



How to Upgrade Oracle Software and Database from Oracle Version 10.2.0.4 to 11g

CONFIDENTIAL INFORMATION

The information herein is the property of Ex Libris Ltd. or its affiliates and any misuse or abuse will result in economic loss. DO NOT COPY UNLESS YOU HAVE BEEN GIVEN SPECIFIC WRITTEN AUTHORIZATION FROM EX LIBRIS LTD.

This document is provided for limited and restricted purposes in accordance with a binding contract with Ex Libris Ltd. or an affiliate. The information herein includes trade secrets and is confidential.

DISCLAIMER

The information in this document will be subject to periodic change and updating. Please confirm that you have the most current documentation. There are no warranties of any kind, express or implied, provided in this documentation, other than those expressly agreed upon in the applicable Ex Libris contract. This information is provided AS IS. Unless otherwise agreed, Ex Libris shall not be liable for any damages for use of this document, including, without limitation, consequential, punitive, indirect or direct damages.

Any references in this document to third-party material (including third-party Web sites) are provided for convenience only and do not in any manner serve as an endorsement of that third-party material or those Web sites. The third-party materials are not part of the materials for this Ex Libris product and Ex Libris has no liability for such materials.

TRADEMARKS

“Ex Libris,” the Ex Libris bridge, Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder and LinkFinder Plus, and other marks are trademarks or registered trademarks of Ex Libris Ltd. or its affiliates.

The absence of a name or logo in this list does not constitute a waiver of any and all intellectual property rights that Ex Libris Ltd. or its affiliates have established in any of its products, features, or service names or logos.

Trademarks of various third-party products, which may include the following, are referenced in this documentation. Ex Libris does not claim any rights in these trademarks. Use of these marks does not imply endorsement by Ex Libris of these third-party products, or endorsement by these third parties of Ex Libris products.

Oracle is a registered trademark of Oracle Corporation.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

Microsoft, the Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32,

Microsoft Windows, the Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, and Windows NT are registered trademarks and ActiveX is a trademark of the Microsoft Corporation in the United States and/or other countries.

Unicode and the Unicode logo are registered trademarks of Unicode, Inc.

Google is a registered trademark of Google, Inc.

Copyright Ex Libris Limited, 2016. All rights reserved.

Document released: February 22, 2016

Web address: <http://www.exlibrisgroup.com>

Table of Contents

Chapter 1	Pre-Upgrade Activities.....	5
	Prerequisites	5
	Preparing the System for the Upgrade.....	6
	The Pre-Upgrade Information Tool	7
	<i>Database</i>	12
	<i>Logfiles</i>	13
	<i>Tablespaces</i>	13
	<i>Update Parameters</i>	13
	<i>Deprecated Parameters</i>	13
	<i>Obsolete Parameters</i>	13
	<i>Components</i>	13
	<i>Miscellaneous Warnings</i>	14
	<i>Validate Objects</i>	14
	<i>SYSAUX Tablespace</i>	15
	Optimizer Statistics	15
	Manual Changes	16
	Backing Up the Database.....	20
Chapter 2	Upgrading the Database	21
Chapter 3	Post-Upgrade Activities	27
	Migrating the Initialization Parameter File to a Server Parameter File	27
	Setting Parameters to Ignore Case-Sensitivity	28
	Adjusting the Initialization Parameter File for the New Release	28
	Changing the Database global_name	29
	Configuring Enterprise Manager	30
	Installing Oracle JVM.....	31
	Installing Oracle XMLDB and the DBMS_CUBE_EXP Package.....	34
	Collecting Statistics for the Application Schemas and Objects.....	36
	Recreating Roles.....	36
Appendix A	Troubleshooting the Upgrade	37
	Resource Limits.....	37

Component Status	38
Rerunning the Upgrade.....	38
Appendix B Configuration Files.....	41

1

Pre-Upgrade Activities

This section includes:

- **Prerequisites** on page 5
- **Preparing the System for the Upgrade** on page 6
- **The Pre-Upgrade Information Tool** on page 7
- **Optimizer Statistics** on page 15

Prerequisites

The upgrade of the Oracle software and database must be carried out by a local DBA only. All recommendations below are for the standard Oracle installation performed in the `/exlibris/app/oracle/product/11` directory. An upgrade to a non-standard installation must be done using the standard Oracle patch set and is the local DBA's responsibility.

Before starting the upgrade, a full backup of all Oracle 10 software and databases is absolutely necessary.

The database needs to be upgraded to version 10.2.0.4 before the upgrade to 11.1 is performed. If the database is still in version 10.2.0.3, refer to the document *How to Upgrade Oracle Software and Database from Oracle Version 10.2.0.1, 10.2.0.3 to 10.2.0.4*.

It is highly recommended to perform and test the upgrade on a test database before upgrading the production database to Oracle Database 11g Release 1 (11.1).

NOTE:

For additional information for customers who install the database by themselves, refer to the document *Database Installation Requirements – RAC and Single Database*.

Preparing the System for the Upgrade

The following terminology is used in this upgrade:

- `<ORACLE_SID>` refers to the Oracle instance name as defined in the environment variable `ORACLE_SID`.
- `<Oracle11 $ORACLE_HOME>` refers to the path defined in the environment variable `$ORACLE_HOME` that is defined for the `oracle11` user. This is usually `exlibris/app/oracle/product/11`.

To prepare the system for the upgrade:

- 1 Create a new `oracle11` user.

For example, as a root user enter:

```
useradd -u <oracle10 uid> -g <dba group gid> -o -d /
exlibris/app/oracle/product/11 -s /bin/tcsh -c
Oracle11 oracle11

passwd oracle11
```

- 2 Check `oracle10` uid and group `dba` gid from user `oracle10`:

```
id
```

- 3 As an `oracle11` user, extract the `install_ora.tar.gz` file from the Ex Libris FTP server `/oracle/scripts` directory to the `/exlibris/ftp_from_exlibris` directory on the DB Unix server and open the archive file to create the `scripts` directory by entering the following:

```
gzip -d install_ora.tar.gz | tar xvf -
```

- 4 Create the new Oracle 11 home directory:

As an `oracle11` user, extract the Oracle 11 quick installation package from the `/exlibris/ftp` directory to the `/exlibris/app/oracle/product` directory:

```
cd /exlibris/app/oracle/product
```

- For SUN:

```
cat /exlibris/ftp_from_exlibris/
orall_solaris.tar.gz_* | gnu_tar -xzvf -
```

- For Linux 32-bit:

```
cat /exlibris/ftp_from_exlibris/  
ora_11_linux.tar.gz_* | tar -xzvf -
```

- For Linux 64-bit:

```
cat /exlibris/ftp_from_exlibris/  
ora_11_linux64.tar.gz_* | tar -xzvf -
```

- For AIX:

```
cat /exlibris/ftp_from_exlibris/orall_aix.tar.gz_*  
| gnu_tar -xzvf -
```

- 5 Verify that the new created directory 11 under /exlibris/app/oracle/product is owned by oracle:dba. If not, change it:

```
chown -R oracle11:dba 11
```

- 6 As an oracle11 user, change the permissions of the Oracle binary files:

```
chmod +s $ORACLE_HOME/bin/oracle  
cd $ORACLE_HOME  
find . -type d | xargs chmod +x  
find . | xargs chmod +r
```

- 7 Make sure the database to be upgraded is up. If required, start it (as an oracle10 user):

```
ps -ef | grep smon_<ORACLE_SID>
```

If you need to restart it, make sure the ORACLE_SID is set correctly:

```
echo $ORACLE_SID
```

- 8 Start the database:

```
sqlplus '/ as sysdba'  
startup
```

The Pre-Upgrade Information Tool

Analyze the database before upgrading it to the new release. This is done by running the Pre-Upgrade Information Tool.

The Pre-Upgrade Information Tool is a SQL script that ships with Oracle Database 11g Release 1 (11.1) and must be copied to and run from the environment of the database being upgraded.

To run the Pre-Upgrade Information Tool:

- 1 Log on to the system as the owner of the Oracle Database 11g Release 1 (11.1) Oracle home directory.
- 2 Copy the Pre-Upgrade Information Tool (`utlu111i.sql`) from the Oracle Database 11g Release 1 (11.1) `ORACLE_HOME/rdbms/admin` directory to a directory outside of the Oracle home, such as the temporary directory on your system.
- 3 Make a note of the new location of this file.
- 4 Log on to the system as the owner of the Oracle home directory of the database to be upgraded (`oracle10`).
- 5 Change to the directory to which you copied the files in Step 2.
- 6 Start SQL*Plus by connecting to the database instance as a user with SYSDBA privileges:

```
sqlplus '/as sysdba'
```

- 7 Set the system to spool results to a log file for later analysis:

```
SQL> SPOOL upgrade_info.log
```

- 8 Run the Pre-Upgrade Information Tool:

```
SQL> @utlu111i.sql
```

- 9 Turn off the spooling of script results to the log file:

```
SQL> SPOOL OFF
```

- 10 Check the output of the Pre-Upgrade Information Tool in:
`upgrade_info.log`

The following is an example of the output generated by the Pre-Upgrade Information Tool.

```
Oracle Database 11.1 Pre-Upgrade Information Tool    09-
25-2008 08:10:23.
*****
Database:
*****
--> name:          ALEPHTOU
--> version:       10.2.0.4.0
--> compatible:    10.2.0.1.0
--> blocksize:     8192
--> platform:      Linux IA (32-bit)
--> timezone file: V4
*****
Tablespaces: [make adjustments in the current
environment]
*****
--> SYSTEM tablespace is adequate for the upgrade.
.... minimum required size: 752 MB
--> UNDOTBS1 tablespace is adequate for the upgrade.
.... minimum required size: 27 MB
--> SYSAUX tablespace is adequate for the upgrade.
.... minimum required size: 349 MB
.... AUTOEXTEND additional space required: 63 MB
--> TEMP tablespace is adequate for the upgrade.
.... minimum required size: 61 MB
```

```
*****
Update Parameters: [Update Oracle Database 11.1 init.ora
or spfile]
*****
-- No update parameter changes are required.
*****
Renamed Parameters: [Update Oracle Database 11.1
init.ora or spfile]
*****
-- No renamed parameters found. No changes are required.
*****
Obsolete/Deprecated Parameters: [Update Oracle Database
11.1 init.ora or spfile]
*****
--> "background_dump_dest" replaced by
"diagnostic_dest"
--> "user_dump_dest" replaced by "diagnostic_dest"
--> "core_dump_dest" replaced by "diagnostic_dest".
*****
Components: [The following database components will be
upgraded or installed]
*****
```

```
--> Oracle Catalog Views          [upgrade] VALID
--> Oracle Packages and Types     [upgrade] VALID
--> JServer JAVA Virtual Machine [upgrade] VALID
--> Oracle XDK for Java           [upgrade] VALID
--> Oracle Workspace Manager      [upgrade] VALID
--> Oracle Text                   [upgrade] VALID
--> Oracle XML Database           [upgrade] VALID
--> Oracle Java Packages          [upgrade] VALID
--> Oracle interMedia             [upgrade] VALID
--> Data Mining                   [upgrade] VALID
--> Expression Filter             [upgrade] VALID
--> Rule Manager                  [upgrade] VALID
*****
Miscellaneous Warnings
*****
WARNING: --> Database contains stale optimizer
statistics.
.... Refer to the 11g Upgrade Guide for instructions to
update
.... statistics prior to upgrading the database.
.... Component Schemas with stale statistics:
....   SYS
WARNING: --> Database contains INVALID objects prior to
upgrade.
.... USER MDSYS has 1 INVALID objects.
.... USER ALEPH_ADMIN has 1 INVALID objects.
WARNING: --> Database contains schemas with objects
dependent on network packages.
.... Refer to the 11g Upgrade Guide for instructions to
configure Network ACLs.
PL/SQL procedure successfully completed.
```

The timezone version should be updated in the `sys.registry$database` table after running the pre-upgrade script. Enter the following command to check if it was updated:

```
sqlplus '/as sysdba'  
SQL>col PLATFORM_ID form 999  
SQL> col PLATFORM_NAME form a50  
select * from sys.registry$database;
```

PLATFORM_ID	PLATFORM_NAME	EDITION	TZ_VERSION
13	Linux x86	64-bit	4

NOTE:

Make sure that the `TZ_VERSION` is updated to 4.

The following sections provide explanations of each of the sections in the above output:

- [Database](#) on page 12
- [Logfiles](#) on page 13
- [Tablespaces](#) on page 13
- [Update Parameters](#) on page 13
- [Deprecated Parameters](#) on page 13
- [Obsolete Parameters](#) on page 13
- [Components](#) on page 13
- [Miscellaneous Warnings](#) on page 14
- [Validate Objects](#) on page 14
- [SYSAUX Tablespace](#) on page 15

Database

This section displays global database information about the current database, such as the database name, release number, and compatibility level. A warning is displayed if you must adjust the `COMPATIBLE` initialization parameter before the database is upgraded. Any required changes are explained in [Appendix B: Configuration Files](#) on page 41.

Logfiles

This section displays a list of redo log files in the current database whose size is less than 4 MB. For each log file, the file name, group number, and recommended size are displayed. New files of at least 4 MB (preferably 10 MB) must be created in the current database. Any redo log files less than 4 MB must be deleted before the database is upgraded. Any required changes are explained in [Manual Changes](#) on page 16.

Tablespaces

This section displays a list of tablespaces in the current database. For each tablespace, the tablespace name and minimum required size is displayed. In addition, a message is displayed if the tablespace is adequate for the upgrade. If the tablespace does not have enough free space, then space must be added to the tablespace in the current database. Tablespace adjustments must be made before the database is upgraded. Any required changes are explained in [Manual Changes](#) on page 16.

Update Parameters

This section displays a list of initialization parameters in the parameter file of the current database that must be adjusted before the database is upgraded. The adjustments must be made to the parameter file after it is copied to Oracle Database 11g Release 1 (11.1). Any required changes are explained in [Manual Changes](#) on page 16.

Deprecated Parameters

This section displays a list of initialization parameters in the parameter file of the current database that are deprecated in Oracle Database 11g Release 1 (11.1). Any required changes are explained in [Manual Changes](#) on page 16.

Obsolete Parameters

This section displays a list of initialization parameters in the parameter file of the current database that are obsolete in Oracle Database 11g Release 1 (11.1). Obsolete initialization parameters must be removed from the parameter file before the database is upgraded. Any required changes are explained in [Manual Changes](#) on page 16.

Components

This section displays a list of database components in Oracle Database 11g Release 1 (11.1) that are upgraded or installed when the current database is upgraded.

Miscellaneous Warnings

This section displays warnings about specific situations that might require attention before or after the upgrade.

For example:

```
*****
Miscellaneous Warnings
*****
WARNING: --> Database contains stale optimizer
statistics.
.... Refer to the 11g Upgrade Guide for instructions to
.... update statistics prior to upgrading the database.
.... Component Schemas with stale statistics:
....   SYS
WARNING: --> Database contains INVALID objects prior to
upgrade.
..... USER CTXSYS has 1 INVALID objects.
```

Stale optimizer statistics issues are explained in [Optimizer Statistics](#) on page 15.

The handling of objects with the status INVALID is explained in [Manual Changes](#) on page 16.

Validate Objects

Before running the upgrade scripts, make sure no invalid objects exists in the DB.

Verify that all expected packages and classes are valid:

Enter the following as an Oracle10 user:

```
sqlplus '/as sysdba'
select OBJECT_NAME,STATUS from dba_objects where STATUS
!= 'VALID' ;
exit
```

If any objects are INVALID, run the following script:

```
cd $ORACLE_HOME/rdbms/admin
sqlplus '/as sysdba'
set pause off
SQL> @utlrp.sql
```

SYSAUX Tablespace

This section describes the minimum required size for the SYSAUX tablespace, which is required in Oracle Database 11g Release 1 (11.1). The SYSAUX tablespace must be created after the new release is started and before the upgrade scripts are invoked.

Optimizer Statistics

When upgrading to Oracle Database 11g Release 1 (11.1), optimizer statistics are collected for dictionary tables that lack statistics. This statistics collection can be time-consuming for databases with a large number of dictionary tables, so statistics gathering only occurs for those tables that lack statistics or are significantly changed during the upgrade.

To decrease the amount of downtime incurred when collecting statistics, you can collect statistics prior to performing the actual database upgrade.

Use the `DBMS_STATS.GATHER_SCHEMA_STATS` procedure to gather statistics. To do this, you must run several scripts. By running these scripts prior to performing the actual database upgrade, you can decrease the amount of downtime incurred during the database upgrade.

This process should be tested on a test database like the other aspects of the upgrade. Also, some schemas referenced in these scripts may not exist if some database components have not been installed.

The following sample script uses the `DBMS_STATS.GATHER_SCHEMA_STATS` procedure to collect statistics for system component schemas.

To run this script:

- 1 Enter the following as an oracle10 user:

```
vi stats.sql
```

- 2 Click this icon and copy the text to the new file:
- 3 Save the file.

- 4 Connect to the database as an SYSDBA user using SQL*Plus, and run the following script as an oracle10 user:

```
sqlplus '/as sysdba'
grant analyze any to sys;
spool gdict
@stats.sql
spool off
Drop the tables sys.plan_table$ and sys.plan_table,
which will be re-created during the upgrade:
sqlplus '/as sysdba'
drop table sys.plan_table;
drop table sys.plan_table$;
exit
```

NOTE:

Statistics collection displays an error message if a particular component schema does not exist in the database. This can happen if a component is not installed or if it is invalid. You can ignore error messages such as:

```
ORA-20000: Schema "LBACSYS" does not exist or insufficient
privileges
```

Manual Changes

Several manual changes must be done before upgrading the database. First you must compile INVALID objects that appeared on the "Miscellaneous Warnings" section of the pre-upgrade tool.

To compile INVALID objects:

- 1 Run `utlrbp.sql` as an oracle10 user to recompile any remaining stored PL/SQL and Java code.

```
cd $ORACLE_HOME/rdbms/admin
sqlplus '/as sysdba'
Set pause off.
SQL> @utlrbp.sql
```

- 2 Prepare the configuration files. Refer to [Appendix B: Configuration Files](#) on page 41.

3 Prepare the network configuration files

- a Copy the `tnsnames.ora` and `listener.ora` files from the old `$ORACLE_HOME/network/admin` to the same location in the oracle11 `$ORACLE_HOME`, and add all the SIDs definitions and correct `oracle_home` for the upgraded database.
- b Log on as an oracle10 user:

```
cd network/admin
cp tnsnames.ora <oracle11 $ORACLE_HOME>/network/
admin
cp listener.ora <oracle11 $ORACLE_HOME>/network/
admin
```

For example:

```
cp tnsnames.ora /exlibris/app/oracle/product/11/
network/admin
cp listener.ora /exlibris/app/oracle/product/11/
network/admin
```

- 4 Make all other necessary changes required to the tablespace and miscellaneous, as described in [The Pre-Upgrade Information Tool](#) on page 7.
- 5 Resize the tablespace:

- a Log on as an oracle10 user.

```
sqlplus '/as sysdba
```

- b Check the file location.

```
SQL>Col file_name format a80
SQL>Select file_name from dba_data_files where
tablespace_name=<TABLESPACE_NAME>;
```

For example:

```
SQL>select file_name from dba_data_files where
tablespace_name='SYSAUX' ;
```

- c Resize the tablespace:

```
SQL>alter database datafile '<full_path>' resize
xxx;
```

For example:

```
SQL>alter database datafile '/exlibris/oradata/
alephtou/sysaux_01.dbf' resize 350m;
```

If you have more than one datafile per tablespace, you can choose any file you want.

- d** Alter the SYSTEM tablespace to be autoextend. For example:

```
alter database datafile '/exlibris/oradata/ alephtou/
system_01.dbf' autoextend on maxsize 2g;
```

- 6** Check the redo log files size. To check the current size:

```
cd /exlibris/oradata/<ORACLE_SID>
ls -l *redo*
```

- 7** If they are less than 500 MB each, increase their size:

- a** Check the size of the current logs:

```
select group#, bytes, status from v$log;
```

- b** Retrieve all the log member names for the groups:

```
SQL> select group#, member from v$logfile;
```

- c** Create new log groups and name them groups 6, 7, 8, etc.

```
SQL> alter database add logfile group 6
'/exlibris/oradata//<OACLE_SID>/
<OACLE_SID>_redo06.log' size 500M;
```

For example:

```
SQL> alter database add logfile group 6
'/exlibris/oradata/aleph20/aleph20_redo06.log' size
500m;
```

- d** Query to view the v\$log status:

```
SQL> select group#, status from v$log;
```

The following is a sample output:

```
GROUP# STATUS
-----
      1 INACTIVE
      2 CURRENT
      3 INACTIVE
      4 UNUSED
      5 UNUSED
      6 UNUSED
```

- e** Switch until you are in log group 6, so that you can drop log groups 1 to 5:

```
SQL> alter system switch logfile;
```

- f** Repeat as necessary until group 6 is CURRENT. If necessary, use the command:

```
alter system checkpoint;)
```

- g** Run the query again to verify the current log group is group 6:

```
SQL> select group#, status from v$log;
```

The following is a sample output:

```
GROUP# STATUS
-----
      1 INACTIVE
      2 INACTIVE
      3 INACTIVE
      4 INACTIVE
      5 INACTIVE
      6 CURRENT
      7 UNUSED
      8 UNUSED
      9 UNUSED
     10 UNUSED
```

h Drop redo log groups 1 to 5:

```
SQL> alter database drop logfile group 1;
```

i Repeat with groups 2-5.

j Verify that the groups were dropped, and the new groups' sizes are correct using the SQL command listed in Step **a**.

k Go to the operating system and delete the files associated with redo log groups 1 to 5 in Step **b** above as they are no longer required:

```
% rm /exlibris/oradata/<ORACLE_SID>//  
<ORACLE_SID>_redo1.dbf
```

l Repeat with files 2-5.

8 As an oracle10 user, make sure the recycle bin is turned off:

```
sqlplus '/as sysdba'  
SELECT flashback_on FROM gv$database;
```

If the result is:

```
FLASHBACK_ON  
-----  
YES
```

Turn off the recycle bin:

```
ALTER DATABASE FLASHBACK OFF;
```

Backing Up the Database

Perform the following as an oracle10 user to shutdown the Oracle 10 instance:

```
setenv ORACLE_SID <sid>  
sqlplus / as sysdba  
SQL> shutdown immediate
```

IMPORTANT:

After running the Pre-Upgrade Information Tool and cleanly shutting down the instance, perform a full backup of the database.

2

Upgrading the Database

To upgrade the database:

- 1 Shut down oracle10 listener.
- 2 As an oracle10 user, enter the following:

```
lsnrctl stop
```

- 3 Log on to the system as the owner of the Oracle Database 11g Release 1 (11.1) Oracle home directory.
- 4 Verify that ORACLE_SID is set correctly.
- 5 Edit the `.cshrc` file under the Oracle 11 home directory, change the ORACLE_SID, and source the file.
- 6 Verify the following:

- The SID is the correct one:

```
echo $ORACLE_SID
```

- The `oratab` file points to the Oracle Database 11g Release 1 (11.1) Oracle home. The `oratab` file is located under `/etc` on Linux and AIX, and under `/var/opt/oracle` on Solaris.
- The file contains a line such as the following:

```
Aleph20:/exlibris/app/oracle/product/11:Y
```

- The following environment variables point to the Oracle Database 11g Release 1 (11.1) directories:
 - ORACLE_HOME

- PATH

Use the command:

```
echo $ORACLE_HOME
echo $PATH
```

- The oracle10 DB is down:

```
ps -ef | grep smon<oracle9_SID>
```

7 At a system prompt, change to the `$ORACLE_HOME/rdbms/admin` directory:

```
cd $ORACLE_HOME/rdbms/admin
```

8 Start SQL*Plus and connect to the database instance as a user with SYSDBA privileges:

```
sqlplus '/as sysdba'
```

9 Start the instance:

```
SQL> set pause off
SQL> STARTUP UPGRADE
```

NOTE:

The UPGRADE keyword enables you to open a database based on an earlier Oracle database release. It also restricts logging on to AS SYSDBA sessions, disables system triggers, and performs additional operations that prepare the environment for the upgrade.

10 Set the system to spool results to a log file for later verification of success:

```
SQL> SPOOL upgrade.log
```

11 Run the `catupgrd.sql` script:

```
SQL> @catupgrd.sql
```

The `catupgrd.sql` script determines which upgrade scripts must be run, runs them, and then shuts down the database. You must run the script in the Oracle Database 11g Release 1 (11.1) environment.

12 Restart the instance to reinitialize the system parameters for normal operation.

```
SQL> STARTUP
```

This restart, following the database shutdown performed as part of the `catupgrd.sql` script, flushes all caches, clears the buffers, and performs other housekeeping activities. These measures are an important final step to ensure the integrity and consistency of the newly upgraded Oracle database software.

The Post-Upgrade Status Tool provides a summary of the upgrade at the end of the spool log. It displays the status of the database components in the upgraded database and the time required to complete each component upgrade. Any errors that occur during the upgrade are listed with each component and must be addressed.

- 13** Run `utlul11s.sql` to display the results of the upgrade:

```
SQL> @utlul11s.sql
```

The Post-Upgrade Status Tool displays output similar to the following:

Component	Status	Version
HH:MM:SS		
Oracle Server	VALID	11.1.0.7.0
00:11:48		
JServer JAVA Virtual Machine	VALID	11.1.0.7.0
00:04:09		
Oracle Workspace Manager	VALID	11.1.0.7.0
00:00:47		
Oracle XDK	VALID	11.1.0.7.0
00:00:19		
Oracle Text	VALID	11.1.0.7.0
00:00:50		
Oracle XML Database	VALID	11.1.0.7.0
00:01:57		
Oracle Database Java Packages	VALID	11.1.0.7.0
00:00:09		
Oracle Multimedia	VALID	11.1.0.7.0
00:10:47		
Oracle Expression Filter	VALID	11.1.0.7.0
00:00:09		
Oracle Rules Manager	VALID	11.1.0.7.0
00:00:09		
Gathering Statistics		
00:01:13		
Total Upgrade Time: 00:32:22		

If the Post-Upgrade Status Tool returns errors or shows components that are not VALID or not the most recent release, see [Appendix A: Troubleshooting the Upgrade](#) on page 37.

- 14 Run `catuppst.sql`, located in the `ORACLE_HOME/rdbms/admin` directory, to perform upgrade actions that do not require the database to be in UPGRADE mode:

```
SQL> @catuppst.sql
```

- 15 Run `utlrbp.sql` to recompile any remaining stored PL/SQL and Java code.

```
SQL> @utlrbp.sql
```

- 16 Verify that all expected packages and classes are valid:

```
select OBJECT_NAME,STATUS from dba_objects where
STATUS != 'VALID';
```

- 17 Exit SQL*Plus.

- 18 Start the listener. As the oracle11 user, enter the command:

```
lsnrctl start
```

Your database is now upgraded to the Oracle Database 11g Release 1 (11.1). Perform the procedures described in [Post-Upgrade Activities](#) on page 27.

CAUTION:

If you retain the old Oracle software, do not start the upgraded database with the old software. Only start the database with the executables in the new Oracle database installation.

3

Post-Upgrade Activities

This section includes:

- **Migrating the Initialization Parameter File to a Server Parameter File** on page 27
- **Setting Parameters to Ignore Case-Sensitivity** on page 28
- **Adjusting the Initialization Parameter File for the New Release** on page 28
- **Changing the Database `global_name`** on page 29
- **Configuring Enterprise Manager** on page 30
- **Installing Oracle JVM** on page 31
- **Installing Oracle XMLDB and the `DBMS_CUBE_EXP` Package** on page 34
- **Collecting Statistics for the Application Schemas and Objects** on page 36
- **Recreating Roles** on page 36

Migrating the Initialization Parameter File to a Server Parameter File

Create a server parameter file using the `CREATE SPFILE` statement. This statement reads the initialization parameter file to create a server parameter file. You are not required to start the database to issue a `CREATE SPFILE` statement.

The following is an example, as an `oracle11` user:

```
cd $ORACLE_HOME/dbs
Sqlplus '/as sysdba'
create spfile from pfile;
```

Start up the instance using the newly-created server parameter file.

```
sqlplus '/as sysdba'  
Startup force
```

Setting Parameters to Ignore Case-Sensitivity

Beginning in Oracle Database 11g Release 1 (11.1), you can enforce case-sensitivity for passwords. For example, the password hPP5620qr fails if it is entered as hpp5620QR or hPp5620Qr. In previous releases, passwords were not case-sensitive.

To disable case-sensitivity:

As an oracle11 user, enter the following commands:

```
sqlplus '/as sysdba'  
alter system set sec_case_sensitive_logon=false  
scope=both;
```

If you want the sys password not to be case sensitive, enter the following as an oracle 11 user:

```
mv orapw<sid> orapw<sid>.old  
ORACLE_HOME/bin/orapwd file=$ORACLE_HOME/dbs/  
orapw${ORACLE_SID} password=<the password> entries=10  
ignorecase=YES
```

For example:

```
mv orapwaleph19 orapwaleph19.old  
ORACLE_HOME/bin/orapwd file=$ORACLE_HOME/dbs/  
orapw${ORACLE_SID} password=manager entries=10  
ignorecase=YES
```

Adjusting the Initialization Parameter File for the New Release

Each release of the Oracle database introduces new initialization parameters, deprecates some initialization parameters, and makes some initialization parameters obsolete. You should adjust the parameter file to account for these

changes and to take advantage of new initialization parameters that might be beneficial to your system.

The COMPATIBLE initialization parameter controls the compatibility level of your database. When you are certain that you no longer need the ability to downgrade your database to its original version, set the COMPATIBLE initialization parameter based on the compatibility level you want for your new database (11.1.0.0.0):

```
sqlplus '/as sysdba'  
alter system set compatible = '11.1.0.0.0' scope=spfile;
```

Changing the Database global_name

If you changed the database name while performing the upgrade, you need to change the database `global_name` (which will in turn affect the prompt viewed while connecting to `sqlplus`).

To change the database `global_name`:

- 1 Query your current `global_name`:

```
Sqlplus '/as sysdba'  
SQL> SELECT * FROM GLOBAL_NAME;
```

- 2 Query the domain name:

```
SQL>select VALUE FROM V$PARAMETER WHERE NAME =  
'db_domain';
```

- 3 Change the database `global_name` by issuing the following command:

```
ALTER DATABASE RENAME GLOBAL_NAME TO  
<db_name>.<domain_name>;
```

For example:

```
ALTER DATABASE RENAME GLOBAL_NAME TO  
ALEPH20.EXLIBRIS.INT.IL;
```

Configuring Enterprise Manager

If you are not yet using Oracle Enterprise Manager to manage your database, install and configure Enterprise Manager Database Control.

To install and configure Enterprise Manager Database Control:

- 1 As an oracle11 user, enter the following command:

```
emca -config dbcontrol db -repos create
```

- 2 Answer the questions regarding SID and passwords.
- 3 If you need to drop an old repository enter the following command:

```
emca -deconfig dbcontrol db -repos drop)
```

- 4 If your database is already being managed by Oracle Enterprise Manager Database Control or Oracle Enterprise Manager Grid Control, update the configuration:

```
emca -upgrade (db | asm | db_asm) [-cluster] [-silent]  
[parameters]
```

You need to run this from the new Oracle Database 11g Oracle home directory.

- 5 When prompted, provide the Oracle home directory from which the configuration is being upgraded.

At the end of the installation, the URL for the database console is displayed.

To view the database console using Mozilla Firefox:

- 1 Open Firefox.
- 2 Select **Options > Advanced > Encryption > View Certificates > Server > Add Exception**.
- 3 In the blank frame, enter the database console URL.

Installing Oracle JVM

Check if the Oracle Java Virtual Machine is installed:

```
sqlplus '/as sysdba'
set linesize 150
col COMP_NAME format a50
select COMP_NAME,VERSION,STATUS from dba_registry where
COMP_NAME like '%Java%' or COMP_NAME like '%JAVA%';
```

If the Oracle Java machine is not installed, you must install it.

To install Oracle JVM:

- 1 Verify that the SYSTEM tablespace has at least 70 MB of free space.

The `initjvm.sql` script in 11g checks if the required resources are available when it runs. If they are not available, the execution of the script terminates and an error message is displayed indicating which resources need to be increased.

- 2 Enter the following as an oracle11 user:

```
cd rdbms/admin/
sqlplus /nolog
set pause off
spool full_jvminst.log;
set echo on
connect / as sysdba
startup mount
alter system set "_system_trig_enabled" = false
scope=memory;
alter database open;
@?/javavm/install/initjvm.sql
select count(*), object_type from all_objects where
object_type like '%JAVA%' group by object_type;
```

The following is a sample output:

```
COUNT(*)  OBJECT_TYPE
-----  -
298  JAVA DATA
685  JAVA RESOURCE
18661  JAVA CLASS

3 rows selected.
```

3 Enter the following:

```
@?/xdk/admin/initxml.sql
Select count(*), object_type from all_objects where
object_type like '%JAVA%' group by object_type;
```

The following is a sample output:

```
COUNT(*)  OBJECT_TYPE
-----  -
298  JAVA DATA
761  JAVA RESOURCE
19661  JAVA CLASS

3 rows selected.
```

4 Enter the following:

```
set echo on
@?/xdk/admin/xmlja.sql
Select count(*), object_type from all_objects where
object_type like '%JAVA%' group by object_type;
```


The following is a sample output:

```
COUNT(*)  OBJECT_TYPE
-----  -
298  JAVA DATA
761  JAVA RESOURCE
19661  JAVA CLASS

3 rows selected.
```

5 Enter the following:

```
set echo on
@?/rdbms/admin/catjava.sql
select count(*), object_type from all_objects where
object_type like '%JAVA%' group by object_type;
```

The following is a sample output:

```
COUNT(*)  OBJECT_TYPE
-----  -
298  JAVA DATA
764  JAVA RESOURCE
20017  JAVA CLASS

3 rows selected.
```

6 Enter the following:

```
Set echo on
@?/rdbms/admin/catexf.sql
Select count(*), object_type from all_objects where
object_type like '%JAVA%' group by object_type;
```

The following is a sample output:

```
COUNT(*)  OBJECT_TYPE
-----  -
      298  JAVA DATA
      765  JAVA RESOURCE
     20060  JAVA CLASS
3 rows selected.
```

7 Enter the following:

```
set echo on
shutdown immediate
set echo off
spool off
exit
```

The Oracle JVM is installed.

To start the database and check that the installation is valid:

```
sqlplus '/as sysdba'
startup
Set linesize 150
col COMP_NAME format a50
Select COMP_NAME,VERSION,STATUS from dba_registry where
COMP_NAME like '%Java%' or COMP_NAME like '%JAVA%';
```

Installing Oracle XMLDB and the DBMS_CUBE_EXP Package

Check if Oracle XMLDB is installed:

```
sqlplus '/as sysdba'
set linesize 150
col COMP_NAME format a50
select COMP_NAME,VERSION,STATUS from dba_registry where
COMP_NAME like '%Oracle XML Database';
```

If Oracle XMLDB is not installed, you must install it.

To install Oracle XMLDB

- 1 As an oracle11 user, enter the following:

```
cd rdbms/admin/  
sqlplus '/as sysdba'  
Set pause off
```

- 2 Run a script to create the tables and views required to run the XML database:

```
@catqm.sql <XDB_password> <XDB_TS_NAME>  
<TEMP_TS_NAME>
```

For example:

```
@catqm.sql change_on_install SYSAUX TEMP;
```

- 3 Add the following dispatcher entry to the `init.ora` file:

```
dispatchers="(PROTOCOL=TCP) (SERVICE=<sid>XDB)"
```

- 4 Restart the database and the listener to enable the Oracle XML database protocol access as oracle11 user:

```
lsnrctl stop  
lsnrctl start
```

- 5 Check that the installation is valid:

```
set linesize 150  
col COMP_NAME format a50  
select COMP_NAME,VERSION,STATUS from dba_registry  
where COMP_NAME like '%Oracle XML Database';
```

- 6 Add the `DBMS_CUBE_EXP` package:

```
SQL>@$ORACLE_HOME/olap/admin/olap.sql SYSAUX TEMP;
```

Collecting Statistics for the Application Schemas and Objects

NOTE:

Collecting Statistics is not applicable to the Aleph and MetaLib products, and should not be applied.

For all other products, this section will be updated once collecting statistics is tested for them.

Recreating Roles

As the system user, run the following script to recreate roles in order to avoid the secure application role configuration of version 11.1.0.7.

To recreate roles:

As an oracle11 user, enter the following commands:

```
cd $ORACLE_HOME/bin
csh -f fix_roles_def.csh
```

A

Troubleshooting the Upgrade

This section explains what to do if something goes wrong with the upgrade.

This section includes:

- [Resource Limits](#) on page 37
- [Component Status](#) on page 38
- [Rerunning the Upgrade](#) on page 38

Resource Limits

If you run out of resources during the upgrade, do the following:

- Increase the resource allocation.
- Perform a SHUTDOWN ABORT.
- Restart the instance (in upgrade mode).
- Rerun the `catupgrd.sql` script or restart DBUA.

There are three resources that generally require increases for a new Oracle database release:

- SYSTEM tablespace

Typically you receive one of the following messages during the upgrade if your SYSTEM tablespace size is insufficient:

- ORA-01650 – Unable to extend rollback segment string by string in tablespace string
- ORA-01651 – Unable to extend save undo segment by string for tablespace string
- ORA-01652 – Unable to extend temp segment by string in tablespace string
- ORA-01653 – Unable to extend table string.string by string in tablespace string

- ORA-01654 – Unable to extend index string.string by string in tablespace string
- ORA-01655 – Unable to extend cluster string.string by string in tablespace string
- SYSAUX tablespace
- UNDO tablespace – Be sure it is at least 400 MB.

Component Status

The Post-Upgrade Status Tool should report a **VALID** status for all components at the end of the upgrade. The following list shows and briefly describes other status values that you might see:

- **NO SCRIPT** – The component upgrade script was not found in `ORACLE_HOME`. Check the install logs, install the component software, and then rerun `catupgrd.sql`.
- **OPTION OFF** – The server option required for the component was not installed or was not linked with the server. Check the `V$OPTION` view as well as the install logs. Install the component or relink the server with the required option, and then rerun `catupgrd.sql`.
- **REMOVED** – The component was not upgraded because it was removed from the database.
- **INVALID** – Some objects for the component were invalid at the completion of the upgrade. If there were no errors during the component upgrade, then running `utlrp.sql` might change the status to **VALID** without rerunning the entire upgrade. Check the `DBA_REGISTRY` view after running `utlrp.sql`.
- **UPGRADING** – The component upgrade did not complete. Resolve the problem and rerun `catupgrd.sql`.

Rerunning the Upgrade

To rerun the upgrade:

- 1 Shut down the database:

```
SQL> SHUTDOWN IMMEDIATE
```

- 2 Restart the database in **UPGRADE** mode:

```
SQL> STARTUP UPGRADE
```

3 Rerun `catupgrd.sql`:

```
SQL> @catupgrd.sql
```

4 Rerun `utlul11s.sql`:

```
SQL> @utlul11s.sql
```


B

Configuration Files

After analyzing the database to be upgraded, you should complete the following steps to prepare the new Oracle home directory.

To create the new Oracle home directory:

- 1 Copy the configuration files from the Oracle home directory of the database being upgraded to the Oracle Database 11g Release 1 (11.1) Oracle home directory. By default, Oracle looks for the parameter file in the `ORACLE_HOME/dbs`. The parameter file can reside anywhere, but it should not reside in the old environment's Oracle home directory after you upgrade to Oracle Database 11g Release 1 (11.1).
- 2 Perform one of the following procedures:
 - If you have a server parameter file (SPFILE), you need to create a text file (PFILE):
 - a Log on as an oracle10 user:

```
cd dbs
```

- b Check whether an spfile exists:

```
ls -l spfile*<ORACLE_SID>*
```

- c Create the PFILE:

```
sqlplus '/as sysdba'  
SQL> create pfile from spfile;
```

- d Move the file:

```
mv init<ORACLE_SID>.ora <oracle11 $ORACLE_HOME>/dbs
```

For example:

```
mv initaleph0.ora /exlibris/app/oracle/product/11/
dbs
```

- If you do not have a server parameter file (SPFILE), log on as an oracle10 user:

```
cp init<ORACLE_SID>.ora <oracle11 $ORACLE_HOME>/dbs
```

For example:

```
cp initaleph0.ora /exlibris/app/oracle/product/11/
dbs
```

- 3 If you have a password file that resides within the old environment's Oracle home directory, move or copy the password file to the Oracle Database 11g Release 1 (11.1) Oracle home directory.

Log on as an oracle10 user:

```
cd dbs
cp orapw<sid> <oracle11 $ORACLE_HOME>
```

For example:

```
cp orapwaleph0 /exlibris/app/oracle/product/11/dbs
```

- 4 Adjust your parameter file in Oracle Database 11g Release 1 (11.1). As an oracle11 user, edit the file `init<ORACLE_SID>.ora`:
 - a Remove obsolete initialization parameters and adjust deprecated initialization parameters. Certain parameters are obsolete in Oracle Database 11g Release 1 (11.1), while other parameters have become deprecated. Remove all obsolete parameters from any parameter file that starts an Oracle Database 11g Release 1 (11.1) instance. Obsolete parameters might cause errors in Oracle Database 11g Release 1 (11.1). Also, alter any parameter whose syntax has changed in the new release. The Pre-Upgrade Information Tool displays any deprecated parameters and obsolete parameters it finds in the `Deprecated Parameters` and `Obsolete Parameters` sections, respectively.
 - b Remove the parameter `cursor_space_for_time`, if it exists.

- c** Replace the parameters `background_dump_dest`, `user_dump_dest`, `core_dump_dest`, and `udit_file_dest` with the parameter `diagnostic_dest` pointing to `$ORACLE_BASE`. For example:

```
diagnostic_dest=/exlibris/app/oracle
```

- d** Replace the parameter `log_archive_dest` with the parameter `log_archive_dest_1`. Use the following syntax:

```
LOG_ARCHIVE_DEST_1 = 'LOCATION=<archive location>'
```

For example:

```
log_archive_dest_1='LOCATION=/exlibris/oradata/aleph20/arch'
```

- e** Replace the parameter `sga_target` with the parameter `memory_target`. Set the value to the same number, 160 MB, at least.
- f** For Aleph only, add the following parameters:
- `OPTIMIZER_DYNAMIC_SAMPLING=0`
 - `OPTIMIZER_MODE=CHOOSE`
 - `SQLTUNE_CATEGORY =limited`
- g** Make sure the `COMPATIBLE` initialization parameter is properly set for Oracle Database 11g Release 1. The Pre-Upgrade Information Tool displays a warning in the Database section if `COMPATIBLE` is not properly set and specifies the correct value.
- h** Adjust the values of the initialization parameters to at least the minimum value indicated by the Pre-Upgrade Information Tool.
- i** Make sure all path names in the parameter file are fully specified. You should not have relative path names in the parameter file.
- 5** Save all of the files you modified after making these adjustments.

