhome_2
```

On the first target database node:

3. As the APPS user, run the following command on the primary database node to de-register the current configuration:

```
sqlplus apps/apps
SQL>exec fnd_conc_clone.setup_clean;
```

Install `appsutil.zip` on each target database node:

4. `cd $ORACLE_HOME`
5. `unzip -o appsutil.zip`
6. Install the `$ORACLE_HOME/appsutil/jre` directory

```
$ cd $ORACLE_HOME/appsutil
$ cp -r $ORACLE_HOME/jdk/jre jre
$ mkdir clone
$ cp -r $ORACLE_HOME/jdk/jre clone/jre
```

7. Generate a new context file

```
$ perl $ORACLE_HOME/appsutil/bin/adbldxml.pl
The context file has been created at:
/u01/app/oracle/product/11.2.0.2/dbhome_2/appsutil/SOLXTTS1_sclczdb01.xml
```

8. Set the virtual hostname in each node's context file

Set the value of `s_virtual_host_name` to point to the virtual hostname for the database host, by editing the database context file:

```
On sclczdb01: $ORACLE_HOME/appsutil/SOLXTTS1_sclczdb01.xml
On sclczdb01: $ORACLE_HOME/appsutil/SOLXTTS2_sclczdb02.xml
```

```
On sclczdb01: <host oa_var="s_virtual_hostname">sclcz01-vip</host>
On sclczdb02: <host oa_var="s_virtual_hostname">sclcz02-vip</host>
```

9. Set the named listener name in each node's context file

```
<DB_LISTENER oa_var="s_db_listener">LISTENER_SOLXTTS</DB_LISTENER>
```

10. Run autoconfig on each database node

```
• $ORACLE_HOME/appsutil/bin/adconfig.sh
  contextfile=/u01/app/oracle/product/11.2.0.2/dbhome_2/appsutil/SOLXTTS1
  _sclczdb01.xml
Or more generically if the $ORACLE_HOME/<CONTEXT_NAME>.env file has been
sourced
• $ORACLE_HOME/appsutil/bin/adconfig.sh contextfile=$CONTEXT_FILE
  ...
    Using Context file          :
  /u01/app/oracle/product/11.2.0.2/dbhome_2/appsutil/SOLXTTS1_sclczdb01.x
  ml

Context Value Management will now update the Context file

    Updating Context file...COMPLETED

    Attempting upload of Context file and templates to
  database...COMPLETED

Updating rdbms version in Context file to db112
Updating rdbms type in Context file to 64 bits
Configuring templates from ORACLE_HOME ...

AutoConfig completed successfully.
```

11. Revise the listener on each database node

The reason the listener requires revision is:

- In 11.2, listeners are configured at the cluster level, and all nodes inherit the port and environment settings. This means that the TNS_ADMIN directory path must be the same on all nodes.
- The current version of AutoConfig creates a listener.ora file under the \$ORACLE_HOME/network/admin/<context-name> directory.
- AutoConfig does support SCAN listeners as of Oracle E-Business Suite 12.1.3. The SCAN was not used in this example. See MOS ID [823587.1](#), “Using Oracle 11g Release 2 Real Application Clusters with Oracle E-Business Suite Release 12,” for further details.

a. On each database node add an ifile entry to the Grid Infrastructure listener.ora and tnsnames.ora network files:

```
/u01/app/11.2.0/grid/network/admin/listener.ora
/u01/app/11.2.0/grid/network/admin/tnsnames.ora
```

```
to point to the Autoconfig generated files under
$ORACLE_HOME/network/admin/<context-name>/listener.ora
```

- i. Update the Grid Infrastructure listener.ora on each database node to point to the Autoconfig generated files under
`$ORACLE_HOME/network/admin/<context-name>/listener.ora`

For example, on database node 1, sclczdb01, add the following line to the end of `/u01/app/11.2.0/grid/network/admin/listener.ora`

```
IFILE=/u01/app/oracle/product/11.2.0.2/dbhome_2/network/admin/SOLXT
TS1_sclczdb01/listener.ora
```

- ii. Update the Grid Infrastructure tnsnames.ora on each database node to point to the Autoconfig generated files under
`$ORACLE_HOME/network/admin/<context-name>/tnsnames.ora`

For example, on database node 1, sclczdb01, add the following line to the end of `/u01/app/11.2.0/grid/network/admin/tnsnames.ora`

```
IFILE=/u01/app/oracle/product/11.2.0.2/dbhome_2/network/admin/SOLXT
TS1_sclczdb01/tnsnames.ora
```

- iii. Set the SDU in the sqlnet.ora ifile in the
`$ORACLE_HOME/network/admin/<CONTEXT_NAME>` directory on each DB node:

```
:::::::::::
sqlnet_ifile.ora
:::::::::::
DEFAULT_SDU_SIZE=32767
```

- b. Set the listener and database environment in OCR and verify it works with `srvctl`

```
srvctl setenv listener -l listener_solxtts -T
TNS_ADMIN=/u01/app/11.2.0/grid/network/admin
```

```
srvctl setenv database -d SOLXTTS -T
TNS_ADMIN=/u01/app/11.2.0/grid/network/admin
```

```
srvctl stop listener -l listener_solxtts
srvctl start listener -l listener_solxtts
```

12. Run autoconfig on database nodes once again

This command needs to be run once on each database node to register the node, then after all nodes are registered it must be run once again on each node to generate the correct Oracle*Net configuration files.

```
$ORACLE_HOME/apputil/bin/adconfig.sh contextfile=$CONTEXT_FILE
```

7. Update the environment file on each database node

- a. on sclczdb01 add the following line to [SOLXTTS.env](#):

```
source $ORACLE_HOME/SOLXTTS1_sclczdb01.env
```

- b. on sclczdb02 add the following line to [SOLXTTS.env](#):

```
source $ORACLE_HOME/SOLXTTS2_sclczdb02.env
```

8. Run AutoConfig on the Application Tier Nodes

- a. Edit SID=<Instance 1> and PORT=<New listener port > in \$TNS_ADMIN/tnsnames.ora file, to set up a connection to one of the instances in the Oracle RAC environment.
- b. Verify the connection


```
sqlplus system/welcome1
```
- c. Edit the context variable jdbc_url in \$CONTEXT_FILE , changing the HOST, PORT, SERVICE_NAME and adding the INSTANCE_NAME to the CONNECT_DATA parameter.

Before

```
<jdbc_url
oa_var="s_apps_jdbc_connect_descriptor">jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(LOAD_BALANCE=YES) (FAILOVER=YES) (ADDRESS=(PROTOCOL=tcp) (HOST=bu
r407-
25.us.oracle.com) (PORT=1523))) (CONNECT_DATA=(SERVICE_NAME=VIS))</jdbc_url
>
```

After

```
<jdbc_url
oa_var="s_apps_jdbc_connect_descriptor">jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(LOAD_BALANCE=YES) (FAILOVER=YES) (ADDRESS=(PROTOCOL=tcp) (HOST=sc
lcz01-vip.us.oracle.com) (PORT=1524)) (ADDRESS=(PROTOCOL=tcp) (HOST=slcz02-
vip.us.oracle.com) (PORT=1524))) (CONNECT_DATA=(SERVICE_NAME=
SOLXTTS))</jdbc_url>
```

1. Run Autoconfig


```
$INST_TOP/admin/scripts/adautoconfig.sh
```

9. Gather statistics for SYS schema

- a. Copy \$APPL_TOP/admin/adstats.sql from the administration server node to the database server node work directory,

```
cp $APPL_TOP/admin/adstats.sql /ebs/expimp
```

- b. Run the script

```
sqlplus "/" as sysdba"
alter system enable restricted session;
@adstats.sql
```

```
sqlplus "/" as sysdba"
alter system disable restricted session;
```

10. 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.

11. Create ConText and AZ objects

Certain ConText objects are not preserved by the import process. The [consolidated export/import utility patch](#) that you applied to the source administration server node in the “[Preparing the Source-system](#)” section 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.

a. Enable Maintenance Mode

Run AD Administration (adadmin) on the target administration server node.

```
Select "5. Change Maintenance Mode" / "1. Enable Maintenance Mode"
```

b. Generate the driver file

```
perl $AU_TOP/patch/115/bin/dpost_imp.pl ctx_az_obj.driv 11
```

c. Apply the patch

Run AutoPatch (adpatch) to apply it on the target administration server node.

12. 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/
SQL> exec icx_cat_sqe_pvt.sync_sqes_for_all_zones;
```

13. Compile invalid objects

On the target database server node, connect to the target database as SYS and run the `$_ORACLE_HOME/rdbms/admin/utlrp.sql` script to compile invalid objects.

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

14. Re-create the XLA Packages

If you ran recommended step, “[Drop Large XLA Packages \(recommended\)](#)”, and dropped the XLA product rule packages prior to the export, then run this step.

1. Copy \$XLA_TOP/patch/115/sql/xla6128278.sql from the administration server node to the database server node.

```
$ cp $XLA_TOP/patch/115/sql/xla6128278.sql /ebs/expimp
```

2. Use SQL*Plus to connect to the database as APPS and run the xla6128278.sql script to re-create the XLA packages that were dropped prior to the export. The output file name is passed as a parameter.

```
$ sqlplus apps/apps @xla6128278.sql xla6128278.lst
```

15. Maintain Applications database objects

Run AD Administration (adadmin) on the target administration server node. From the “AD Administration Main Menu” menu, perform the following tasks:

- a. **Compile flexfield data in AOL tables**

Select “3. Compile/Reload Applications Database Entities menu” / “3. Compile flexfields”

- b. **Re-create grants and synonyms for the APPS schema**

Select “4. Maintain Applications Database Entities menu” / “2. Re-create grants and synonyms for APPS schema”

- c. **Compile the APPS schema**

Select “3. Compile/Reload Applications Database Entities menu” / “1. Compile APPS schema”

- d. **Disable Maintenance Mode**

Select “5. Change Maintenance Mode” / “2. Disable Maintenance Mode”

16. Validate the APPS Schema

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

- a. **Run the “Validate APPS Schema” utility**

Select “4. Maintain Applications Database Entities menu” / “1. Validate APPS schema”

- b. **Review the output file and make repairs as necessary.**

The output file from the “Validate APPS Schema” utility will be at

```
$APPL_TOP/admin/`echo $CONTEXT_NAME | cut -d _ -f1`/out/APPS.lst.
```

Review that file and follow the instructions for any invalid objects identified.

17. Start Applications server processes

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

```
$INST_TOP/admin/scripts/adstrtal.sh
```

18. Create DQM indexes

Create DQM indexes by following these steps:

a. Log on to Oracle Applications with the "Trading Community Manager" responsibility

Login with "operations/welcome" and then select "Trading Community Manager"

b. Click Control > Request > Run

c. Select "Single Request" option

d. Enter "DQM Staging Program" in the "Name" field

e. Enter the following parameters:

- Number of Parallel Staging Workers: 4
- Staging Command: CREATE_INDEXES
- Continue Previous Execution: NO
- Index Creation: SERIAL

f. Click "Submit"

Further Configuration

Set Up Parallel Concurrent Processing

To set up Parallel Concurrent Processing (PCP), you must have more than one Concurrent Processing node in your environment. If you do not have this, follow the appropriate instructions in MOS ID [406982.1](#), “*Cloning Oracle Applications Release 12 with Rapid Clone.*”

The steps for setting up PCP are documented in section [3.9 Configure Parallel Concurrent Processing](#) of “*Using Oracle 11g Release 2 Real Application Clusters with Oracle E-Business Suite Release 12*”, [MOS ID [823587.1](#)].

Set Up Load Balancing

Load balancing can be defined as the technology and associated processes that distribute a website's traffic among several machines in a cluster using a network-based device called a *server load balancer*. Cluster architectures allow all members of a group of machines to run the same web application, appearing to the outside world as a single machine.

A server load balancer intercepts traffic destined for a site, and redirects it to various machines in the cluster according to the particular load-balancing method chosen. This optimizes use of system resources, resulting in higher performance, availability and scalability. For details on configuring a load balancer with the Oracle E-Business Suite see the following references:

- MOS ID [727171.1](#) - Implementing Load Balancing On Oracle E-Business Suite - Documentation For Specific Load Balancer Hardware
- MOS ID [380489.1](#) - Using Load-Balancers with Oracle E-Business Suite Release 12. (this is linked to in MOS ID [727171.1](#))
- [Oracle E-Business Suite 12 \(BIG-IP LTM, WebAccelerator, FirePass\)](#)

Conclusion

A successful migration of Oracle E-Business Suite to Oracle Exadata Database Machine is dependent upon thorough preparation that includes planning, testing, and review of all the steps involved. The intent of this paper was to illustrate the steps to migrate an existing Oracle E-Business Suite R12 installation's release 11.2 database to Oracle Exadata Database Machine using Cross-platform Transportable Tablespaces (XTTS). Currently this process is only supported for Oracle E-Business Suite R12 source systems with an 11.2 database. The strategy of migrating your E-Business Suite database to Oracle Exadata Database Machine using Transportable Tablespace, and implementing Oracle RAC and Oracle ASM, may introduce a number of features not previously used in your environment. Thorough testing is required.

Any improvements to this process will be updated in this paper and reposted to the Maximum Availability Architecture website at <http://www.oracle.com/goto/maa>.

Appendix

Test Environment Details

The hardware and software details for the test environment are as follows:

Source Database System

Single node, single instance install

- Name: bur407-25
- Solaris 10, SunOS b5.10 Generic_142909-11, 64-bit
- Oracle Enterprise Edition 11.0.1
- ORACLE_HOME= /oracle/11.2.0
- DB_NAME=VIS
- ORACLE_SID=VIS
- Database datafile location = /oracle/data

Application Tier Node

- Name: haovm021.us.oracle.com
- Oracle Enterprise Linux 2.6.18-128.0.0.0.2. x86_64
- E-Business Suite Version 12.1.1
- ORACLE_HOME /ebs/VIS/apps/tech_st/10.1.2
- \$APPL_TOP /ebs/VIS/apps/apps_st/appl
- \$INST_TOP /ebs/VIS/inst/apps/VIS_haovm021
- NFS mounted storage from dscbbg03:/export2/ on /ebs

Target Database System

Oracle Exadata Database Machine quarter rack

- Compute Nodes
 - Names: sclczdb01 and sclczdb02
 - IP Addresses
 - sclczdb01 10.204.74.168
 - sclcz01-vip 10.204.77.181
 - sclczdb02 10.204.74.169
 - sclcz02-vip 10.204.77.182
 - Database ORACLE_HOME /u01/app/oracle/product/11.2.0.2/dbhome_2

- ORACLE_SID=SOLXTTS1 and SOLXTTS2 respectively
 - DB_NAME=SOLXTTS
 - Grid ORACLE_HOME /u01/app/11.2.0/grid
 - ASM ORACLE_SID=+ASM1 and +ASM2 respectively
 - ASM Disk groups: +DATA and +RECO
- DB_NAME=SOLXTTS
- Each with:
- Oracle Enterprise Linux 2.6.18-194.3.1.0.3.el5 x86_64
 - Oracle Enterprise Edition 11.2.0.2
 - 2 Quad-Core Intel Xeon® E5540 Processors (2.53 GHz)
 - 72 GB memory
 - Disk Controller HBA with 512MB Battery Backed Write Cache
 - 4 x 146 GB SAS 10,000 RPM disks
 - Dual-Port QDR InfiniBand Host Channel Adapter
 - 4 Embedded Gigabit Ethernet Ports
- Storage:
 - 3 Exadata Cells (sclczcel01-03)
- Each cell with:
- 2 Quad-core Intel Xeon E5540 (2.53GHz) processors
 - Exadata Smart Flash Cache 384 GB
 - 12 x 1TB 7,200 RPM SATA drives
 - NFS mounted storage from dscbbg03:/export2/ on /ebs

Target Database Nodes Environment Files

sclczdb01 SOLXTTS.env

```
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/11.2.0.2/dbhome_2
export ORACLE_SID=SOLXTTS1
export
PATH=$ORACLE_HOME/bin:./usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/oracle/bin:$ORACLE_HOME/OPatch
```

```
# Add this/uncomment after running autoconfig on the new database node
#source $ORACLE_HOME/SOLXTTS1_sclczdb01.env
```

sclczdb02 SOLXTTS.env

```
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/11.2.0.2/dbhome_2
export ORACLE_SID=SOLXTTS2
export
PATH=$ORACLE_HOME/bin:./usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/oracle/bin:$ORACLE_HOME/OPatch
```

```
# Add this/uncomment after running autoconfig on the new database node
#source $ORACLE_HOME/SOLXTTS2_sclczdb02.env
```

Target Database Starting Parameter File

The source of this file was the single instance source database system. The file was then modified to accommodate the new database name of SOLXTTS, Oracle ASM and Oracle RAC setup prior to the database creation. The CLUSTER_DATABASE parameter is commented out for database creation. The changes from the original source file are highlighted in red and also have a comment for the “Old setting” and the “New target”.

```
*._b_tree_bitmap_plans=FALSE
*._disable_fast_validate=TRUE
*._fast_full_scan_enabled=FALSE
*._index_join_enabled=FALSE
*._like_with_bind_as_equality=TRUE
*._optimizer_autostats_job=FALSE
*._sort_elimination_cost_ratio=5
*._sqlexec_progression_cost=2147483647
*._system_trig_enabled=true
*._trace_files_public=TRUE
*.aq_tm_processes=1
# COMPATIBLE must be greater than or equal to the source database
*.compatible='11.2.0.2'
# Old setting
#*.control_files='/ebs/VIS/db/apps_st/data/cntrl01.dbf','/ebs/VIS/db/apps_st/data/cntrl02.dbf','/ebs/VIS/db/apps_st/data/cntrl03.dbf'
# New target
*.control_files='+DATA/SOLXTTS/CONTROLFILE/cntrl01.dbf','+DATA/SOLXTTS/CONTROLFILE/cntrl02.dbf','+RECO/SOLXTTS/CONTROLFILE/cntrl03.dbf'
*.cursor_sharing='EXACT'
*.db_block_checking='FALSE'
*.db_block_checksum='TRUE'
*.db_block_size=8192
*.db_files=512
# Old setting
#*.db_name='VIS'
# New target
*.db_name='SOLXTTS'
*.diagnostic_dest='/u01/app/oracle'
*.dml_locks=10000
*.event=''
*.java_pool_size=209715200
*.job_queue_processes=2
*.log_buffer=10485760
*.log_checkpoint_interval=100000
*.log_checkpoint_timeout=1200
*.log_checkpoints_to_alert=TRUE
*.max_dump_file_size='20480'
*.nls_comp='binary'
*.nls_date_format='DD-MON-RR'
*.nls_language='american'
*.nls_length_semantics='BYTE'
*.nls_numeric_characters=',.'
*.nls_sort='binary'
*.nls_territory='america'
*.olap_page_pool_size=4194304
*.open_cursors=600
*.optimizer_secure_view_merging=FALSE
*.parallel_max_servers=8
```

```

*.parallel_min_servers=0
*.pga_aggregate_target=1073741824
*.plsql_code_type='INTERPRETED'
*.plsql_optimize_level=2
*.processes=200
*.query_rewrite_enabled='true'
*.recyclebin='OFF'
*.sec_case_sensitive_logon=FALSE
*.session_cached_cursors=500
*.sessions=400
*.sga_target=1073741824
*.shared_pool_reserved_size=41943040
*.shared_pool_size=41943040
*.timed_statistics=TRUE
*.undo_management='AUTO'
# Old setting
#*.undo_tablespace='UNDO_TBS'
# Old setting
#*.utl_file_dir='/usr/tmp','/usr/tmp','/ebs/VIS/db/tech_st/11.1.0/appsutil/outbound/VIS_bur407-25','/usr/tmp'
# New target
SOLXTTS1.utl_file_dir='/usr/tmp','/usr/tmp','/u01/app/oracle/product/11.2.0.2/dbhome_2/appsutil/outbound/SOLXTTS_sclczdb01','/usr/tmp'
SOLXTTS2.utl_file_dir='/usr/tmp','/usr/tmp','/u01/app/oracle/product/11.2.0.2/dbhome_2/appsutil/outbound/SOLXTTS_sclczdb02','/usr/tmp'
*.workarea_size_policy='AUTO'
#
# Added for ASM and Oracle Managed Files
*.db_recovery_file_dest='+RECO'
*.db_recovery_file_dest_size=500G
*.db_create_file_dest='+DATA'
# RAC Additions
#*.cluster_database=true
SOLXTTS1.instance_number=1
SOLXTTS2.instance_number=2
SOLXTTS1.local_listener='(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=sclcz01-vip)(PORT=1524))))'
SOLXTTS2.local_listener='(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=sclcz02-vip)(PORT=1524))))'
SOLXTTS1.thread=1
SOLXTTS2.thread=2
SOLXTTS1.undo_tablespace='UNDO_TBS1'
SOLXTTS2.undo_tablespace='UNDO_TBS2'
# Added for issues with NFS for import, MOS ID 739570.1
*.event="10298 trace name context forever, level 32"

```

Initial listener.ora Files

Node 1

File Name: **\$ORACLE_HOME/network/admin/listener.ora file**

```

LISTENER_SOLXTTS=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=IPC)(KEY=LISTENER_SOLXTTS)))) # line added by Agent
ENABLE_GLOBAL_DYNAMIC_ENDPOINT_LISTENER_SOLXTTS=ON # line added by Agent
#
SID_LIST_LISTENER_SOLXTTS =
  (SID_LIST =
    (SID_DESC =
      (ORACLE_HOME= /u01/app/oracle/product/11.2.0.2/dbhome_2)
      (SID_NAME = SOLXTTS1)
    )
  )
)

```

Node 2

File Name: `$ORACLE_HOME/network/admin/listener.ora file`

```
LISTENER_SOLXTTTS=(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=IPC)(KEY=
LISTENER_SOLXTTTS)))) # line added by Agent
ENABLE_GLOBAL_DYNAMIC_ENDPOINT_LISTENER_SOLXTTTS=ON # line added by
Agent
#
SID_LIST_LISTENER_SOLXTTTS =
  (SID_LIST =
    (SID_DESC =
      (ORACLE_HOME= /u01/app/oracle/product/11.2.0.2/dbhome_2)
      (SID_NAME = SOLXTTTS2)
    )
  )
```

Timestamp Script

This script can be used to prepend a timestamp to a file while it is being generated. This file is named tailTime.

```
#!/bin/bash
# tail of a file and add a timestamp to the output
#
export logfile=$1
tail -f $logfile | (
while true; do
read var; printf "%s " `date '+%y%m%d %H:%M:%S'` $var;
echo
done
)
echo "done!"
```

Generated SQL Files

These scripts were generated on the source system using the `$AU_TOP/patch/115/sql/auxttspre.sql` script.

Modified aucrdb.sql

This file is used to create the target database in section, “[Edit the target database instance creation script aucrdb.sql](#).” The addition of ‘AUTOEXTEND ON’ and the creation of the SYSTEM and SYSAUX datafiles with the default AUTOALLOCATE were changes made from the generated aucrdb.sql file.. The change of the datafile paths to the Oracle ASM ‘+DATA’ disk group with the DB_NAME of SOLXTTTS was done with a file editor. Note that the redo log file size is dependent upon your transaction volume and that a general guideline is to switch logs at most once every twenty minutes. The sizes that are used when creating aucrdb.sql come from the source database. See MOS ID [781999.1](#), “*General Guideline For Sizing The Online Redo Log Files*” for further detail.

```
REM BEFORE RUNNING THE SCRIPT YOU MUST REVIEW
REM IT FOR THE FOLLOWING:
REM - Database Characterset information
```

```

REM      - Location and size of the data files
REM      - Undo Tablespace information
REM      - Temporary Tablespace information
CREATE DATABASE
MAXDATAFILES 512
CHARACTER SET UTF8
NATIONAL CHARACTER SET AL16UTF16
LOGFILE
GROUP 1 ('+DATA','+RECO') SIZE 1024M,
GROUP 2 ('+DATA','+RECO') SIZE 1024M,
GROUP 3 ('+DATA','+RECO') SIZE 1024M
DATAFILE
'+DATA' SIZE 21000M AUTOEXTEND ON
SYSAUX DATAFILE
'+DATA' SIZE 2048M AUTOEXTEND ON
UNDO TABLESPACE "UNDO_TBS1"
DATAFILE
'+DATA' SIZE 18000M AUTOEXTEND ON
EXTENT MANAGEMENT LOCAL
DEFAULT TEMPORARY TABLESPACE "TEMP2"
TEMPFILE
'+DATA' SIZE 10000M AUTOEXTEND ON
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1M
/
CREATE TEMPORARY TABLESPACE "TEMP1"
TEMPFILE
'+DATA' SIZE 10000M AUTOEXTEND ON
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1M
/
CREATE UNDO TABLESPACE "UNDO_TBS2"
DATAFILE
'+DATA' SIZE 18000M AUTOEXTEND ON
EXTENT MANAGEMENT LOCAL
/
ALTER TABLESPACE "TEMP2" TABLESPACE GROUP "TEMP";
ALTER TABLESPACE "TEMP1" TABLESPACE GROUP "TEMP";
ALTER DATABASE DEFAULT TEMPORARY TABLESPACE "TEMP1";
prompt
prompt Ignore any errors related to the setting
prompt of the default temporary tablespace
prompt

```

auxttsread.sql

```

REM
REM Ensure that you have no Apps processes running
REM before running this script
REM
connect / as sysdba;
alter tablespace APPS_CALCLIP read only;
alter tablespace APPS_OMO read only;
alter tablespace APPS_TS_ARCHIVE read only;
alter tablespace APPS_TS_DISCO read only;
alter tablespace APPS_TS_DISCO_OLAP read only;
alter tablespace APPS_TS_INTERFACE read only;
alter tablespace APPS_TS_MEDIA read only;
alter tablespace APPS_TS_NOLOGGING read only;
alter tablespace APPS_TS_QUEUES read only;
alter tablespace APPS_TS_SEED read only;
alter tablespace APPS_TS_SUMMARY read only;
alter tablespace APPS_TS_TOOLS read only;
alter tablespace APPS_TS_TX_DATA read only;
alter tablespace APPS_TS_TX_IDX read only;

```

```
alter tablespace B2B_DT read only;
alter tablespace B2B_IDX read only;
alter tablespace B2B_LOB read only;
alter tablespace B2B_RT read only;
alter tablespace BAM read only;
alter tablespace CTXSYS read only;
alter tablespace CWMLITE read only;
alter tablespace DCM read only;
alter tablespace DISCO_PTM5_CACHE read only;
alter tablespace DISCO_PTM5_META read only;
alter tablespace DSGATEWAY_TAB read only;
alter tablespace GEOR_TBS read only;
alter tablespace HTMLDB read only;
alter tablespace IAS_META read only;
alter tablespace MTR read only;
alter tablespace NOETIX read only;
alter tablespace NON_APPS_TS_DATA read only;
alter tablespace OCATS read only;
alter tablespace ODM_DATA read only;
alter tablespace OLTS_ATTRSTORE read only;
alter tablespace OLTS_BATTRSTORE read only;
alter tablespace OLTS_CT_STORE read only;
alter tablespace OLTS_DEFAULT read only;
alter tablespace OLTS_SVRMGSTORE read only;
alter tablespace OPMOR read only;
alter tablespace ORABPEL read only;
alter tablespace OWAPUB read only;
alter tablespace PORTAL read only;
alter tablespace PORTAL_DOC read only;
alter tablespace PORTAL_IDX read only;
alter tablespace PORTAL_LOG read only;
alter tablespace RE read only;
alter tablespace SYNCSEVER read only;
alter tablespace UDDISYS_TS read only;
alter tablespace WCRSYS_TS read only;
alter tablespace XDB read only;
```

auxttwrite.sql

```
REM
REM Ensure that you have no Apps processes running
REM before running this script
REM
connect / as sysdba;
alter tablespace APPS_CALCLIP read write;
alter tablespace APPS_OMO read write;
alter tablespace APPS_TS_ARCHIVE read write;
alter tablespace APPS_TS_DISCO read write;
alter tablespace APPS_TS_DISCO_OLAP read write;
alter tablespace APPS_TS_INTERFACE read write;
alter tablespace APPS_TS_MEDIA read write;
alter tablespace APPS_TS_NOLOGGING read write;
alter tablespace APPS_TS_QUEUES read write;
alter tablespace APPS_TS_SEED read write;
alter tablespace APPS_TS_SUMMARY read write;
alter tablespace APPS_TS_TOOLS read write;
alter tablespace APPS_TS_TX_DATA read write;
alter tablespace APPS_TS_TX_IDX read write;
alter tablespace B2B_DT read write;
alter tablespace B2B_IDX read write;
alter tablespace B2B_LOB read write;
alter tablespace B2B_RT read write;
alter tablespace BAM read write;
```

```

alter tablespace CTXSYS read write;
alter tablespace CWMLITE read write;
alter tablespace DCM read write;
alter tablespace DISCO_PTM5_CACHE read write;
alter tablespace DISCO_PTM5_META read write;
alter tablespace DSGATEWAY_TAB read write;
alter tablespace GEOR_TBS read write;
alter tablespace HTMLDB read write;
alter tablespace IAS_META read write;
alter tablespace MTR read write;
alter tablespace NOETIX read write;
alter tablespace NON_APPS_TS_DATA read write;
alter tablespace OCATS read write;
alter tablespace ODM_DATA read write;
alter tablespace OLTS_ATTRSTORE read write;
alter tablespace OLTS_BATTRSTORE read write;
alter tablespace OLTS_CT_STORE read write;
alter tablespace OLTS_DEFAULT read write;
alter tablespace OLTS_SVRMGSTORE read write;
alter tablespace OPMOR read write;
alter tablespace ORABPEL read write;
alter tablespace OWAPUB read write;
alter tablespace PORTAL read write;
alter tablespace PORTAL_DOC read write;
alter tablespace PORTAL_IDX read write;
alter tablespace PORTAL_LOG read write;
alter tablespace RE read write;
alter tablespace SYNCSEVER read write;
alter tablespace UDDISYS_TS read write;
alter tablespace WCRSYS_TS read write;
alter tablespace XDB read write;

```

audeftbsp.sql

```

connect / as sysdba;
ALTER USER ORACLE_OCM DEFAULT TABLESPACE SYSTEM;
ALTER USER APPQOSSYS DEFAULT TABLESPACE SYSAUX;
ALTER USER XS$NULL DEFAULT TABLESPACE SYSTEM;
ALTER USER ORDDATA DEFAULT TABLESPACE SYSAUX;
ALTER USER SPATIAL_WFS_ADMIN_USR DEFAULT TABLESPACE SYSTEM;
ALTER USER SPATIAL_CSW_ADMIN_USR DEFAULT TABLESPACE SYSTEM;
ALTER USER OTA DEFAULT TABLESPACE APPS_TS_TX_DATA;
ALTER USER MGDSYS DEFAULT TABLESPACE SYSTEM;
ALTER USER CTXTTEST DEFAULT TABLESPACE SYSTEM;
...
...
ALTER USER DDR DEFAULT TABLESPACE APPS_TS_TX_DATA;
ALTER USER INL DEFAULT TABLESPACE APPS_TS_TX_DATA;
ALTER USER APEX_PUBLIC_USER DEFAULT TABLESPACE SYSTEM;
ALTER USER APEX_030200 DEFAULT TABLESPACE HTMLDB;

```

Data Pump Parameter Files

Export Parameter Files

auexpcall.dat

```

directory=dmpdir
dumpfile=xttsCallout%U.dmp

```

```
full=y
filesize=1048576000
logfile=expCallout.log
include=PRE_SYSTEM/SYSTEM_CALLOUT, SCHEMA/SCHEMA_CALLOUT, SCHEMA/TABLE/INSTANCE_
CALLOUT
```

auexpcallclean.dat

```
directory=dmpdir
dumpfile=xttscall%U.dmp
full=Y
filesize=1048576000
logfile=expcallclean.log
metrics=y
include=POST_SYSTEM/SYSTEM_CALLOUT, POST_SCHEMA/SCHEMA_CALLOUT,
POST_INSTANCE/INSTANCE_CALLOUT
```

auexpxtts.dat

```
directory=dmpdir
dumpfile=xttsmeta%U.dmp
filesize=1048576000
logfile=expxtts.log
transport_tablespaces=
APPS_CALCLIP,
APPS_OMO,
APPS_TS_ARCHIVE,
APPS_TS_DISCO,
APPS_TS_DISCO_OLAP,
APPS_TS_INTERFACE,
APPS_TS_MEDIA,
APPS_TS_NOLOGGING,
APPS_TS_QUEUES,
APPS_TS_SEED,
APPS_TS_SUMMARY,
APPS_TS_TOOLS,
APPS_TS_TX_DATA,
APPS_TS_TX_IDX,
B2B_DT,
B2B_IDX,
B2B_LOB,
B2B_RT,
BAM,
CTXSYS,
CWMLITE,
DCM,
DISCO_PTM5_CACHE,
DISCO_PTM5_META,
DSGATEWAY_TAB,
GEOR_TBS,
HTMLDB,
IAS_META,
MTR,
NOETIX,
NON_APPS_TS_DATA,
OCATS,
```

ODM_DATA,
OLTS_ATTRSTORE,
OLTS_BATTRSTORE,
OLTS_CT_STORE,
OLTS_DEFAULT,
OLTS_SVRMGSTORE,
OPMOR,
ORABPEL,
OWAPUB,
PORTAL,
PORTAL_DOC,
PORTAL_IDX,
PORTAL_LOG,
RE,
SYNCSERVER,
UDDISYS_TS,
WCRSYS_TS,
XDB

auexpsys.dat

```
directory=dmpdir
dumpfile=xttssys%U.dmp
filesize=1048576000
logfile=expsys.log
SCHEMAS=
SI_INFORMTN_SCHEMA,
OLAPSYS,
ORDPLUGINS,
SPATIAL_WFS_ADMIN_USR,
SPATIAL_CSW_ADMIN_USR,
ORDSYS,
XDB,
EXFSYS,
DIP,
ORDDATA,
ORACLE_OCM,
DBSNMP,
XS$NULL,
APPQOSSYS,
MDSYS,
APEX_PUBLIC_USER,
ANONYMOUS,
MDDATA,
SYSTEM,
SYS,
PTG,
SYSMAN,
MGMT_VIEW,
CTXTEST,
SIZZLE_DDR,
MGDSYS,
CTXSYS,
REPADMIN,
AOLDEMO,
OUTLN,
```

```

PROJMFG,
PTE,
HTMLDB_PUBLIC_USER,
IP,
D4OSYS,
TRACESVR,
DMSYS,
CSDUMMY,
DLD,
EDGE,
AD_MONITOR,
EM_MONITOR

```

auexpglb.dat

```

file=gtab.dmp
filesize=1048576000
log=expgtab.log
tables=(
'EXFSYS.EXF$JAVAMSG',
'ORDDATA.ORDDCM_DOCS_TMP',
'ORDDATA.ORDDCM_STD_ATTRS_TMP',
'ORDDATA.ORDDCM_PRV_ATTRS_TMP',
'ORDDATA.ORDDCM_DICT_ATTRS_TMP',
'ORDDATA.ORDDCM_DOC_REFS_TMP',
...
...
'SYSMAN.MGMT_V_RAC_ASSOC_MEMBER_LIST',
'SYSMAN.MGMT_V_CLUSTER_RAC_POL_DETAIL',
'APEX_030200.WWV_FLOW_LOV_TEMP',
'APEX_030200.WWV_FLOW_TEMP_TABLE',
'APEX_030200.WWV_FLOW_TEMP_TREES')

```

auexpmsc.dat

```

directory=dmpdir
dumpfile=xttsmsc%U.dmp
full=y
filesize=1048576000
logfile=expmsc.log
metrics=y
exclude=TABLE, INDEX, CONSTRAINT, COMMENT, RLS_POLICY, MATERIALIZED_VIEW,
MATERIALIZED_VIEW_LOG, SCHEMA_CALLOUT

```

Import Parameter Files

-aurman.dat

```

CONVERT datafile
'/ebs/VIS/db/apps_st/data/apps_ts_interface8.dbf',
'/ebs/VIS/db/apps_st/data/apps_ts_interface15.dbf',
'/ebs/VIS/db/apps_st/data/apps_ts_interface16.dbf',
...
...
'/ebs/VIS/db/apps_st/data/tx_data29.dbf',
'/ebs/VIS/db/apps_st/data/tx_data38.dbf'

```

```
FROM PLATFORM 'Solaris[tm] OE (64-bit)'
PARALLELISM 8
DB_FILE_NAME_CONVERT '/ebs/stage_datafiles/', '+DATA/XTTS_STAGE'
```

auimpmsc1.dat

```
directory=dmpdir
dumpfile=xttsmsc%U.dmp
full=y
logfile=impmsc1.log
EXCLUDE=PROCOBJ,RLS_CONTEXT,RLS_GROUP,TABLESPACE
metrics=y
remap_tablespace=
APPS_CALCLIP:SYSTEM,
APPS_OMO:SYSTEM,
APPS_TS_ARCHIVE:SYSTEM,
APPS_TS_DISCO:SYSTEM,
APPS_TS_DISCO_OLAP:SYSTEM,
APPS_TS_INTERFACE:SYSTEM,
APPS_TS_MEDIA:SYSTEM,
...
...
TS_SIM:SYSTEM,
TS_SIM_X:SYSTEM,
UDDISYS_TS:SYSTEM,
WCRSYS_TS:SYSTEM
```

auimpmsc2.dat

```
directory=dmpdir
dumpfile=xttsmsc%U.dmp
full=y
logfile=impmsc2.log
INCLUDE=PROCOBJ,PRE_SYSTEM_ACTIONS,RLS_CONTEXT,RLS_GROUP
metrics=y
```

auimpxtts.dat

This file will require modification to use the correct staging directory for the staged data files. In this paper example the datafiles were staged to the Oracle ASM location at +DATA/STAGE_VISXTTS. Thus, the generated auimpxtts.dat file had to be edited to do a change all on the path with the outcome looking like the following:

```
directory=dmpdir
dumpfile=xttsmeta%U.dmp
logfile=impxtts.log
exclude=TYPE
transport_datafiles=
'+DATA/STAGE_VISXTTS/apps_calclip_81.dbf',
'+DATA/STAGE_VISXTTS/apps_calclip_82.dbf',
...
'+DATA/STAGE_VISXTTS/ts_sim_x_215.dbf',
'+DATA/STAGE_VISXTTS/uddisys_ts_172.dbf',
```

```
'+DATA/STAGE_VISXTTS/wcrsys_ts_164.dbf'
```

auimpglb.dat

```
file=gtab.dmp
filesize=1048576000
full=y
log=impgtab.log
```

auimpsys.dat

```
directory=dmpdir
dumpfile=xttssys%U.dmp
full=y
logfile=impsys.log
METRICS=Y
```

Other SQL Files and Scripts

These files are used for various other purposes described in the detailed steps.

tts_system_user_obj.sql

```
select owner, segment_name, segment_type
from dba_segments
where tablespace_name in ('SYSTEM', 'SYSAUX')
and owner not in (select name
                  from system.logstdby$skip_support
                  where action=0);
```

genSCPsh.sql

```
REM Using the generated shell script requires password-less scp to be set up
REM
set pages 0 head off feedback off lines 170 echo off trim on verify off
termout off
```

```
spool scpDataFiles.sh
select '#!/bin/ksh' || CHR(13)||CHR(10)|| '#' from dual;
spool off
REM
REM Add the target host and target directory here
define TARGET_HOST=sclczdb01
define TARGET_DIR=/ebs/stage_datafiles/
spool scpDataFiles.sh append
select 'scp ' || b.file_name || ' ' || '&TARGET_HOST' || ':' || '&TARGET_DIR'
from dba_tablespaces a, dba_data_files b
where a.tablespace_name not in
      (select unique (default_tablespace)
       from dba_users where username in
        ('ORDSYS', 'MDSYS', 'CTXSYS', 'ORDPLUGINS', 'LBACSYS',
         'XDB', 'SI_INFORMTN_SCHEMA', 'DIP', 'DMSYS',
         'DBSNMP', 'EXFSYS', 'WMSYS', 'ORACLE_OCM'))
and a.tablespace_name not in ('SYSTEM', 'SYSAUX')
and a.contents <> 'UNDO';

spool off
```

```
host chmod 755 scpDataFiles.sh
exit
```

genRMANconvert.sql

```
set head off pages 0 feedback off verify off echo off lines 150 termout off
spool aurman.dat
REM This is the V$DATABASE:PLATFORM_NAME from the source system
define SOURCE_PLATFORM="Solaris[tm] OE (64-bit)"
REM Directory where datafiles are staged or if NFS mounted
define SOURCE_DIR="/ebs/stage_datafiles"
REM Target staging area in ASM
define ASM_TARGET="+DATA/XTTS_STAGE"
REM
REM
select 'CONVERT datafile' from dual;
spool off
REM Generate the list of datafiles enclosed in quotes
spool RMANconvert.tmp
select chr(39) || FILE_NAME || chr(39) || ','
from dba_tablespaces a, dba_data_files b
where a.tablespace_name = b.tablespace_name
and a.tablespace_name not in
    (select unique (default_tablespace)
     from dba_users where username in
      ('ORDSYS', 'MDSYS', 'CTXSYS', 'ORDPLUGINS', 'LBACSYS',
       'XDB', 'SI_INFORMTN_SCHEMA', 'DIP', 'DMSYS',
       'DBSNMP', 'EXFSYS', 'WMSYS', 'ORACLE_OCM'))
and a.tablespace_name not in ('SYSTEM', 'SYSAUX')
and a.contents <> 'UNDO'
order by bytes
/
spool off
REM Remove the comma from the last datafile line
host sed '$s/,//' RMANconvert.tmp > RMANconvert.tmp2
host cat RMANconvert.tmp2 >> aurman.dat
host echo "FROM PLATFORM '&SOURCE_PLATFORM'" >> aurman.dat
host echo "PARALLELISM 8" >> aurman.dat
host echo "DB_FILE_NAME_CONVERT '&SOURCE_DIR','&ASM_TARGET';" >> aurman.dat
exit
```

RMAN Convert Script

```
#!/bin/ksh
#
export NLS_LANG=AMERICAN_AMERICA.WE8DEC
export NLS_DATE_FORMAT="MM/DD/YYYY HH24:MI:SS"
export TIME_STAMP=`date +%Y%m%d_%H%M%S`
rman target / << EOF! | tee -a rmanConvert_${ORACLE_SID}_`date
+%Y%m%d_%H%M%S`.l
og
set echo on
show all;
@aurman.dat
exit
```

EOF!

genDropXLA.sql

```
set pages 0 head off feedback off lines 170 echo off trim on verify off
termout off timing on
spool dropXLA_AAD_Pkgs.sql
select distinct('drop package ' || db.owner || '.' || db.object_name ||
';')
from dba_objects db, xla_subledgers xl
where db.object_type='PACKAGE BODY' and db.object_name like 'XLA%AAD%PKG'
and substr(db.object_name,1,9) =
'XLA_' || LPAD(SUBSTR(TO_CHAR(ABS(xl.application_id)), 1, 5), 5, '0')
and db.object_name NOT IN
('XLA_AAD_HDR_ACCT_ATTRS_F_PKG','XLA_AMB_AAD_PKG')
order by 1;
spool off
set timing on
@dropXLA_AAD_Pkgs
```

References

1. Oracle Maximum Availability Architecture Web site
<http://www.otn.oracle.com/goto/maa>
2. Oracle Exadata Database Machine
<http://www.oracle.com/us/products/database/database-machine>
3. A Roadmap for Migrating Oracle E-Business Suite to the Oracle Exadata Database Machine [ID [1133355.1](#)]
4. [Oracle Applications Installation and Upgrade Notes Release 12 \(12.1.1\) for Linux x86-64 \[ID 761566.1\]](#)
5. [Oracle Database Utilities, 11g Release 2 \(11.2\)](#)
6. Interoperability Notes EBS R12 with Database 11gR2 [ID [1058763.1](#)]
7. Oracle E-Business Suite Release 11i & R12 Patches Required with Oracle Database 11g Release 2 (11.2.0) on Exadata [ID [1392527.1](#)]
8. [Using Oracle 11g Release 2 Real Application Clusters with Oracle E-Business Suite Release 12 \[ID 823587.1\]](#)
9. Steven Chan's [E-Business Suite Technology Blog](#)
 - [11gR2 Database Certified with E-Business Suite 11i](#)
 - [11gR2 Database Certified with E-Business Suite 12](#)
10. [Upgrade to 11g Performance Best Practices](#)
11. Oracle 11.2 Upgrade Companion [ID [785351.1](#)]
12. [Configuring DBFS on Oracle Database Machine \[ID 1054431.1\]](#)
13. Database Machine and Exadata Storage Server 11g Release 2 (11.2) Supported Versions [ID [888828.1](#)]
14. Where Can I Find the Latest Version of Opatch? [ID [224346.1](#)]
15. [Database Examples Installation Guide](#)
16. [bde_chk_cbo.sql - EBS initialization parameters - Healthcheck \[ID 174605.1\]](#)
17. Using Oracle 11g Release 2 Real Application Clusters with Oracle E-Business Suite Release 12 [ID [823587.1](#)]
18. Database Initialization Parameters for Oracle Applications Release 12 [ID [396009.1](#)]
19. Database Machine HealthCheck [ID [1070954.1](#)]

Change Record

Date	Summary of Changes
2/6/12	<ul style="list-style-type: none"> Reversed the order of this table Added reference to new MOS note Oracle E-Business Suite Release 11i & R12 Patches Required with Oracle Database 11g Release 2 (11.2.0) on Exadata [ID 1392527.1]
11/25/11	<ul style="list-style-type: none"> Fixed syntax error for missing quote in “alter database add logfile instance ‘SOLXITTS2’ “ under “Set up Oracle RAC” step.
11/10/11	<ul style="list-style-type: none"> Added reference to Using Transportable Tablespaces for EBS Release 12 Using Database 11gR2 [ID 1311487.1] and reworded the Introduciton.
11/2/11	<ul style="list-style-type: none"> Added “Synchronize Oracle Text Indexes” step to the “Preparing the Source-system” section.
10/14/11	<ul style="list-style-type: none"> Added details on modifying the generated aurman.dat file including removal of any old rollback segment tablespace datafiles under the “Convert the endian format of the datafiles” step in the “Exporting the Source Database” section. Added TABLESPACE to the EXCLUDE line in auiimpsc1.dat Removed duplicate SQL from the Importing section in step “Verify all non-system tablespaces and datafiles are imported” Added patch 10185319 to the “Preparing the Source-system” and “Preparing the Target Database” sections. Added conditional step to convert any non-system dictionary-managed tablespaces to locally managed in the “Preparing the Source-system” section under, “Convert Non-SYSTEM Dictionary Managed Tablespaces to Locally Managed.” Moved the “Re-create the XLA Packages” step to after the “Compile invalid objects” step in the “Updating the Imported Database” section.
9/26/11	<ul style="list-style-type: none"> Corrected use of the script under “Re-create the XLA Packages”

Date	Summary of Changes
7/18/11	<ul style="list-style-type: none"> Removed “Purge unused columns” section from the “Exporting the Source Database” section (see below). This was deemed unnecessary for 11g+ databases per a data pump fix that was made in 10.2. <p>Use SQL*Plus to connect to the source database as sysdba and run the following commands to find and purge any unused columns:</p> <pre>SQL> select * from sys.dba_unused_col_tabs; SQL> alter table [table name] drop unused columns;</pre>
4/19/11	<ul style="list-style-type: none"> Updated the query in genDropXLA.sql Added clarification on XLA export step, recommended for an Oracle E-Business Suite release 12 source system
3/24/11	<ul style="list-style-type: none"> Updated the Hugepages MOS note to 361468.1
3/10/2011	<ul style="list-style-type: none"> Under the “Preparing the Target Database” section in step 8, “Create the Target Database,” Added a step to ensure the time zone of the target database matches the source database. Added the following optional steps at the end of the export: <ul style="list-style-type: none"> <i>Modify source tablespaces back to read/write mode (optional)</i> <i>Convert callout data back to metadata (optional)</i> Added rows for auexpcclean.dat and auexpcall.dat to “Table 6 aux ttspre.sql Generated Files.” Added a reference to DBMS_FILE_TRANSFER section of MOS ID 371556.1, “How to Move Tablespaces Across Platforms Using Transportable Tablespaces With RMAN.” Under step 13, “Copy the read only data files to the target” under the “Exporting the Source Database” section. Added a conditional step under the “Updating the Imported Database” section to “Re-create custom database links (conditional)”, now step 11.
2/14/2011	Published



Migrating the Oracle E-Business Suite Database
to Oracle Exadata Machine Using Transportable

Tablespaces

February 2012

Author: Ray Dutcher

Contributing Authors: MAA team, Glen Ong,
Dean Gagne

Editor: Viv Schupmann

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



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2011, Oracle and/or its affiliates. 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 is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.