Oracle® Database

Upgrade Guide

11*g* Release 2 (11.2)

E23633-07

September 2012

Oracle Database Upgrade Guide, 11*g* Release 2 (11.2)

E23633-07

Primary Author: Cathy Shea

Contributors: Toshinori Ami, Vickie Carbonneau, Ian Dai, Mike Dietrich, Joseph Errede, Ayako Ito, Masakazu Ito, Cindy Lim, Brian McCarthy, Tony Morales, Carol Palmer, Satish Panchumarthy, Ravi Pattabhi, Kathy Rich, Mark Richwine, Craig Santelman, Viv Schupmann, Janet Stern, Carol Tagliaferri, Venkateshwaran Venkataramani, Douglas Williams

Contents

2 Preparing to Upgrade Oracle Database

3 Upgrading to the New Release of Oracle Database

5 Upgrading Applications After Upgrading Oracle Database

6 Downgrading Oracle Database to an Earlier Release

7 Moving Data Using Oracle Data Pump

A Behavior Changes After Upgrading Oracle Database

B Gathering Optimizer Statistics for Upgrading Oracle Database

Index

Preface

This manual guides you through the process of planning and executing Oracle Database upgrades. In addition, this manual provides information about compatibility, upgrading applications, and important changes in the new Oracle Database release, such as initialization parameter changes and data dictionary changes.

Oracle Database Upgrade Guide contains information that describes the features and functions of Oracle Database (also known as the standard edition) and Oracle Database Enterprise Edition products. Oracle Database and Oracle Database Enterprise Edition have the same basic features. However, several advanced features are available only with the Enterprise Edition, and some of these are optional. For example, to use application failover, you must have the Enterprise Edition with the Oracle Real Application Clusters option.

> **See Also:** *Oracle Database New Features Guide* for information about the differences between Oracle Database and Oracle Database Enterprise Edition and the features and options that are available to you.

This preface contains these topics:

- Audience
- Documentation Accessibility
- Related Documentation
- Conventions

Audience

Oracle Database Upgrade Guide is intended for database administrators (DBAs), application developers, security administrators, system operators, and anyone who plans or executes Oracle Database upgrades.

To use this document, you must be familiar with the following:

- Relational database concepts
- Your current release of Oracle Database
- Your operating system environment

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documentation

For more information, see these Oracle resources:

- "Oracle Upgrade Companion" Note ID 785351.1 on My Oracle Support at `https://support.oracle.com`.
- *Oracle Database Concepts* for a comprehensive introduction to the concepts and terminology used in this manual
- *Oracle Database Administrator's Guide* for information about administering Oracle Database
- *Oracle Database SQL Language Reference* for information on Oracle Database SQL commands and functions
- *Oracle Database Utilities* for information about utilities bundled with Oracle Database, including Data Pump, Export, Import, and SQL*Loader
- *Oracle Database Net Services Administrator's Guide* for information about Oracle Net Services
- *Oracle Database Enterprise User Security Administrator's Guide* for information about Oracle Label Security
- *Oracle Database High Availability Best Practices* for Oracle operational best practices and help choosing an upgrade method that minimizes downtime.

Many of the examples in this guide use the sample schemas, which are installed by default when you select the Basic Installation option with an Oracle Database installation. Refer to *Oracle Database Sample Schemas* for information on how these schemas were created and how you can use them yourself.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Introduction to the Upgrade Process for Oracle Database

Oracle provides upgrade options that are tailored to your database environment and tools that automate the process for upgrading Oracle Database.

This chapter contains the following topics:

Where to Find the Latest Information About Upgrading Oracle Database

Oracle provides late-breaking updates, discussions, and best practices about pre-upgrade, post-upgrade, compatibility, and interoperability on the My Oracle Support Web site at `https://support.oracle.com`. You can search on note IDs or key words like "Database Upgrade."

- For the complete knowledge base and latest information about patch sets, go to My Oracle Support at `https://support.oracle.com` and enter your search criteria.
- For information about upgrading to Oracle Database 11*g* Release 2, see *Oracle Upgrade Companion* ID 785351.1 at My Oracle Support at `https://support.oracle.com`. Other upgrade companions are also available for upgrading to earlier releases.
- For information about downloading and running the pre-upgrade information tool, see *Oracle Database Pre-Upgrade Utility* ID 884522.1 at My Oracle Support at `https://support.oracle.com`.

Overview of Oracle Database Upgrade Tools and Processes

The upgrade procedure transforms an existing Oracle Database system (including associated applications) into an Oracle Database 11*g* Release 2 (11.2) system. The upgrade steps apply to all operating systems, with the possible exception of a few operating system-specific details identified in your operating system-specific Oracle documentation.

Oracle Database 11*g* Release 2 (11.2) supports the following tools and methods for upgrading a database to the new release:

- Database Upgrade Assistant (DBUA)

  Provides a graphical user interface that guides you through the upgrade of a database. DBUA can be launched during installation with the Oracle Universal Installer, or you can launch DBUA as a standalone tool at any time in the future.

  > **Note:** DBUA is the recommended method for performing a major release upgrade or patch release upgrade.

- Manual upgrade using SQL scripts and utilities

  Enables upgrades to be performed at the command line with SQL scripts and utilities.

- Export and Import utilities

  Uses Oracle Data Pump utilities, or the native Export and Import features within Oracle Database. These utilities perform a full or partial export from your database, followed by a full or partial import into the new Oracle Database 11*g*. Export/Import can copy a subset of the data, leaving the database unchanged.

- `CREATE TABLE AS` SQL statement

  Copies data from a database into a new Oracle Database 11*g* database. Data copying can copy a subset of the data, leaving the database unchanged.

  See Also: "Choose an Upgrade Method for Upgrading Oracle Database" on page 2-3 for more details about the upgrade tools and methods

Major Steps in the Upgrade Process for Oracle Database

Detailed instructions for upgrading an existing Oracle database to the new Oracle Database 11*g* release are provided in the appropriate chapters in *Oracle Database Upgrade Guide*.

The following steps outline the major procedures performed during the upgrade process:

- Step 1: Prepare to Upgrade
- Step 2: Test the Upgrade Process
- Step 3: Test the Upgraded Test Database
- Step 4: Prepare and Preserve the Production Database
- Step 5: Upgrade the Production Database
- Step 6: Tune and Adjust the New Production Database

Step 1: Prepare to Upgrade

- Become familiar with the features of the new Oracle Database 11*g* release.
- Determine the upgrade path to the new release.
- Choose an upgrade method.
- Choose an Oracle home directory for the new release.
- Develop a testing plan.
- Prepare a backup strategy.

> **Note:** During the upgrade, consider running multiple releases of the database software so you can use the existing release as your production environment while you test the new release. See "About Running Multiple Oracle Releases" on page 5.

Step 2: Test the Upgrade Process

- Perform a test upgrade using a test database. The test upgrade should be conducted in an environment created for testing and should not interfere with the actual production database.

Step 3: Test the Upgraded Test Database

- Perform the tests you planned in Step 1 on the test database and on the test database that was upgraded to the new Oracle Database 11*g* release.
- Compare results, noting anomalies between test results on the test database and on the upgraded database.
- Investigate ways to correct any anomalies you find and then implement the corrections.
- Repeat Step 1, Step 2, and the first parts of Step 3, as necessary, until the test upgrade is completely successful and works with any required applications.

Chapter 2, "Preparing to Upgrade Oracle Database" provides detailed information about Steps 1 through 3.

Step 4: Prepare and Preserve the Production Database

- Prepare the current production database as appropriate to ensure the upgrade to the new Oracle Database 11*g* release is successful.
- Schedule the downtime required for backing up and upgrading the production database.
- Back up the current production database. Perform a full or an incremental backup, as necessary, to ensure your database is protected against data loss.

Step 5: Upgrade the Production Database

- Upgrade the production database to the new Oracle Database 11*g* release.
- After the upgrade, perform a full backup of the production database and perform other post-upgrade tasks.

Chapter 3, "Upgrading to the New Release of Oracle Database" describes Steps 4 and 5 when using DBUA or when performing a manual upgrade. Chapter 4, "Post-Upgrade Tasks for Oracle Database" describes the backup procedure for Step 5 after the upgrade, and other post-upgrade tasks for Step 6.

Step 6: Tune and Adjust the New Production Database

- Tune the new Oracle Database 11*g* production database. The new production database should perform to the same standards, or better than, the database before the upgrade.

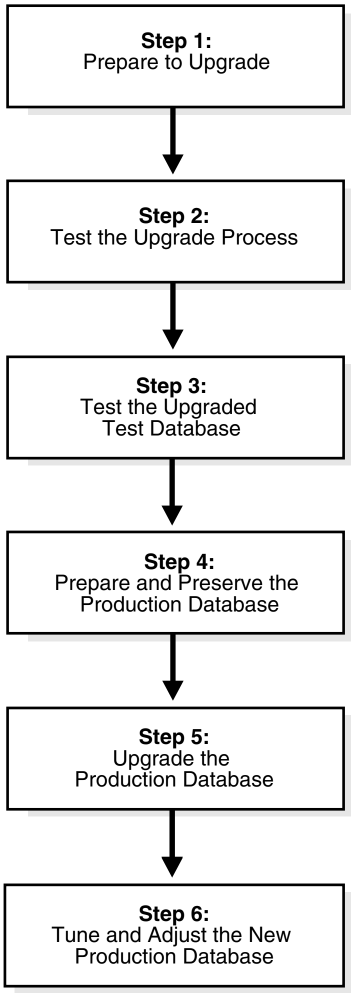
  Chapter 4, "Post-Upgrade Tasks for Oracle Database" describes these adjustments.

- Determine which features of the new Oracle Database 11*g* release you want to use, and update your applications accordingly.
- Develop new database administration procedures as needed.
- Do not upgrade production users to the new release until all applications have been tested and operate properly.

  Chapter 5, "Upgrading Applications After Upgrading Oracle Database" describes considerations for updating applications.

Figure 1–1 illustrates the major steps in the upgrade process.

Figure 1–1 Upgrade Steps

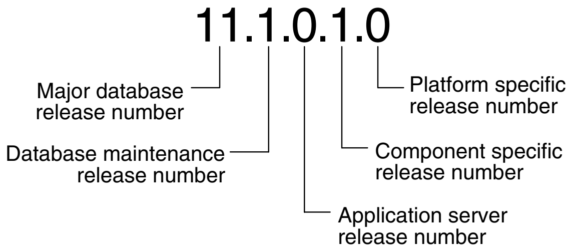

Step 1:
Prepare to Upgrade
Step 2:
Test the Upgrade Process
Step 3:
Test the Upgraded Test Database
Step 4:
Prepare and Preserve the Production Database
Step 5:
Upgrade the Production Database
Step 6:
Tune and Adjust the New Production Database

About Oracle Database Release Numbers

Oracle Database Upgrade Guide describes moving between different releases of Oracle Database. Starting with Oracle9*i* Release 2 (9.2), maintenance releases of Oracle Database are denoted by a change to the second digit of a release number. In earlier releases, the third digit indicated a particular maintenance release.

Figure 1–2 illustrates each part of a release number and what it represents.

Figure 1–2 Example of an Oracle Release Number


11.1.0.1.0
Major database release number
Database maintenance release number
Application server release number
Component specific release number
Platform specific release number

More information about release numbers are contained in these topics:

- Convention for Referring to Release Numbers in Oracle Database Upgrade Guide
- About Running Multiple Oracle Releases

> **See Also:** *Oracle Database Administrator's Guide* for more information about Oracle release numbers

Convention for Referring to Release Numbers in *Oracle Database Upgrade Guide*

When a statement is made in *Oracle Database Upgrade Guide* about a major database release number, the statement applies to all releases within that major database release. References to Oracle Database 11*g* include Oracle Database 11*g* Release 1 (11.1) and Oracle Database 11*g* Release 2 (11.2). References to Oracle Database 10*g* include Oracle Database 10*g* Release 1 (10.1) and Oracle Database 10*g* Release 2 (10.2), and references to Oracle9*i* include Oracle9*i* Release 1 (9.0.1) and Oracle9*i* Release 2 (9.2) and so forth.

Similarly, when a statement is made in *Oracle Database Upgrade Guide* about a maintenance release, the statement applies to all component-specific and platform-specific releases (also referred to as patchset releases) within that maintenance release. Therefore, a statement about Oracle Database 11*g* Release 2 (11.2) applies to releases 11.2.0.1, 11.2.0.2, and 11.2.0.3, and all other platform-specific releases within Oracle Database 11*g* Release 2 (11.2).

About Running Multiple Oracle Releases

You can run different releases of Oracle Database on the same computer at the same time. However, you must observe the following conditions when simultaneously using multiple releases:

- An Oracle Database release must be installed in a new Oracle home that is separate from earlier releases of Oracle.

  There cannot be multiple releases for each Oracle home. Oracle recommends that you adopt an Optimal Flexible Architecture (OFA) when creating multiple Oracle homes.

> **See Also:** "Optimal Flexible Architecture (OFA) in Oracle Database" on page 10

- Each database server can access only a database that is consistent with its release number.

  For example, if you have Oracle9*i* and Oracle Database 11*g* installed on the same computer, then the Oracle9*i* database server can access Oracle9*i* databases but not Oracle Database 11*g* databases, and the Oracle Database 11*g* database server can access Oracle Database 11*g* databases but not Oracle9*i* databases.

  > **See Also:** Your operating system-specific Oracle documentation for more information about running multiple releases of Oracle Database on your operating system and any restrictions that may apply

The following topics provide general information about running multiple releases of Oracle Database:

- Databases in Multiple Oracle Homes on the Same Computer
- Databases in Multiple Oracle Homes on Separate Computers
- About Earlier Oracle Database Releases and Upgrading to the Current Release
- About Upgrading Clients to the Current Release of Oracle Database

Databases in Multiple Oracle Homes on the Same Computer

You may not be able to install earlier releases of Oracle Database on the same computer system and have clients connect to these databases. For example, you cannot have Oracle8*i*, Oracle9*i*, Oracle Database 10*g*, and Oracle Database 11*g* databases in multiple (or separate) Oracle homes on the same computer and have Oracle8*i*, Oracle9*i*, Oracle Database 10*g*, and Oracle Database 11*g* clients connecting to any or all of the databases. You may be able to have a combination of some releases on one system. However, be sure to obtain the latest information on compatibility and supported configurations.

> **See Also:** Note 207303.1 "Client / Server / Interoperability Support Between Different Oracle Versions" on My Oracle Support at `https://support.oracle.com`

Databases in Multiple Oracle Homes on Separate Computers

You can install Oracle8*i*, Oracle9*i*, Oracle Database 10*g*, and Oracle Database 11*g* databases in multiple (separate) Oracle homes on separate computers and have Oracle8*i*, Oracle9*i*, Oracle Database 10*g*, and Oracle Database 11*g* clients connecting to any or all of the databases.

About Earlier Oracle Database Releases and Upgrading to the Current Release

You can upgrade an Oracle8*i*, Oracle9*i*, Oracle Database 10*g*, or Oracle Database 11*g* database to the new Oracle Database 11*g* release and have Oracle8*i*, Oracle9*i*, Oracle Database 10*g*, and Oracle Database 11*g* clients connecting to the upgraded database. Be sure to obtain the latest information on compatibility and supported configurations.

About Upgrading Clients to the Current Release of Oracle Database

You can upgrade any or all of your Oracle8*i*, Oracle9*i*, Oracle Database 10*g*, or Oracle Database 11*g* clients to the new Oracle Database 11*g* release. The new Oracle Database

11*g* release client can access your Oracle8*i*, Oracle9*i*, Oracle Database 10*g*, and Oracle Database 11*g* databases.

Compatibility and Interoperability Between Oracle Database Releases

Compatibility and interoperability issues may arise because of differences between Oracle Database releases. These differences might affect general database administration and existing applications.

The following topics discuss compatibility and interoperability:

- What Is Compatibility for Oracle Database?
- What Is Interoperability for Oracle Database?

> **See Also:** Appendix A, "Behavior Changes After Upgrading Oracle Database" for more information on compatibility and interoperability for specific releases

What Is Compatibility for Oracle Database?

Databases from different releases of Oracle Database software are compatible if they support the same features and those features perform the same way. When you upgrade to a new release of Oracle Database, certain new features might make your database incompatible with your earlier release.

Your upgraded database becomes incompatible with your earlier release under the following conditions:

- A new feature stores any data on disk (including data dictionary changes) that cannot be processed with your earlier release.
- An existing feature behaves differently in the new environment as compared to the old environment.

The following topics contain more information about compatibility:

- The COMPATIBLE Initialization Parameter in Oracle Database
- Default, Minimum, and Maximum Values of the COMPATIBLE Initialization Parameter for Oracle
- Considerations for Downgrading Oracle Database and Compatibility
- How the COMPATIBLE Initialization Parameter Operates in Oracle Database
- Checking the Compatibility Level of Oracle Database
- When to Set the COMPATIBLE Initialization Parameter in Oracle Database

The COMPATIBLE Initialization Parameter in Oracle Database

Oracle Database enables you to control the compatibility of your database with the `COMPATIBLE` initialization parameter. By default, when the `COMPATIBLE` initialization parameter is not set in your parameter file, it defaults to `11.2.0` for Oracle Database 11*g* Release 2 (11.2). You cannot use new Oracle Database 11*g* Release 2 (11.2) features that would make your upgraded database incompatible unless the `COMPATIBLE` initialization parameter is set to this value.

Note:

- Before upgrading to Oracle Database 11*g* Release 2 (11.2), the `COMPATIBLE` initialization parameter must be set to at least `10.0.0`, which is the lowest possible setting for Oracle Database 11*g* Release 2 (11.2).
- Oracle recommends increasing the `COMPATIBLE` parameter **only after thorough testing of the upgraded database** has been performed.

See Also: *Oracle Database Administrator's Guide* for information about managing initialization parameters

Default, Minimum, and Maximum Values of the COMPATIBLE Initialization Parameter for Oracle

Table 1–1 lists the default, minimum, and maximum values of the `COMPATIBLE` initialization parameter in Oracle Database 11*g* Release 2 (11.2) and in each release supported for upgrading to Oracle Database 11*g* Release 2 (11.2).

Table 1–1 The COMPATIBLE Initialization Parameter

Oracle Database Release	Default Value	Minimum Value	Maximum Value
Oracle9*i* Release 2 (9.2)	`8.1.0`	`8.1.0.0.0`	`9.2.0.`*`n.n`*
Oracle Database 10*g* Release 1 (10.1)	`10.0.0`	`9.2.0.0.0`	`10.1.0.`*`n.n`*
Oracle Database 10*g* Release 2 (10.2)	`10.2.0`	`9.2.0.0.0`	`10.2.0.`*`n.n`*
Oracle Database 11*g* Release 1 (11.1)	`11.0.0`	`10.0.0.0.0`	`11.1.0.`*`n.n`*
Oracle Database 11*g* Release 2 (11.2)	`11.2.0`	`10.0.0.0.0`	`11.2.0.`*`n.n`*

Considerations for Downgrading Oracle Database and Compatibility

Before upgrading to Oracle Database 11*g* Release 2 (11.2), the `COMPATIBLE` initialization parameter must be set to at least `10.0.0`. Only a subset of Oracle Database 11*g* Release 2 (11.2) features are available while the `COMPATIBLE` initialization parameter is set to this value.

After upgrading to Oracle Database 11*g* Release 2 (11.2), you can set the `COMPATIBLE` initialization parameter to match the release number of the new release. Doing so enables you to use all features of the new release, but prevents you from downgrading to your earlier release.

If, after upgrading, you want to downgrade, then the `COMPATIBLE` initialization parameter must be set as follows after the upgrade:

- Keep the setting at `10.1.0` if you upgraded from Oracle Database 10*g* Release 1 (10.1)
- Set to `10.2.0` or 10.1.0 if you upgraded from Oracle Database 10*g* Release 2 (10.2)
- Set to `11.1.0`, `10.2.0`, or `10.1.0` if you upgraded from Oracle Database 11*g* Release 1 (11.1)

Important: After you increase the COMPATIBLE parameter, the database cannot subsequently be downgraded to releases earlier than what is set for compatibility.

See Also: Chapter 6, "Downgrading Oracle Database to an Earlier Release" for more information about downgrading

How the COMPATIBLE Initialization Parameter Operates in Oracle Database

The COMPATIBLE initialization parameter operates in the following way:

- It controls the behavior of your database. For example, if you run an Oracle Database 11*g* Release 2 (11.2) database with the COMPATIBLE initialization parameter set to 10.1.0, then it generates database structures on disk that are compatible with Oracle Database 10*g* Release 1 (10.1). Therefore, the COMPATIBLE initialization parameter enables or disables the use of features. If you try to use any new features that make the database incompatible with the COMPATIBLE initialization parameter, then an error is displayed. However, any new features that do not make incompatible changes on disk are enabled.
- It makes sure that the database is compatible with its setting. If the database becomes incompatible with its setting, then the database does not start and terminates with an error. If this happens, then you must set the COMPATIBLE initialization parameter to an appropriate value for the database.

See Also: *Oracle Database Concepts* for more information about database structures

Checking the Compatibility Level of Oracle Database

The compatibility level of your database corresponds to the value of the COMPATIBLE initialization parameter. For example, if you set the COMPATIBLE initialization parameter to 11.2.0, then the database runs at release 11.2.0 compatibility level.

To check the current value of the COMPATIBLE initialization parameter

- Enter the following SQL statement:

```
SQL> SELECT name, value FROM v$parameter
         WHERE name = 'compatible';
```

When to Set the COMPATIBLE Initialization Parameter in Oracle Database

Oracle recommends increasing the COMPATIBLE parameter only after complete testing of the upgraded database has been performed. After the upgrade is complete, you can increase the setting of the COMPATIBLE initialization parameter to the maximum level for the new Oracle Database 11*g* release. However, after you increase the COMPATIBLE parameter, the database cannot subsequently be downgraded.

What Is Interoperability for Oracle Database?

In Oracle Database, *interoperability* is the ability of different releases of Oracle Database to communicate and work in a distributed environment. A distributed database system can comprise different releases of Oracle Database, and all supported releases of Oracle Database can participate in the distributed database system. However, the applications that work with a distributed database must also be able to interoperate with the features and functions that are available at each node in the system.

Interoperability across disperate operating systems and operating system versions might be a problem (especially during rolling upgrades) because the minimum requirements for the new Oracle Database 11*g* release might require you to upgrade the operating systems on some or all of your hosts. Therefore, you must check for compatibilities between drivers, network, and storage for all the interim states of the system during the rolling upgrade.

Note: Because *Oracle Database Upgrade Guide* discusses upgrading and downgrading between different releases of Oracle Database, the definition of *interoperability* is for Oracle Database releases. Other Oracle documentation might use a broader definition of the term *interoperability*. For example interoperability might in some cases describe communication between different hardware platforms and operating systems.

See Also: Note ID 207303.1 "Client / Server / Interoperability Support Between Different Oracle Versions" on My Oracle Support at `https://support.oracle.com`

Optimal Flexible Architecture (OFA) in Oracle Database

Oracle recommends the Optimal Flexible Architecture (OFA) standard for Oracle Database installations. The OFA standard is a set of configuration guidelines for efficient and reliable Oracle databases, which results in more streamlined maintenance.

OFA provides the following benefits:

- Organizes large amounts of complicated software and data on disk to avoid device bottlenecks and poor performance
- Facilitates routine administrative tasks, such as software and data backup functions, which are often vulnerable to data corruption
- Alleviates switching among multiple Oracle databases
- Adequately manages and administers database growth
- Helps to eliminate fragmentation of free space in the data dictionary, isolates other fragmentation, and minimizes resource contention

If you are not currently using the OFA standard, then switching to the OFA standard involves modifying your directory structure and relocating your database files.

See Also:

- Your operating system-specific Oracle documentation for more information about OFA
- *Oracle Database Administrator's Guide* for information about modifying your directory structure and relocating database files

Converting Databases to 64-bit Oracle Database Software

If you are installing 64-bit Oracle Database 11*g* software but were previously using a 32-bit Oracle Database installation, the database is automatically converted to 64-bit during a patch release or major release upgrade to the new Oracle Database 11*g* release.

To complete the conversion, you must manually perform the following post-upgrade tasks:

- Increase initialization parameters affecting the system global area, such as `SGA_TARGET` and `SHARED_POOL_SIZE`, to support 64-bit operations.
- Reconnect after starting up the database so that a new session is created with the appropriate 64-bit initialization. For example:

```
CONNECT  AS SYSDBA
STARTUP UPGRADE
CONNECT  AS SYSDBA
SPOOL UPGRADE.LOG
@catupgrd.sql
```

See Also:

- *Oracle Database Installation Guide* for your platform-specific information about 64-bit software installations
- Note 341880.1 "How to convert a 32-bit database to a 64-bit database on Linux" on My Oracle Support at `http://support.oracle.com`

Troubleshooting 32-bit to 64-bit Conversion for Oracle Database

When migrating a 32-bit database to 64-bit, when running the `utlrp.sql` script, there is a known error, ORA-07445.

ORA-07445: exception encountered: core dump `[_intel_new_memcpy()+2132] [ACCESS_VIOLATION] [ADDR:0x11D1F0F67] [PC:0x621BEA4] [UNABLE_TO_READ] []`

The `utlrp.sql` script, which is located in the *`ORACLE_HOME`*`/rdbms/admin` directory, is used to recompile stored PL/SQL and Java code. This error and workaround are described in Note 341880.1 "How to convert a 32-bit database to a 64-bit database on Linux" on My Oracle Support at `http://support.oracle.com`.

About Rolling Upgrades for Oracle Database

The rolling upgrade process supports Oracle Automatic Storage Management (Oracle ASM) instances in an Oracle Real Application Clusters (Oracle RAC) environment one at a time, without stopping the database. Depending on the method you choose to perform a rolling upgrade, you can upgrade the Oracle Database software or Oracle ASM instances, apply patchsets, or apply individual patches (sometimes referred to as one-off patches), incurring little or no database downtime. Database instance rolling upgrade is not supported.

See Also:

- *Oracle Database High Availability Best Practices* for help choosing a method to perform database upgrades
- "Summary of Methods for Performing Rolling Upgrades for Oracle Database" on page 11

Summary of Methods for Performing Rolling Upgrades for Oracle Database

Oracle provides different methods and options for performing rolling upgrades depending on your environment and applications. Table 1–2 summarizes the various methods for performing rolling upgrades and provides cross-references to the appropriate documentation.

Table 1–2 Methods for Performing Rolling Upgrades for Oracle Database

Method	Description	Reference
Oracle Data Guard SQL Apply	Use SQL Apply and logical standby databases to upgrade Oracle Database software and patchsets.	"About Upgrading Using Standby Databases" on page 3-6
Oracle Data Guard Physical Standby Database	Use an existing physical standby database to perform a rolling database upgrade by temporarily converting it to a logical standby database.	"About Upgrading Using Standby Databases" on page 3-6
Oracle Streams	By using Oracle Streams source and destination databases, you can upgrade to a new release of Oracle Database software, migrate an Oracle database to a different operating system and/or character set, upgrade user-created applications, and apply Oracle Database patches.	*Oracle Streams Concepts and Administration*
Oracle RAC with the OPatch Utility	Use the OPatch Utility to perform rolling patch upgrades with Oracle RAC. You can use the OPatch utility only to apply individual patches, not patchset releases. This method enables some instances of the Oracle RAC to remain available during the patch upgrade. Only the Oracle RAC instance being patched must be brought down; the other instances can continue to remain available. The OPatch Utility enables you to apply the patch successively to the different instances of Oracle RAC.	*Oracle Universal Installer and OPatch User's Guide for Windows and UNIX* *Oracle Database High Availability Best Practices*
Oracle Clusterware and Oracle Universal Installer (OUI)	Use OUI and Oracle Clusterware to perform a rolling upgrade to apply patchset releases of Oracle Clusterware. This method enables some instances of the Oracle RAC to remain available during the patchset upgrades. Only the node that is currently being patched must be brought down; the other instances remain available. OUI enables you to apply the patchset successively to the different instances of the cluster.	*Oracle Grid Infrastructure Installation Guide* *Oracle Universal Installer and OPatch User's Guide for Windows and UNIX*
Oracle Automatic Storage Management (Oracle ASM)	Use Oracle ASM to independently upgrade or patch clustered Oracle ASM instances. This method allows all of the features of a clustered Oracle ASM environment to continue to function even while one or more Oracle ASM instances run different software versions. **Note:** An Oracle ASM rolling upgrade to Oracle Database 11*g* Release 2 (11.2) moves the Oracle ASM instance to an Oracle Grid Infrastructure home.	*Oracle Grid Infrastructure Installation Guide* "Using Oracle ASM Rolling Upgrade" in *Oracle Database Storage Administrator's Guide*

Moving From Standard Edition to Enterprise Edition of Oracle Database

If you have Oracle Database Standard Edition at a release earlier than the new Oracle Database 11*g* release, then you can change it to Oracle Database Enterprise Edition by installing Oracle Enterprise Edition software and subsequently following the normal upgrade procedures, as described in *Oracle Database Upgrade Guide*.

Caution: Performing this procedure deinstalls the Standard Edition software and results in deleting database files that exist under the Oracle home. Therefore, you must back up any database files under the current Oracle home that you need to keep. This is explained in steps 4 and 5.

To change your existing Standard Edition database to an Enterprise Edition database

1. Ensure that the release number of your Standard Edition server software is the same release as Enterprise Edition server software.

   For example, if your Standard Edition server software is release 11.1.0.6, then you must upgrade to release 11.1.0.6 of Enterprise Edition.

2. Shut down your database.

3. If your operating system is Windows, then stop all Oracle services, including the `OracleService`*`SID`* Oracle service, where *`SID`* is the instance name.

4. Make a backup copy of files under `$Oracle_Home/network/admin`. This ensures that the listener is configured after you complete the steps in this procedure.

5. Back up all database files under the current Oracle home that you need to keep.

   - On Linux, back up `$Oracle_Home/dbs/`.
   - On Windows, back up `$Oracle_Home/database/`.

6. Edit the `orabase_cleanup.lst` file found in the `Oracle_Home/utl` directory and remove the "oradata" and "admin" entries. Refer to "Known Issue with the Deinstallation Tool for This Release" on page 13.

7. Deinstall Standard Edition server software. (**This step deletes all existing database files that reside under Oracle home.**)

   Run the `deinstall` tool from the Oracle home.

8. Install Enterprise Edition server software using Oracle Universal Installer (OUI).

   Select the same Oracle home that was used for the Standard Edition that you uninstalled. During the installation, be sure to select Enterprise Edition. When prompted, choose Software Only from the Database Configuration screen.

9. Start up your database.

Your database is now upgraded to Enterprise Edition.

Known Issue with the Deinstallation Tool for This Release

Cause: After upgrading from 11.2.0.1 or 11.2.0.2 to 11.2.0.3, deinstallation of the Oracle home in the earlier release of Oracle Database may result in the deletion of the old Oracle base that was associated with it. This may also result in the deletion of data files, audit files, etc., which are stored under the old Oracle base.

Action: Before deinstalling the Oracle home in the earlier release, edit the `orabase_cleanup.lst` file found in the *`$Oracle_Home`*`/utl` directory and remove the "oradata" and "admin" entries. Then, deinstall the Oracle home using the 11.2.0.3 deinstallation tool.

About Moving From Enterprise Edition to Standard Edition of Oracle Database

To properly convert from an Enterprise Edition database to a Standard Edition database you must perform an Export/Import operation. Oracle recommends using the Standard Edition Export utility to export the data. If you only install Standard Edition software, then some data dictionary objects become invalid and create problems when maintaining the database.

Enterprise Edition contains data dictionary objects that are not available in Standard Edition. The Export/Import operation does not introduce data dictionary objects specific to the Enterprise Edition, because the `SYS` schema objects are not exported.

After the Import in the Standard Edition database, you are only required to drop all user schemas related to Enterprise Edition features, such as the `MDSYS` account used with Oracle Spatial.

See Also: Chapter 7, "Moving Data Using Oracle Data Pump"

About Upgrading from Oracle Database Express Edition to Oracle Database

To upgrade Oracle Database 10*g* Express Edition (Oracle Database XE) to Oracle Database 11*g*, you must install Oracle Database 11*g* on the same system as Oracle Database XE and use DBUA to perform the upgrade.

See Also:

- *Oracle Universal Installer and OPatch User's Guide for Windows and UNIX*
- Chapter 3, "Upgrading to the New Release of Oracle Database"

2 Preparing to Upgrade Oracle Database

The process of upgrading Oracle Database includes understanding system considerations and requirements and troubleshooting various issues before actually performing the upgrade steps. Before you upgrade Oracle Database, you must become familiar with the new features and behavior changes. In preparation for upgrading you install the new Oracle software. The new Oracle software for this release provides the latest Pre-Upgrade Information Tool to help you understand requirements and complete pre-upgrade tasks.

This chapter contains the following topics:

- The Main Steps for Preparing to Upgrade Oracle Database
- Testing the Upgrade Process for Oracle Database
- Testing the Upgraded Test Oracle Database

The Main Steps for Preparing to Upgrade Oracle Database

In preparation for upgrading Oracle Database, you review the new features, determine the best upgrade path and method. Oracle recommends that you test the upgrade process and prepare a backup strategy.

Complete the following tasks to prepare to upgrade:

- Become Familiar with New Oracle Database Features for Upgrading
- Determine the Upgrade Path for Upgrading Oracle Database
- Choose an Upgrade Method for Upgrading Oracle Database
- Choose a Location for the New Oracle Home for Upgrading Oracle Database
- Develop a Testing Plan to Upgrade Oracle Database
- Prepare a Backup Strategy for Upgrading Oracle Database

Become Familiar with New Oracle Database Features for Upgrading

Before you plan the upgrade process, become familiar with the features of the new Oracle Database 11*g* release. *Oracle Database New Features Guide* is a good starting point for learning the differences between Oracle Database releases. Also, check specific guides in the Oracle Database 11*g* documentation library to find information about new features for a certain component. For example, see *Oracle Real Application Clusters Administration and Deployment Guide* for changes in Oracle Real Application Clusters.

Note:

- Oracle Database training classes are an excellent way to learn how to take full advantage of the features and functions available with Oracle Database. More information can be found at `http://education.oracle.com/`
- My Oracle Support provides detailed notes on how to obtain the latest patches, plus tools for lifecycle management and automated patching. For information about getting started with My Oracle Support, go to `http://www.oracle.com/us/support/software/premier/my-oracle-support-068523.html`

See Also: My Oracle Support Note ID 854428.1, "Patch Set Updates for Oracle Products" at `https://support.oracle.com`

Determine the Upgrade Path for Upgrading Oracle Database

The path that you must take to upgrade to the new Oracle Database 11*g* release depends on the release number of your current database. It might not be possible to directly upgrade from your current release of Oracle Database to the latest release. Depending on your current release, you might be required to upgrade through one or more intermediate releases to upgrade to the new Oracle Database 11*g* release.

For example, if the current database is running release 9*i*, then follow these steps:

1. Upgrade release 9.0.1.4 to release 10.2.0.4 using the instructions in *Oracle Database Upgrade Guide* Release 2 (10.2).
2. Upgrade release 10.2.0.4 to the new Oracle Database 11*g* release using the instructions in this guide.

Table 2–1 contains the required upgrade path for each release of Oracle Database. Use the upgrade path and the specified documentation to upgrade your database.

Table 2–1 Supported Upgrade Paths for Upgrading Oracle Database

Current Release	Upgrade Path
9.0.1.3 (or earlier)	Direct upgrade is *not* supported. Upgrade to an intermediate Oracle Database release before you can upgrade to the new Oracle Database 11*g* release, as follows: ■ 9.0.1.3 (or earlier) -> 9.0.1.4 -> 10.2.0.4 -> 11.2 When upgrading to an intermediate Oracle Database release, follow the instructions in the intermediate release's documentation. Then, upgrade the intermediate release database to the new Oracle Database 11*g* release using the instructions in Chapter 3, "Upgrading to the New Release of Oracle Database".
9.2.0.8 10.1.0.5 10.2.0.2 11.1.0.6	Direct upgrade to the new Oracle Database 11*g* release is supported from 9.2.0.8 or higher, 10.1.0.5 or higher, 10.2.0.2 or higher, and 11.1.0.6 or higher. Note that Oracle Clusterware release 10.2.0.*n* must be at release 10.2.0.3 (or higher), before you attempt to upgrade it to Oracle Clusterware 11*g*. See "About Upgrading an Oracle Real Application Clusters (Oracle RAC) Database" on page 3-4. For release 9.2.0.3, you must first upgrade to an intermediate Oracle Database release, as follows: 9.2.0.3 (or earlier) -> 9.2.0.8 -> 11.2 To upgrade to the new Oracle Database 11*g* release, follow the instructions in Chapter 3, "Upgrading to the New Release of Oracle Database".

> **See Also:** "Supported Releases for Downgrading Oracle Database" on page 6-1 for information related to downgrading your database

Choose an Upgrade Method for Upgrading Oracle Database

The upgrade methods you can use to upgrade your database to the new Oracle Database 11*g* release are:

- Database Upgrade Assistant (DBUA)
- Manual Upgrade
- Oracle Data Pump Export and Import and Oracle Database Upgrade

Database Upgrade Assistant (DBUA)

Database Upgrade Assistant (DBUA) interactively steps you through the upgrade process and configures the database for the new Oracle Database 11*g* release. DBUA automates the upgrade process by performing all of the tasks normally performed manually. DBUA makes appropriate recommendations for configuration options such as tablespaces and redo logs. You can then act on these recommendations.

DBUA provides support for Oracle Real Application Clusters (Oracle RAC). In an Oracle RAC environment, DBUA upgrades all the database and configuration files on all nodes in the cluster.

Manual Upgrade

A manual upgrade consists of running SQL scripts and utilities from a command line to upgrade a database to the new Oracle Database 11*g* release.

While a manual upgrade gives you finer control over the upgrade process, it is more susceptible to error if any of the upgrade or pre-upgrade steps are either not followed or are performed out of order.

Before the Upgrade The following list provides a high-level summary of the manual upgrade steps:

- Analyze the database using the Pre-Upgrade Information Tool. The Pre-Upgrade Information Tool is a SQL script that is supplied with the new Oracle Database 11*g* release, and DBUA uses this script as part of its upgrade process. Run the script on the database you are upgrading.

  The Pre-Upgrade Information Tool displays warnings about possible upgrade issues with the database. It also displays information about required initialization parameters for the new Oracle Database 11*g* release.

- Prepare the new Oracle home.

  > **See Also:** "Choose a Location for the New Oracle Home for Upgrading Oracle Database" on page 5 for more information

- Perform a backup of the database.

Depending on the release of the database being upgraded, you might be required to perform additional pre-upgrade steps:

- Adjust the parameter file for the upgrade.
- Remove obsolete initialization parameters.
- Adjust initialization parameters that might cause upgrade problems.

See Also: *Oracle Database Administrator's Guide* and *Oracle Real Application Clusters Administration and Deployment Guide*

- Set the `COMPATIBLE` parameter if not already explicitly set.

See Also: "Setting the COMPATIBLE Initialization Parameter" on page 4-14 for information about setting the `COMPATIBLE` initialization parameter

After the Upgrade Review the upgrade spool log file and use the Post-Upgrade Status Tool. The Post-Upgrade Status Tool is a SQL script that ships with the new Oracle Database 11*g* release. You run the Post-Upgrade Status Tool in the environment of the new release. The Post-Upgrade Status Tool can be run any time after upgrading the database.

See Also: "Upgrading Oracle Database Manually" on page 3-47

Oracle Data Pump Export and Import and Oracle Database Upgrade

Unlike DBUA or a manual upgrade, the Oracle Data Pump Export and Import utilities physically copy data from your current database to a new database. When upgrading from Oracle Database 10*g* Release 1 (10.1) or higher, Data Pump Export and Import are recommended for higher performance and to ensure support for new datatypes.

The Export utility of the current database copies specified parts of the database into an export dump file. Then, the Import utility of the new Oracle Database 11*g* release loads the exported data into a new database. However, the new Oracle Database 11*g* database must currently exist before it can be loaded from the export dump file.

When importing data from an earlier release, the Import utility of the new Oracle Database 11*g* release makes appropriate changes to data definitions as it reads export dump files from earlier releases.

The following sections highlight aspects of Export/Import that might help you to decide whether to use Export/Import to upgrade your database.

Note:

- If your database is earlier than Oracle Database release 10.1, then you can use the original Export and Import utilities to perform a full or partial export from your database, followed by a full or partial import into a new Oracle Database 11*g* database. Export/Import can copy a subset of the data in a database, leaving the original database unchanged.
- The original Export utility is no longer being updated to support new datatypes.

Export/Import Effects on Upgraded Databases The Export/Import upgrade method does not change the current database, which enables the database to remain available throughout the upgrade process. However, if a consistent snapshot of the database is required (for data integrity or other purposes), then the database must run in restricted mode or must otherwise be protected from changes during the export procedure. Because the current database can remain available, you can, for example, keep an existing production database running while the new Oracle Database 11*g* database is being built at the same time by Export/Import. During the upgrade, to maintain complete database consistency, changes to the data in the database cannot be permitted without the same changes to the data in the new Oracle Database 11*g* database.

Most importantly, the Export/Import operation results in a completely new database. Although the current database ultimately contains a copy of the specified data, the upgraded database might perform differently from the original database. For example, although Export/Import creates an identical copy of the database, other factors, such as disk placement of data and unset tuning parameters, might cause unexpected performance problems.

Export/Import Benefits Upgrading using Data Pump Export/Import:

- Defragments the data. You can compress the imported data to improve performance.
- Restructures the database. You can create new tablespaces or modify existing tables, tablespaces, or partitions to be populated by imported data.
- Facilitates side-by-side testing of the old and new versions of Oracle Database because an entirely new database is created.
- Enables the copying of specified database objects or users. Importing only the objects, users, and other items you need is useful for establishing a test environment for the new software on only a subset of the production data. Data Pump Export / Import provides flexible data subsetting capabilities.
- Serves as a backup archive - you can use a full database export as an archive of the current database.
- Enables the upgraded database to be established on an operating system or hardware platform that is different from that which is supporting the database being upgraded.
- Network-based Data Pump Import allows the new Oracle database to be directly loaded across the network from the old database being upgraded. Thus, no intervening dump files are required.

Time Requirements for Export/Import Upgrading an entire database by using Export/Import can take a long time, especially compared to using DBUA or performing a manual upgrade. Therefore, you might be required to schedule the upgrade during non-peak hours or make provisions for propagating to the new database any changes that are made to the current database during the upgrade.

See Also: Chapter 7, "Moving Data Using Oracle Data Pump"

Choose a Location for the New Oracle Home for Upgrading Oracle Database

You must choose a location for Oracle home on the new Oracle Database 11*g* release that is separate from the Oracle home of your current release. You cannot install the new software into the same location for Oracle home as your current release, unless you are installing an Oracle Database 11*g* patchset release. For a patchset release, you can use the same Oracle Database 11*g* Oracle home.

Using separate installation directories enables you to keep your existing software installed along with the new software. This method enables you to test the upgrade process on a test database before replacing your production environment entirely.

Develop a Testing Plan to Upgrade Oracle Database

You need a series of carefully designed tests to validate all stages of the upgrade process. Executed rigorously and completed successfully, these tests ensure that the process of upgrading the production database is well understood, predictable, and successful. Perform as much testing as possible before upgrading the production database. Do not underestimate the importance of a complete and repeatable testing process.

The types of tests to perform are the same whether you use Real Application Testing features like Database Replay or SQL Performance Analyzer, or perform testing manually.

Your test plan must include these types of tests:

- Oracle Database Upgrade Testing
- Minimal Testing for Upgrading Oracle Database
- Functional Testing for Upgrading Oracle Database
- High Availability Testing for Upgrading Oracle Database
- Integration Testing for Upgrading Oracle Database
- Performance Testing for Upgrading Oracle Database
- Volume and Load Stress Testing for Upgrading Oracle Database

Oracle Database Upgrade Testing

Upgrade testing entails planning and testing the upgrade path from your current software to the new Oracle Database 11*g* release, whether you use DBUA, perform a manual upgrade, or use Export/Import or other data-copying methods. Regardless of the upgrade method you choose, you must establish, test, and validate an upgrade plan.

Minimal Testing for Upgrading Oracle Database

Minimal testing entails moving all or part of an application from the current database to the new database and running the application without enabling any new database features. Minimal testing might not reveal problems that would appear in an actual production environment. However, minimal testing immediately reveals any application startup or invocation problems.

Functional Testing for Upgrading Oracle Database

Functional testing is a set of tests in which new and existing features and functions of the system are tested after the upgrade. Functional testing includes all database, networking, and application components. The objective of functional testing is to verify that each component of the system functions as it did before upgrading and to verify that new functions are working properly.

High Availability Testing for Upgrading Oracle Database

High availability testing entails:

- Ensuring that Recovery Time Objective (RTO) and Recovery Point Objective (RPO) business requirements are still met by the upgraded system. For example, in an Oracle RAC environment, injecting node or instance failures during stress testing help evaluate if the Oracle RAC recovery capability has changed.
- Testing your fallback plans and procedures.

- Checking the database performance and stability, and resolving performance problems.

> **See Also:** The *Oracle Database High Availability Overview* and "The Upgrade Companion" Web site available in Note 785351.1 on My Oracle Support at `http://support.oracle.com/`.

Integration Testing for Upgrading Oracle Database

Integration testing examines the interactions among components of the system. Consider the following factors when you plan your integration testing:

- Pro*C/C++ applications running against a new Oracle Database 11*g* instance should be tested to ensure that there are no problems with the new software.
- Graphical user interfaces should be tested with other components.
- Subtle changes in the new Oracle Database 11*g* release, such as data types, data in the data dictionary (additional rows in the data dictionary, object type changes, and so on) can have an effect all the way up to the front-end application, even if the application is not directly connected to a new Oracle Database 11*g* instance.
- If the connection between two components involves Oracle Net or Oracle Net Services, then those connections should also be tested and stress tested.

Performance Testing for Upgrading Oracle Database

Performance testing of the new database compares the performance of various SQL statements in the new database with the performance of those same statements in the current database. Before upgrading, you should understand the performance profile of the application under the current database. Specifically, you should understand the calls the application makes to the database server.

This section describes the following types of performance testing:

- Database Replay
- SQL Performance Analyzer
- SQL Plan Management

Database Replay Starting with Oracle Database 11*g* Release 1 (11.1), you can use the new Database Replay feature to perform real-world testing of a database upgrade on your site's production workload before actually upgrading the production database. This feature captures the actual database workload on the production system and replays it on the test system. It also provides analysis and reporting to highlight potential problems—for example, errors encountered, divergence in performance, and so forth. In addition, all the regular Enterprise Manager performance monitoring and reporting tools such as Automatic Database Diagnostic Monitor, Automatic Workload Repository (AWR), and Activity Session History are available to address any problems.

> **Note:** You can change the stored procedure logic in the database but the stored PL/SQL procedures that implement the application logic *must* maintain the same interfaces as before the upgrade. If an upgrade affects the stored procedures of an application, then the workload might not be replayable. By using the Database Replay tool in this way, you have good diagnostics to see if the new application logic in the server is performing as expected after the upgrade.

> **See Also:** *Oracle Database Real Application Testing User's Guide* for complete information about how to capture and replay workloads

SQL Performance Analyzer Starting with Oracle Database 11*g* Release 1 (11.1), you can use the SQL Performance Analyzer to forecast the impact of system changes on a SQL workload. SQL Performance Analyzer enables evaluating the impact of a change such as database upgrade by identifying the SQL statements impacted by the upgrade and measuring their performance divergence. The analysis enables you to assess the overall effect of the upgrade on SQL performance and makes it possible to avoid any negative outcome before users can be impacted.

> **See Also:** *Oracle Database Real Application Testing User's Guide* for complete information and examples using the SQL Performance Analyzer to perform what-if analysis on potential database changes

SQL Plan Management SQL plan management prevents performance regressions resulting from sudden changes to the execution plan of a SQL statement by providing components for capturing, selecting, and evolving SQL plan information. SQL plan management is a preventative mechanism that records and evaluates the execution plans of SQL statements over time, and builds SQL plan baselines composed of a set of existing plans known to be efficient. The SQL plan baselines are then used to preserve performance of corresponding SQL statements, regardless of changes occurring in the system.

A database upgrade that installs a new optimizer version usually results in plan changes for a small percentage of SQL statements, with most of the plan changes resulting in either no performance change or improvement. However, certain plan changes may cause performance regressions.

SQL plan management prevents performance regressions resulting from sudden changes to the execution plan of a SQL statement by providing components for capturing, selecting, and evolving SQL plan information. If you are performing a database upgrade that installs a new optimizer version, then it can result in plan changes for a small percentage of SQL statements, with most of the plan changes resulting in either no performance change or improvement. However, certain plan changes may cause performance regressions.

With SQL plan management, the optimizer automatically manages execution plans and ensures that only known or verified plans are used. When a new plan is found for a SQL statement, the plan is not used until it has been verified by the database to have comparable or better performance than the current plan. Therefore, if you seed SQL plan management with your current (pre-Oracle Database 11*g*) execution plan, which is to become the SQL plan baseline for each statement, then the optimizer uses these plans after the upgrade. If the Oracle Database 11*g* optimizer determines that a different plan is necessary, then the new plan is queued for verification and is not used until it has been confirmed to have comparable or better performance than the current plan.

There are two ways to seed or populate a SQL Management Base (SMB) with execution plans:

- Automatic capture of execution plans (available starting with Oracle Database 11*g*)
- Bulk load execution plans or preexisting SQL plan baselines

Bulk loading of execution plans or SQL plan baselines is especially useful when upgrading a database from a previous release to Oracle Database 11*g*. SQL plans that are bulk loaded are automatically accepted and added to existing or new plan histories as SQL plan baselines.

To bulk load the SQL Management Base as part of an upgrade:

- Populate the execution plans for a given SQL Tuning Set (STS), as described in "Bulk Loading a SQL Management Base with a SQL Tuning Set (STS)" on page 2-9.

  Or

- Unpack existing SQL plan baselines from a staging table, as described in "Unpacking Existing Oracle Database SQL Plan Baselines from a Staging Table" on page 2-9.

Bulk Loading a SQL Management Base with a SQL Tuning Set (STS)

To bulk load the SQL Management Base with an execution plan from an STS

1. In Oracle Database 10*g* Release 2 (10.2), create an STS that includes the execution plan for each of the SQL statements.
2. Load the STS into a staging table and export the staging table into a dump file.
3. Import the staging table from a dump file into Oracle Database 11*g* and unload the STS.
4. Use Oracle Enterprise Manager or `DBMS_SPM.LOAD_PLANS_FROM_SQLSET` to load the execution plans into the SQL Management Base.

Unpacking Existing Oracle Database SQL Plan Baselines from a Staging Table

Perform the steps in this procedure to test and tune all of your critical SQL queries on an Oracle Database 11*g* test environment and then move those exact SQL execution plans to your Oracle Database 11*g* production environment.

To test and tune your critical SQL queries on an Oracle Database 11g test environment

1. On the Oracle Database 11*g* test system, after completing all testing and tuning, use the `DBMS_SPM.LOAD_PLAN_FROM_CURSOR_CACHE` procedure or Enterprise Manager to load all of the execution plans in the cursor cache into the SQL Management Base.
2. Create a staging table using the `DBMS_SPM.CREATE_STGTAB_BASELINE` procedure.
3. Pack the SQL plan baselines you created in step 1 into the staging table using the `DBMS_SPM.PACK_STGTAB_BASELINE` function.
4. Export the staging table into a flat file using the Export command or Data Pump.
5. Transfer this flat file to the target system.
6. Import the staging table from the flat file using the Import command or Data Pump.
7. Unpack the SQL plan baselines from the staging table into the SQL Management Base on the target system using the `DBMS_SPM.UNPACK_STGTAB_BASELINE` function.

> **See Also:** *Oracle Database Performance Tuning Guide* for more information about using SQL Plan Management

Volume and Load Stress Testing for Upgrading Oracle Database

Volume and load stress testing tests the entire upgraded database under high volume and loads. Volume describes the amount of data being manipulated. Load describes the level of concurrent demand on the system. The objective of volume and load testing is to emulate how a production system might behave under various volumes and loads.

Volume and load stress testing is crucial, but is commonly overlooked. Oracle has found that customers often do not conduct any kind of volume or load stress testing. Instead, customers often rely on benchmarks that do not characterize business applications. Benchmarks of the application should be conducted to uncover problems relating to functions, performance, and integration, but they cannot replace volume and load stress testing.

Load testing involves running an application load against the new release of the database to ensure that the application does not encounter problems such as new errors or performance issues under load conditions likely to be encountered in production. Many times, problems manifest under certain load conditions and are normally not seen in functional testing. The Database Replay feature is ideal for such load testing as it enables capturing the system workload from a production environment and replay it in identical fashion on the test system.

Prepare a Backup Strategy for Upgrading Oracle Database

The ultimate success of your upgrade depends heavily on the design and execution of an appropriate backup strategy.

To develop a backup strategy, consider the following questions:

- How long can the production database remain inoperable before business consequences become intolerable?
- What backup strategy is necessary to meet your availability requirements?
- Are backups archived in a safe, offsite location?
- How quickly can backups be restored (including backups in offsite storage)?
- Have recovery procedures been tested successfully?

Your backup strategy should answer all of these questions and include procedures for successfully backing up and recovering your database.

> **See Also:** *Oracle Database Backup and Recovery User's Guide* for information on database backups

Testing the Upgrade Process for Oracle Database

Create a test environment that does not interfere with the current production database.

Your test environment depends on the upgrade method you have chosen:

- If you plan to use DBUA or perform a manual upgrade, then create a test version (typically a subset) of the current production database to test the upgrade.
- If you plan to use Export/Import, then export and import small test pieces of the current production database.

Practice upgrading the database using the test environment. The best upgrade test, if possible for you to create, is performed on an exact copy of the database to be upgraded, rather than on a downsized copy or test data. If for some reason an exact copy is impractical, then carefully chose a representative subset of your data to move over to your test environment and test the upgrade on that data.

Ensure that you upgrade any OCI and precompiler applications that you plan to use with your new Oracle database. Then, you can test these applications on a sample database before upgrading your current production database.

> **See Also:** "Upgrading Precompiler and OCI Applications" on page 5-2 for more information

Testing the Upgraded Test Oracle Database

Perform the planned tests on the current database and on the test database that you upgraded to the new Oracle Database 11*g* release. Compare the results, noting anomalies. Repeat the test upgrade as many times as necessary.

Test the newly upgraded test database with existing applications to verify that they operate properly with a new Oracle database. You also might test enhanced functions by adding available Oracle Database features. However, first ensure that the applications operate in the same manner as they did in the current database.

> **See Also:** Chapter 5, "Upgrading Applications After Upgrading Oracle Database" for more information on using applications with Oracle Database

3

Upgrading to the New Release of Oracle Database

Upgrading to a new release of Oracle Database is a basic part of database maintenance and administration. Oracle provides the Database Upgrade Assistant and specialized scripts and tools to assist with the upgrade process and to automate many steps. This chapter discusses important information and requirements and guides you through the process of upgrading a database to Oracle Database 11*g* Release 2 (11.2).

This chapter contains the following topics:

- System Considerations and Requirements for Upgrading Oracle Database
- Installing the New Oracle Database Software for the Upgrade
- About the Latest Patch Set Updates and Any Required Patches
- Using the Pre-Upgrade Information Tool
- Configuring the Listener When Upgrading from Oracle9i
- Upgrading with Database Upgrade Assistant on Linux, UNIX, and Windows
- Optionally Performing an In-Place Upgrade (Into the Same Oracle Home)
- Upgrading Oracle Database Manually
- Troubleshooting the Upgrade of Oracle Database
- Rerunning the Upgrade for Oracle Database
- Cancelling the Upgrade for Oracle Database
- About Upgrading an Oracle ASM Instance

> **Caution: If you retain the old Oracle software, then never start the upgraded database with the old Oracle software. Only start the database with the executables in the new Oracle Database installation.**

System Considerations and Requirements for Upgrading Oracle Database

When you upgrade to a new release of Oracle Database, typically there are new or changed system requirements. The following topics describe the various checks and considerations to perform before starting an upgrade of Oracle Database:

- Relocating Existing Data Files to the New Oracle Database Environment
- About Upgrading PL/SQL Packages That Are Not Installed by Default

- About Upgrading Oracle ASM Installed with Oracle Grid Infrastructure
- Considerations for Upgrading Oracle Clusterware and Oracle ASM Instances
- About Upgrading with Read-Only and Offline Tablespaces
- About Upgrading Using Standby Databases
- About Upgrading Your Operating System
- Migrating Data to a Different Operating System
- About Upgrading Databases That Use Oracle Streams Downstream Capture
- About Upgrading Databases That Use Oracle Database Vault

Important: To upgrade to this release, you must install the Oracle Grid Infrastructure and Oracle Database software into a new Oracle home instead of applying a patch set to the existing Oracle home. This is referred to as an *out-of-place* upgrade and is different from patch set releases for earlier releases of Oracle Database, where the patch set was always installed *in place*.

Note: Oracle Grid Infrastructure does not support in-place upgrade. Only out-of-place upgrade into a new grid intrastructure home is supported.

See Also:

- *Oracle Database Net Services Administrator's Guide* for information about upgrade considerations for Oracle Net Services
- Your operating system-specific Oracle documentation for additional information about preparing to upgrade
- "About Rolling Upgrades for Oracle Database" on page 1-11 for information about rolling upgrades

Relocating Existing Data Files to the New Oracle Database Environment

Before you remove the old Oracle environment, you must relocate any data files in that environment to the new Oracle Database environment.

To relocate data files to the new Oracle Database environment

- Use Database Upgrade Assistant (DBUA) and select the **Move Database Files** option during the upgrade.

See Also:

"Upgrading with Database Upgrade Assistant on Linux, UNIX, and Windows" on page 24 for more information

Oracle Database Administrator's Guide, if you perform a manual upgrade, for information about relocating data files

About Upgrading PL/SQL Packages That Are Not Installed by Default

Packages that were previously installed on the database for which you are preparing to upgrade to the current release may not be upgraded automatically. You may need to separately check if the package is available in the current release and reinstall that package to ensure you have the latest version.

> **See Also:** "Managing and Updating Access Control Lists and Network Utility Packages" on page 16

About Upgrading Oracle ASM Installed with Oracle Grid Infrastructure

Oracle ASM is upgraded as part of the Oracle Grid Infrastructure upgrade. You cannot upgrade Oracle ASM before you upgrade Oracle Grid Infrastucture.

> **See Also:**
>
> - "About Upgrading an Oracle ASM Instance" on page 61
> - *Oracle Grid Infrastructure Installation Guide*

Considerations for Upgrading Oracle Clusterware and Oracle ASM Instances

Starting with Oracle Database 11*g* Release 2 (11.2.0.1), Oracle Clusterware and Oracle Real Application Clusters (Oracle RAC) must be installed into a separate, new home from the existing installation. This reduces the downtime required to upgrade a node in the cluster and facilitates the provisioning of clusters within an enterprise. The reduction in planned outage time required for cluster upgrades helps in meeting availability service levels and also makes it easier to provide a standard installation across the enterprise.

The following considerations pertain to upgrading Oracle Clusterware and Oracle ASM instances:

- For releases earlier than Oracle Database 11*g*, either all Oracle software installations were owned by the Oracle user, typically `oracle`, or Oracle Database software was owned by the user `oracle`, and Oracle Clusterware software was owned by a separate user, typically `crsuser`.
- Starting with Oracle Database 11*g*, the user account that is designated as *owner* of the release 10*g* Cluster Ready Services (CRS) software must perform the Oracle Clusterware 11*g* upgrade. The user account that is performing this upgrade must also be the user that owns the ASM home of the earlier release (that is, previous to release 11.2). If the pre-11.2 ASM home has a different owner, then the *owner* account must be changed before performing the upgrade.
- As of Oracle Database 11*g* release 2 (11.2), the Oracle Clusterware software must be upgraded to a new home location in the Oracle Grid Infrastructure home. Additionally, Oracle ASM and Oracle Clusterware (and Oracle Restart for single-instance databases) must run in the same Oracle Grid Infrastructure home. When upgrading Oracle Clusterware to release 11.2, OUI automatically calls Oracle ASM Cluster Assistant (ASMCA) to perform the upgrade into the Oracle Grid Infrastructure home.
- To upgrade to Oracle Database release 11.2.0.3, you must install the Oracle Grid Infrastructure and Oracle Database software into a new Oracle home instead of into the existing Oracle home.
- For a single-instance configuration, Oracle ASM and Oracle Restart run from the Oracle Grid Infrastructure home, and the cluster synchronization services daemon (CSS) runs from this same home. Thus, Oracle ASM and CSS are upgraded to Oracle Database 11*g* Release 2 (11.2) at the same time.

- For Oracle Database release 11.1 and earlier releases, if your configuration does not include Oracle ASM, then you must shut down the CSS daemon and delete the CSS service from the system by running the `localconfig` command with the `delete` option. For example:

  ```
  ORACLE_HOME/bin/localconfig delete
  ```

- You must ensure that the database compatibility attribute for Oracle ASM disk groups matches the compatibility parameter that is set in `init.ora`.

Determining If Oracle ASM is Included in the Current Database Configuration

If you do not know whether or not your configuration includes Oracle ASM, then issue the following SQL statement on the database instance:

```
select count(*) from v$asm_client where status = 'CONNECTED';
```

If this statement returns one or more rows, then the database is actively using an Oracle ASM disk group.

About Upgrading an Oracle Real Application Clusters (Oracle RAC) Database

You can use Database Upgrade Assistant (DBUA) to upgrade an existing Oracle RAC database to the current release of Oracle Database. DBUA guides you through the upgrade process and configures your database for the new release. DBUA automates the upgrade process and makes appropriate recommendations for configuration options such as tablespaces and online redo log files.

If you are manually upgrading an Oracle RAC database, then most of the actions are to be performed on only one node of the system. Actions that must be performed on more than one node are indicated in the relevant steps.

About Requirements for Time Synchronization on Oracle RAC

Oracle Clusterware 11*g* release 2 (11.2) requires time synchronization across all nodes within a cluster when Oracle RAC is deployed.

You can use one of the following options for time synchronizaion:

- Your operating system-configured network time protocol (NTP)
- Oracle Cluster Time Synchronization Service

> **See Also:** *Oracle Grid Infrastructure Installation Guide* for your operating system for information on configuring NTP and Oracle Cluster Time Synchronization Service

Considerations for Upgrading Oracle RAC and Databases That Use Oracle ASM

If you are upgrading Oracle RAC and your databases use Oracle ASM for data file storage, then you need to consider additional restrictions and requirements.

- A subset of nodes cannot be selected when upgrading from an earlier release to 11.2.0.3.

  Before the new database release 11.2.0.3 software can be installed on the system, the root script for upgrading Oracle Grid Infrastructure invokes ASMCA to upgrade Oracle ASM to release 11.2.0.3.

- Where Oracle Clusterware software has been upgraded to release 11.2.0.1, but where Oracle ASM is running an earlier release, then DBUA displays an error indicating that the Oracle ASM cluster has not been upgraded. In this case, the root script invokes Oracle ASM Cluster Assistant (ASMCA) to upgrade Oracle ASM to release 11.2.0.1 before installing any Oracle Database release 11.2.0.3 software on the system.

- When upgrading from release 11.2.0.1 to 11.2.0.3, DBUA needs the Oracle Clusterware stack to be up. If DBUA finds that the Oracle Clusterware stack is down on the local node, then it displays an error message indicating the commands that need to be run to start up the stack on the local node.
- DBUA does not permit a single-instance database (SIDB) release 11.2.0.3 to be installed on 11.2.0.1 clusters.
- If DBUA detects Oracle Restart, then it presents a page asking you whether to register the database with Oracle Restart. If you choose to register with Oracle Restart, then DBUA displays a message prompting you to first upgrade Oracle Grid infrastructure to release 11.2.0.3.

> **Note:** Oracle Restart was previously referred to as Oracle Single-Instance High Availability (SIHA).

About Upgrading System Authentication for Oracle ASM Instances

Starting with Oracle Database 11*g* Release 1 (11.1), you should use the `SYSASM` privilege to separate database management and storage management responsibilities. Also, you have the option to create separate operating system credentials for Oracle ASM and each database. This separation allows for an even greater division of database management and storage management responsibilities. For instance, if there are *n* databases using Oracle ASM on a given node, then you can configure *n* + 1 sets of operating system credentials groups whose members have `SYS` privileges: one `OSDBA` group for each database with `SYSDBA` privileges, and one `OSASM` group for the Oracle ASM instance with `SYSASM` privileges.

Considerations for Upgrading an Oracle ASM Instance

Before upgrading an Oracle ASM instance to the new Oracle Database 11*g* release, you must add a user and password combination to the password file that is local to a node's Oracle ASM instance using the SQL*Plus `CREATE USER` statement:

```
CREATE USER user_name IDENTIFIED BY password;
```

- It is only necessary to add a user and password combination to the password file when upgrading the Oracle ASM instance. It is not necessary when upgrading a database to the new Oracle Database 11*g* release without upgrading Oracle ASM.
- If the default Oracle Database 11*g* security settings are in place, then passwords must be at least eight characters, and passwords such as `welcome` and `oracle` are not allowed.

> **See Also:**
>
> - *Oracle Database Storage Administrator's Guide* for more information about authentication for accessing Oracle ASM instances
> - *Oracle Database SQL Language Reference* for complete syntax about `CREATE USER`
> - *Oracle Database Security Guide* for password management guidelines and other security recommendations

About Upgrading with Read-Only and Offline Tablespaces

Oracle Database 11*g* Release 1 (11.1) and later releases can read file headers created in previous releases, so you are not required to do anything to them during the upgrade. The only exception to this is if you want to transport tablespaces created previously than Oracle Database 11*g* Release 1 (11.1) to another platform. In this case, the file headers must be made read/write at some point before the transport. However, there are no special actions required on them during the upgrade.

The file headers of offline data files are updated later when they are brought online, and the file headers of read-only tablespaces are updated if and when they are made read/write sometime after the upgrade. You are never required to make read-only tablespaces read/write in any other circumstance.

> **See Also:** *Oracle Database Administrator's Guide* for more information about read-only tablespaces and transporting tablespaces between databases

About Upgrading Using Standby Databases

To upgrade the Oracle Database software when standby databases are present in an Oracle Data Guard configuration, see *Oracle Data Guard Concepts and Administration*. To upgrade or downgrade Oracle Database and Oracle Enterprise Manager software in an Oracle Data Guard broker configuration, see *Oracle Data Guard Broker*.

During a rolling upgrade, you can run different releases of Oracle Database software on the primary and standby databases while you upgrade them, one at a time, incurring minimal downtime on the primary database by using either of the following methods:

- SQL Apply and logical standby databases

  Starting with Oracle Database 10*g* Release 1 (10.1.0.3), you can use Oracle Data Guard SQL Apply on a logical standby database to perform a rolling upgrade to the new Oracle Database 11*g* release. For example, you can upgrade the Oracle Database software from patch set release 10.1.0.*n* to the next database 10.1.0.(*n*+1) patch set release, or upgrade Oracle Database 10*g* Release 1 (10.1) to Oracle Database 11*g* Release 2 (11.2).

- Use of physical standby database for rolling upgrades

  A physical standby database can now take advantage of the rolling upgrade feature provided by a logical standby. Through the use of the new `KEEP IDENTITY` clause option to the SQL `ALTER DATABASE RECOVER TO LOGICAL STANDBY` statement, a physical standby database can be temporarily converted into a logical standby database for the rolling upgrade, and then reverted back to the original configuration of a primary database and a physical standby database when the upgrade is done.

See Also:

- *Oracle Database High Availability Best Practices*
- The following Oracle Maximum Availability Architecture (MAA) white papers at `http://www.oracle.com/technetwork/database/features/availability/maa-090890.html`:

  "Rolling Database Upgrades for Physical Standby Databases Using Transient Logical Standby 11*g*"

  "Rolling Database Upgrades using Data Guard SQL Apply"

About Upgrading Your Operating System

When you upgrade to a new release of Oracle software, the operating system requirements may have changed. If required, upgrade the operating system before upgrading Oracle Database.

See Also:

- The *Oracle Database Installation Guide* for your platform to obtain a list of supported operating systems
- Your operating system-specific documentation for information about how to perform an operating system upgrade

Migrating Data to a Different Operating System

When using DBUA or when performing a manual upgrade, you *cannot* migrate data in a database on one operating system to a database on another operating system. For example, you cannot migrate data in an Oracle9*i* database on Solaris to an Oracle 11*g* database on Windows using DBUA.

To migrate Oracle Database software to a different operating system

1. Upgrade to the new Oracle Database 11*g* release on your current operating system platform following the instructions in this guide.
2. Test the upgraded database on your current operating system platform.
3. Use Oracle Data Guard and physical standby databases to migrate between operating systems.

   See Also: Note 413484.1 on My Oracle Support (formerly Oracle*MetaLink*) at `https://support.oracle.com/` for more information on Oracle Data Guard support for heterogeneous primary and standby systems in the same Oracle Data Guard configuration

4. If cross-platform physical standby database is not available for the platform combination to be migrated, then you can use the Oracle Database 11*g* cross-platform transportable tablespace feature or the Oracle Data Pump Export and Import utilities to migrate the upgraded database to the different operating system.

   Note: Transportable tablespaces do not support migrating `SYSTEM` or `SYSAUX` tablespaces. All non-segment user data, such as roles, triggers, views, and procedures, must be moved to the new operating system with scripts or export/import.

5. If the two operating systems are in the same endianness group, then you can use the Oracle Database 11*g* cross-platform transportable database feature to migrate the entire database.

6. You can also use Oracle Streams to migrate data between operating systems. Oracle Streams has data type limitations and restrictions, such as for advanced queues and object types, and it requires additional administrative overhead.

See Also:

- *Oracle Database Backup and Recovery User's Guide* for more information on transportable tablespaces
- *Oracle Database High Availability Overview* for more information on migrations using Data Guard or transportable databases
- *Oracle Streams Concepts and Administration* for more information on migrations using Oracle Streams
- Chapter 7, "Moving Data Using Oracle Data Pump"

About Upgrading Databases That Use Oracle Streams Downstream Capture

In an Oracle Streams replication environment, downstream captures means that a capture process runs on a database other than the source database. When you upgrade the databases in such an environment, upgrade the database with the downstream capture process before you upgrade the source database. Upgrading the databases in this order ensures that the downstream capture database can continue to function after the source database is upgraded.

See Also: *Oracle Streams Concepts and Administration* for more information about downstream capture

About Upgrading Databases That Use Oracle Database Vault

When upgrading from Oracle Database release 10.2, if you have enabled Oracle Database Vault in your current Oracle home, then you must disable Oracle Database Vault in the target Oracle home where the new release 11.2.0.3 software is installed. You must do this before upgrading the database. Enable Oracle Database Vault again once the upgrade is complete.

See Also: *Oracle Database Vault Administrator's Guide* for instructions about disabling Oracle Database Vault

Installing the New Oracle Database Software for the Upgrade

You must install the software for the new Oracle Database release before you can perform the upgrade of Oracle Database. The installation procedure for the new Oracle Database 11*g* Release 2 (11.2) installs the Oracle software into a new Oracle home. This is referred to as an *out-of-place* upgrade and is different from patch set releases for earlier releases of Oracle Database, where the patch set was always installed *in place*. Oracle strongly recommends that you follow the steps in this procedure to ensure minimal downtime for the upgrade process and integrity of the new binaries and software libraries.

Important: It is not possible to upgrade a database using DBUA when the source and target Oracle homes are owned by different users. Attempting to do so returns error PRKH-1014. Either ensure that the source and target databases have the same owner, or perform the manual steps described in "Upgrading Oracle Database Manually" on page 47.

To install the new Oracle Database software for this release

1. If you are upgrading an Oracle RAC database, then you must perform the following steps in the order shown:

   a. Upgrade Oracle Clusterware first as described in "Considerations for Upgrading Oracle Clusterware and Oracle ASM Instances" on page 3.

   Note: When upgrading a non-Oracle RAC database, you must run Oracle Net Configuration Assistant (NETCA) before running DBUA. See "Configuring the Listener When Upgrading from Oracle9i" on page 23. When upgrading an Oracle RAC database, as part of the Oracle Clusterware upgrade, OUI automatically runs NETCA to upgrade the network listener. Therefore, you do not need to manually run NETCA.

   b. Mount the Oracle Grid Infrastructure installation media.

   c. Perform operating system prerequisite checks on each of the nodes that you intend to upgrade, to ensure that they meet the system prerequisites for Oracle Grid Infrastructure (Oracle Clusterware and Oracle ASM).

   d. If necessary, perform patch upgrades of the earlier release of Oracle Clusterware or Oracle Cluster Ready Services software to the most recent patch version.

   e. Ensure that you are logged in as the user that you want to own the Oracle Grid Infrastructure installation, and run the Oracle Grid Infrastructure installation. Provide information as prompted by the installer.

   f. When prompted, open a separate terminal session, log in as `root`, and run `root.sh`.

   See Also:

   - *Oracle Grid Infrastructure Installation Guide*
   - *Oracle Real Application Clusters Installation Guide*

2. After upgrading Oracle Clusterware, follow the instructions in your Oracle operating system-specific documentation to prepare for installation of Oracle Database software and start the Oracle Universal Installer.

   - Oracle recommends that you run the Pre-Upgrade Information Tool before you upgrade using DBUA, so that you can preview the types of items DBUA checks. (See "Using the Pre-Upgrade Information Tool" on page 10.) You can then run DBUA independently after the installation is complete.
   - If you use Oracle Label Security, Oracle Database Vault, or both, then select **Enterprise Edition** on the Select Database Edition page, click **Select Options**,

and enable one or both components from the components list. See *Oracle Label Security Administrator's Guide* and *Oracle Database Vault Administrator's Guide* for more information.

When installation of Oracle Database software has completed successfully, click **Exit** to close Oracle Universal Installer.

See Also:

- "Configuring the Listener When Upgrading from Oracle9i" on page 23
- "Upgrading with Database Upgrade Assistant on Linux, UNIX, and Windows" on page 24
- "Performing an In-Place Upgrade for Single-Instance Oracle Database" on page 39
- "Performing an In-Place Upgrade for an Oracle RAC Database" on page 40

About the Latest Patch Set Updates and Any Required Patches

The software for Oracle Database 11*g* Release 2 (11.2) contains a full release that includes all the latest patches and updates for Oracle Database. It is not necessary to check for patch set updates before proceeding with the upgrade process for this release. However, Oracle recommends that you periodically check for patches and patch set updates as a part of database administration.

See Also:

- *Oracle Universal Installer and OPatch User's Guide for Windows and UNIX*
- "Oracle Database Upgrade Path Reference List" (Note ID 730365.1) on My Oracle Support at `https://support.oracle.com`, which contains an upgrade reference list for most available Oracle Database releases, including download information, patch numbers, and links to other notes

Using the Pre-Upgrade Information Tool

After you have installed the software for Oracle Database 11*g* Release 2 (11.2) and any required patches, Oracle recommends that you analyze your database before upgrading it to the new release. This is done by running the Pre-Upgrade Information Tool from the environment of the database you are to upgrade. The Pre-Upgrade Information Tool is a SQL script included with Oracle Database 11*g* Release 2 (11.2) software. This is a required step if you are upgrading manually; otherwise, the `catupgrd.sql` script terminates with errors. Running the Pre-Upgrade Information Tool is also recommended if you are upgrading with DBUA, so that you can preview the items that DBUA checks.

These topics contain additional information about the Pre-Upgrade Information Tool:

- About the Output of the Pre-Upgrade Information Tool
- Pre-Upgrade Information Tool Miscellaneous Warnings

> **See Also:** Note 884522.1 "How to Download and Run Oracle's Database Pre-Upgrade Utility" available from My Oracle Support at `https://support.oracle.com`, which contains the latest version of the Pre-Upgrade Information Tool. Oracle strongly recommends that you use the latest version of this script available in Note 884522.1.

To run the Pre-Upgrade Information Tool

1. Log in to the system as the owner of the environment of the database being upgraded.

   > **Important: The Pre-Upgrade Information Tool must be copied to and must be run from the environment of the database being upgraded.**

2. Start SQL*Plus.
3. Connect to the database instance as a user with `SYSDBA` privileges.
4. Set the system to spool results to a log file for later analysis:

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

5. Run the Pre-Upgrade Information Tool:

   ```
   SQL> @$11g_ORACLE_HOME/rdbms/admin/utlu112i.sql
   ```

6. Turn off the spooling of script results to the log file:

   ```
   SQL> SPOOL OFF
   ```

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

   > **Note:** Oracle *inter*Media became Oracle Multimedia in Oracle Database 11*g* Release 1 (11.1).

About the Output of the Pre-Upgrade Information Tool

The Pre-Upgrade Information Tool displays warnings about possible upgrade issues with the database. The output in Example 3–1 shows the report that is generated from running the Pre-Upgrade Information Tool after installing the software for Oracle Database 11*g* Release 2 (11.2). In addition to the warnings that the tool displays, you must address any errors described in the output of the Pre-Upgrade Information Tool before performing the upgrade.

For invalid objects or invalid components, Oracle recommends running the the `utlrp.sql` before starting the upgrade as a means to minimize the number of invalid objects and components marked with `WARNING`.

The following topics contain sample output of the Pre-Upgrade Information Tool and provide descriptions of each section of the output.

- Database Section Contents
- Tablespaces Section Contents
- Rollback Segments Section Contents
- Flashback Section Contents

- Update Parameters Section Contents
- Renamed Parameters Section Contents
- Obsolete/Deprecated Parameters Section Contents
- Components Section Contents
- Miscellaneous Warnings Section Contents
- Recommendations Section Contents

> **Important: If you see a warning about the presence of the release 10*g* `DMSYS` schema in the database, then you must drop the `DMSYS` schema before proceeding with the upgrade. See *Oracle Data Mining Administrator's Guide* for procedures.**

Example 3–1 Pre-Upgrade Information Tool Sample Output

```
Oracle Database 11.2 Pre-Upgrade Information Tool 07-12-2011 12:51:34
Script Version: 11.2.0.3.0 Build: 001
.
**********************************************************************
Database:
**********************************************************************
--> name:   O112
--> version:   11.2.0.1.0
--> compatible:    11.2.0.0.0
--> blocksize:   8192
--> platform:   Linux x86 64-bit
--> timezone file: V11
.
**********************************************************************
Tablespaces: [make adjustments in the current environment]
**********************************************************************
--> SYSTEM tablespace is adequate for the upgrade.
.... minimum required size: 701 MB
--> SYSAUX tablespace is adequate for the upgrade.
.... minimum required size: 453 MB
--> UNDOTBS1 tablespace is adequate for the upgrade.
.... minimum required size: 400 MB
--> TEMP tablespace is adequate for the upgrade.
.... minimum required size: 60 MB
--> EXAMPLE tablespace is adequate for the upgrade.
.... minimum required size: 78 MB
.
**********************************************************************
Flashback: OFF
**********************************************************************
**********************************************************************
Update Parameters: [Update Oracle Database 11.2 init.ora or spfile]
Note: Pre-upgrade tool was run on a lower version 64-bit database.
**********************************************************************
--> If Target Oracle is 32-Bit, refer here for Update Parameters:
-- No update parameter changes are required.
.

--> If Target Oracle is 64-Bit, refer here for Update Parameters:
-- No update parameter changes are required.
.
**********************************************************************
```

```
Renamed Parameters: [Update Oracle Database 11.2 init.ora or spfile]
**********************************************************************
-- No renamed parameters found. No changes are required.
.
**********************************************************************
Obsolete/Deprecated Parameters: [Update Oracle Database 11.2 init.ora or spfile]
**********************************************************************
-- No obsolete parameters found. No changes are required
.

**********************************************************************
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
--> OLAP Analytic Workspace [upgrade]  VALID
--> OLAP Catalog [upgrade]  VALID
--> Oracle Label Security [upgrade]  VALID
--> Oracle Database Vault [upgrade]  VALID
... To successfully upgrade Oracle Database Vault, choose
... 'Select Options' in Oracle installer and then select
... Oracle Label Security.
--> EM Repository [upgrade]  VALID
--> Oracle Text  [upgrade]  VALID
--> Oracle XML Database  [upgrade]  VALID
--> Oracle Java Packages [upgrade]  VALID
--> Oracle interMedia [upgrade]  VALID
--> Spatial [upgrade]  VALID
--> Expression Filter [upgrade]  VALID
--> Rule Manager [upgrade]  VALID
--> Oracle Application Express [upgrade]  VALID
... APEX will only be upgraded if the version of APEX in
... the target Oracle home is higher than the current one.
--> Oracle OLAP API [upgrade]  VALID
.
**********************************************************************
Miscellaneous Warnings
**********************************************************************
WARNING: --> Database is using a timezone file older than version 14.
.... After the release migration, it is recommended that DBMS_DST package
.... be used to upgrade the 11.2.0.1.0 database timezone version
.... to the latest version which comes with the new release.
WARNING: --> Database contains schemas with objects dependent on DBMS_LDAP
package.
.... Refer to the 11g Upgrade Guide for instructions to configure Network ACLs.
.... USER APEX_030200 has dependent objects.
.
**********************************************************************
Recommendations
**********************************************************************
Oracle recommends gathering dictionary statistics prior to
upgrading the database.
To gather dictionary statistics execute the following command
while connected as SYSDBA:

    EXECUTE dbms_stats.gather_dictionary_stats;
```

```
**********************************************************************
Oracle recommends reviewing any defined events prior to upgrading.

To view existing non-default events execute the following commands
while connected AS SYSDBA:
  Events:
    SELECT (translate(value,chr(13)||chr(10),' ')) FROM sys.v$parameter2
      WHERE  UPPER(name) ='EVENT' AND  isdefault='FALSE'

  Trace Events:
    SELECT (translate(value,chr(13)||chr(10),' ')) from sys.v$parameter2
      WHERE UPPER(name) = '_TRACE_EVENTS' AND isdefault='FALSE'

Changes will need to be made in the init.ora or spfile.

**********************************************************************
```

Database Section Contents

The Database section displays global database information about the current database, such as the database name, release number (version), compatibility level, blocksize, OS platform, and time zone file. A warning is displayed if you must adjust the `COMPATIBLE` initialization parameter before the database is upgraded.

> **See Also:** "Setting the COMPATIBLE Initialization Parameter" on page 4-14 for information about setting the `COMPATIBLE` initialization parameter

Tablespaces Section Contents

The Tablespaces 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 displays next to each tablespace confirming that the tablespace is adequate for the upgrade. If the minimum required size is not met, then you must make adjustments, which the tool recommends.

In a manual upgrade using SQL scripts and utilities, space must be added to tablespaces that do not have enough free space in the current database. These tablespace adjustments must be made before the database is upgraded. Some of these tasks are performed automatically by DBUA.

Rollback Segments Section Contents

The Rollback Segments section shows the status for rollback segments in the `SYSTEM` tablespace and displays a warning about any adjustments that need to be made before performing the upgrade. This section only appears if there are rollback segments.

Flashback Section Contents

The Flashback section shows whether flashback is ON or OFF, and displays warnings about pool size and cache size that must be increased. The status of `undo_management` is also displayed with any adjustment that must be made. You can execute an `ALTER SYSTEM SET` command to change the pool sizes and database cache size.

For example execute the following command to to set the `java_pool_size` to the recommended size as follows:

```
SQL> ALTER SYSTEM SET java_pool_size='nnnM' SCOPE=spfile;
```

The same can be done for changing the other settings.

Update Parameters Section Contents

The Update Parameters section displays a list of initialization parameters in the parameter file (`init.ora` or `spfile`) 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 the new Oracle Database 11*g* release.

If you are using Oracle ASM disk groups, then you must ensure that the database compatibility attribute for the disk groups matches the compatibility parameter that is set in `init.ora`.

> **See Also:** Appendix A, "Behavior Changes After Upgrading Oracle Database" for more information about changes to initialization parameters in this Oracle Database 11*g* release

Renamed Parameters Section Contents

The Renamed Parameters section displays a list of initialization parameters in the parameter file of the current database that are renamed in the new Oracle Database 11*g* release. New default values are also given.

> **See Also:** Appendix A, "Behavior Changes After Upgrading Oracle Database" for initialization parameters that are renamed in the new Oracle Database 11*g* release

Obsolete/Deprecated Parameters Section Contents

The Obsolete/Deprecated Parameters section displays a list of initialization parameters in the parameter file of the current database that are obsolete or deprecated in the new Oracle Database 11*g* release. Obsolete initialization parameters must be removed from the parameter file before the database is upgraded.

> **See Also:** Appendix A, "Behavior Changes After Upgrading Oracle Database" for a list of initialization parameters that are obsolete or deprecated in the new Oracle Database 11*g* release

Components Section Contents

The Components section displays a list of database components in the database to be upgraded or installed when the current database is upgraded to the new Oracle Database 11*g* release. Software components are pre-built code libraries that provide specific database funtionality like Oracle Text, Oracle Java packages, and Oracle RAC.

Miscellaneous Warnings Section Contents

The Miscellaneous Warnings section provides warnings about specific situations that require attention before or after the upgrade. For example, if the database is using a time zone file that is a version older than what is required for the upgrade, then a warning displays with the required action.

> **See Also:** "Pre-Upgrade Information Tool Miscellaneous Warnings" on page 16

Recommendations Section Contents

The Recommendations section provides Oracle recommendations, including the recommended SQL statements and commands, that should be performed before upgrading to the new Oracle Database 11*g* release.

Pre-Upgrade Information Tool Miscellaneous Warnings

Before upgrading to the new Oracle Database 11*g* release, Oracle recommends analysis of the information and warnings displayed by the Pre-Upgrade Information Tool. The following topics describe warnings and the appropriate actions to take.

- Updating the CONNECT Role from Earlier Releases
- Managing and Updating Access Control Lists and Network Utility Packages
- About Database Links with Passwords from Earlier Releases
- About Warnings for TIMESTAMP WITH TIME ZONE Data Type
- Decreasing Downtime for Gathering Optimizer Statistics (Optional)
- Identifying Invalid Objects
- Saving Database Control Files and Data with the emdwgrd Utility
- Verifying That Materialized View Refreshes Have Completed
- Ensuring That No Files Need Media Recovery
- Ensuring That No Files Are in Backup Mode
- Resolving Outstanding Distributed Transactions
- Synchronizing a Standby Database with the Primary Database
- Purging the Database Recycle Bin

Updating the CONNECT Role from Earlier Releases

After upgrading to the new Oracle Database 11*g* release from Oracle9*i* Release 2 (9.2) or Oracle Database 10*g* Release 1 (10.1), the CONNECT role has only the CREATE SESSION privilege. The other privileges granted to the CONNECT role in earlier releases are revoked during the upgrade.

To identify which users and roles in your database are granted the CONNECT role

1. Perform the following query:

```
SELECT grantee FROM dba_role_privs
     WHERE granted_role = 'CONNECT' and
                     grantee NOT IN (
                                'SYS', 'OUTLN', 'SYSTEM', 'CTXSYS', 'DBSNMP',
                                'LOGSTDBY_ADMINISTRATOR', 'ORDSYS',
                                'ORDPLUGINS',  'OEM_MONITOR', 'WKSYS',
'WKPROXY',
                                'WK_TEST', 'WKUSER', 'MDSYS', 'LBACSYS',
'DMSYS',
                                'WMSYS', 'EXFSYS', 'SYSMAN', 'MDDATA',
                                 'SI_INFORMTN_SCHEMA', 'XDB', 'ODM');
```

2. If users or roles require privileges other than CREATE SESSION, then grant the specific required privileges before upgrading.

   The upgrade scripts provided by Oracle adjust the privileges for the Oracle-supplied users.

Managing and Updating Access Control Lists and Network Utility Packages

The new Oracle Database 11*g* release includes fine-grained access control to the UTL_TCP, UTL_SMTP, UTL_MAIL, UTL_HTTP, and UTL_INADDR packages using Oracle

XML DB. If you have applications that use one of these packages, then you must install Oracle XML DB if it is not currently installed.

To update ACLs and Network Utility packages

1. Install Oracle XML DB if it is not currently installed.

   See Also: *Oracle XML DB Developer's Guide*

2. New behavior for the `DBMS_LDAP` PL/SQL package and the `HttpUriType` type requires the creation or update of access control lists (ACLs) after performing the upgrade to the new Oracle Database release 11.2.

   For example, if your application depends on the `DBMS_LDAP` package, then the error "ORA-24247: network access denied by access control list (ACL)" may occur. The logged-in user must have the `connect` privilege for the host and port specified by `DBMS_LDAP.init`.

3. If Oracle XML DB is installed on the database to be upgraded, and therefore one or all of the `UTL_TCP`, `UTL_SMTP`, `UTL_MAIL`, `UTL_HTTP`, and `UTL_INADDR` packages are installed, then you may need to re-install these packages after performing the upgrade to ensure that you have the latest version of these packages for the new release.

   See Also: "Managing Fine-Grained Access in PL/SQL Packages and Types" in *Oracle Database Security Guide* and "Assessing Dependencies and Adding ACLs for Network Utility Packages" on page 3-17

Assessing Dependencies and Adding ACLs for Network Utility Packages

After upgrading to the new Oracle Database 11*g* release, you may need to assess the dependencies of network utility packages and provide access by adding the appropriate access control lists (ACLs).

To check the status of access and add ACLs for network utility packages

1. Run the pre-upgrade information tool as described in "Using the Pre-Upgrade Information Tool" on page 3-10.

2. Check the output from the pre-upgrade information tool (*`upgrade_info`*`.log`) for messages such as the following:

   ```
   WARNING: --> Database contains schemas with objects dependent on network
   packages.
   .... Refer to the 11g Upgrade Guide for instructions to configure Network ACLs.
   .... USER WKSYS has dependent objects.
   .... USER SYSMAN has dependent objects.
   .... USER FLOWS_010600 has dependent objects.
   .
   ```

3. Query the `DBA_DEPENDENCIES` view to obtain more information about the dependencies. For example:

   ```
   SELECT * FROM DBA_DEPENDENCIES
   WHERE referenced_name IN ('UTL_TCP','UTL_SMTP','UTL_MAIL','UTL_HTTP','UTL_
   INADDR','DBMS_LDAP')
     AND owner NOT IN ('SYS','PUBLIC','ORDPLUGINS');
   ```

4. Prepare post-upgrade scripts now to make the scripts available for use in the test environment. This ensures the new access controls are part of your upgrade testing.

To configure network access control lists (ACLs) in the database so that these packages can work as they did in prior releases, see the example script provided in "Configuring Fine-Grained Access to External Network Services After Upgrading Oracle Database" on page 4-4. This script shows how to use the DBMS_NETWORK_ACL_ADMIN package to create, assign, and add privileges to the access control list.

5. After the upgrade, you must grant the specific required privileges. Access is based on the usage in the original database.

About Database Links with Passwords from Earlier Releases

This information is important only if you need to downgrade to your original database release after performing the upgrade. During the upgrade to the new Oracle Database 11*g* release from Oracle9*i* Release 2 (9.2) or Oracle Database 10*g* Release 1 (10.1), any passwords in database links are encrypted.

- To downgrade to the original release, all of the database links with encrypted passwords must be dropped before the downgrade. Consequently, the database links are nonexistent in the downgraded database.
- If you anticipate a requirement to be able to downgrade to your original release, then save the information about affected database links from the SYS.LINK$ table, so that you can re-create the database links after the downgrade.

About Warnings for TIMESTAMP WITH TIME ZONE Data Type

The time zone files that are supplied with Oracle Database 11*g* Release 2 (11.2) have been updated to reflect changes in transition rules for some time-zone regions. The changes might affect existing data of the TIMESTAMP WITH TIME ZONE data type.

The time zone behavior for Oracle Database 11*g* Release 2 (11.2) has changed significantly from earlier releases. You must make sure to obtain the latest time zone files before you upgrade the database. If the time zone file version of the database being upgraded is not the most recent version of the time zone file available for the new 11.2 release, then the Pre-Upgrade Information Tool displays a warning and describes how to proceed. Table 3–1 describes the warnings and summarizes how to resolve a mismatch in time zone file versions.

Caution: The TIMESTAMP WITH TIME ZONE data stored in the database can become corrupted during the upgrade if there is a time zone file version mismatch.

Table 3–1 Choices for Fixing the Time Zone File Version

IF the time zone version on the database being upgraded is...	THEN fix the time zone files...
Earlier than the most current version included in the new database release and the Pre-Upgrade Information Tool displays "Database is using a time zone file older than version *n*."	*After* completing the database upgrade. Use the DBMS_DST PL/SQL package and follow the instructions in "Steps to Upgrade Time Zone File and Timestamp with Time Zone Data" in *Oracle Database Globalization Support Guide*.

Table 3–1 (Cont.) Choices for Fixing the Time Zone File Version

IF the time zone version on the database being upgraded is...	THEN fix the time zone files...
Later than the version included in the new database release and the Pre-Upgrade Information Tool displays "Database is using a time zone file greater than version *n*."	*Before* beginning the database upgrade. You must patch (`RDBMS DST` patch) the new Oracle home with the same version of the time zone file as the one currently used in the database being upgraded. Otherwise the upgrade will fail. You must patch the Oracle home with the appropriate patch for the time zone file version in use. Apply the patch for each database to be upgraded. Otherwise, the upgrade script terminates without upgrading the database.

See Also:

- The support note "Updated DST Transitions and New Time Zones in Oracle Time Zone File Patches" (ID 412160.1) from My Oracle Support at `https://support.oracle.com`
- *Oracle Database Globalization Support Guide* for detailed descriptions of time zone upgrade

Decreasing Downtime for Gathering Optimizer Statistics (Optional)

When upgrading to the new Oracle Database 11*g* release, optimizer statistics are collected for dictionary tables that lack statistics. This statistics collection process can be time consuming for databases with a large number of dictionary tables, but 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

- Collect statistics before performing the actual database upgrade. Oracle recommends that you use the `DBMS_STATS.GATHER_DICTIONARY_STATS` procedure to gather these statistics. For example, you can enter the following SQL statement:

```
EXEC DBMS_STATS.GATHER_DICTIONARY_STATS;
```

Note: If you are using Oracle9*i* Release 2 (9.2), then you should use the `DBMS_STATS.GATHER_SCHEMA_STATS` procedure to gather statistics. To do this, you can run the scripts provided in Appendix B in this guide.

System Components and Schemas Used for Optimizer Statistics

Table 3–2 lists the system components and schemas that are checked for statistics collection during the upgrade.

Table 3–2 Statistics Collection for System Components and Schemas

Component Name	Schema
JServer JAVA Virtual Machine	`SYS`
OLAP Analytic Workspace	`SYS`
Oracle Database Catalog Views	`SYS`

Table 3–2 (Cont.) Statistics Collection for System Components and Schemas

Component Name	Schema
Oracle Database JAVA Packages	`SYS`
Oracle Database Packages and Types	`SYS, DBSNMP, OUTLN, SYSTEM, DIP`
Oracle Database Vault	`DVSYS`
Oracle Enterprise Manager	`SYSMAN`
Oracle Expression Filter	`EXFSYS`
Oracle Multimedia	`ORDSYS, ORDPLUGINS, SI_INFORMTN_SCHEMA`
Oracle Label Security	`LBACSYS`
Oracle OLAP API	`SYS`
Oracle Spatial	`MDSYS, MDDATA`
Oracle Text	`CTXSYS`
Oracle Workspace Manager	`WMSYS`
Oracle XDK	`SYS`
Oracle XML Database	`XDB`

Identifying Invalid Objects

Any invalid `SYS/SYSTEM` objects found before upgrading the database are stored in the table named `registry$sys_inv_objs`. Any invalid non-`SYS/SYSTEM` objects found before upgrading the database are stored in `registry$nonsys_inv_objs`.

To identify any new invalid objects due to the upgrade

- After the upgrade, run `ORACLE_HOME/rdbms/admin/utluiobj.sql`

Saving Database Control Files and Data with the emdwgrd Utility

If you plan to downgrade Oracle Enterprise Manager Database Control after upgrading to the new Oracle Database 11*g* release, then you must save your database control files and data *before* upgrading your database. Oracle provides the `emdwgrd` utility to use before upgrading your database to keep a copy of your database control files and data.

The `emdwgrd` utility resides in the `ORACLE_HOME/bin` directory in the new Oracle Database 11*g* release. The `emdwgrd` utility consists of `emdwgrd` and `emdwgrd.pl` for Linux and UNIX, and `emdwgrd.bat` and `emdwgrd.pl` for Windows. Before running the utility, you must install the software for the new Oracle Database 11*g* release, and then invoke the script from the new Oracle home. The `emdwgrd` utility, however, requires that you set `ORACLE_HOME` to the old Oracle home.

To save your database control files and data using emdwgrd

1. Install the software for the new Oracle Database 11*g* release. (This step is not required for an in-place patch set upgrade.)

   See Also: "Installing the New Oracle Database Software for the Upgrade" on page 3-8

2. Set `ORACLE_HOME` to your old Oracle home. (This step is not required for an in-place patch set upgrade.)
3. Set `ORACLE_SID` to the SID of the database being upgraded.

4. Set PATH, LD_LIBRARY_PATH, and SHLIB_PATH to point to the Oracle home from which the database is being upgraded.
5. Go to the Oracle home of the new Oracle Database 11*g* release.
6. Execute one of the following commands:
   - For a single-instance database, run the following command, where *old_SID* is the SID of the database being upgraded and *save_directory* is the path to the storage place you have chosen for your database control files and data:

     ```
     emdwgrd -save -sid old_SID -path save_directory
     ```

     Note: The script is in emdwgrd.sh on Linux and UNIX platforms. On Windows, the script is in emdwgrd.bat.

   - If the database is an Oracle RAC database, remote copy is required across the cluster nodes. Define an environment variable to indicate which remote copy is configured. For example:

     ```
     setenv EM_REMCP /usr/bin/scp
     ```

     Then, execute the following save command:

     ```
     emdwgrd -save -cluster -sid old_SID -path save_directory
     ```

     If the release 10*g* Oracle home is on a shared device, add -shared to the previous command line.
7. Enter the SYS password for the database to be upgraded.

Note: The DBUA backup and restore process also allows you to revert to your previous Oracle Enterprise Manager Database Control environment after upgrading your database. However, all user data that is accumulated between the time of upgrade and restore operations is lost. Saving your database control files and data enables you to downgrade both your database and database control. All user data is retained even though all database control data that is accumulated between the time of upgrade and downgrade is lost.

Verifying That Materialized View Refreshes Have Completed

You must wait until all materialized views have completed refreshing before upgrading. You can query the system to determine if there are any materialized view refreshes still in progress.

To determine if there are any materialized view refreshes still in progress

- Run the following query

```
SQL> SELECT * FROM sys.obj$ o, sys.user$ u, sys.sum$ s
WHERE o.type# = 42 AND bitand(s.mflags, 8) = 8;
```

Ensuring That No Files Need Media Recovery

Before upgrading the database, you must ensure that there are no files requiring media recovery. You can query the system to get a list of files and then recover them as appropriate.

To get a list of files that require media recovery

- Issue the following statement:

```
SQL> SELECT * FROM v$recover_file;
```

See Also:

- *Oracle Database Backup and Recovery User's Guide*
- "Performing Block Media Recovery" in *Oracle Database Backup and Recovery Reference*

Ensuring That No Files Are in Backup Mode

Files must not be in backup mode when performing the upgrade; therefore, you must wait until backups are completed. You can query the system to see a list of any files in backup mode and then take appropriate action by either waiting for the backup to complete, or by aborting any backups that are not needed.

To get a list of files in backup mode

- Issue the following statement:

```
SQL> SELECT * FROM v$backup WHERE status != 'NOT ACTIVE';
```

Resolving Outstanding Distributed Transactions

You must resolve outstanding distributed transactions before performing the upgrade. You can do this by first querying to see any pending transactions, and then committing the transactions. You must wait until all pending distributed transactions have been committed.

To resolve outstanding distributed transactions

1. Issue the following statement:

```
SQL> SELECT * FROM dba_2pc_pending;
```

2. If the query in the previous step returns any rows, then issue the following statements:

```
SQL> SELECT local_tran_id FROM dba_2pc_pending;
SQL> EXECUTE dbms_transaction.purge_lost_db_entry('');
SQL> COMMIT;
```

Synchronizing a Standby Database with the Primary Database

If a standby database exists, then you must synchronize it with the primary database.

To check if a standby database exists and to synchronize it

1. Issue the following query:

```
SQL> SELECT SUBSTR(value,INSTR(value,'=',INSTR(UPPER(value),'SERVICE'))+1)
FROM v$parameter
WHERE name LIKE 'log_archive_dest%' AND UPPER(value) LIKE 'SERVICE%';
```

2. If the query in the previous step returns a row, then synchronize the standby database with the primary database.

   - Make sure all the logs are transported to the standby server after a final log switch in the primary.
   - Start the recovery of the standby database with the `NODELAY` option.

Purging the Database Recycle Bin

The database recycle bin must be empty before you begin the upgrade process. You use the PURGE statement to remove items and their associated objects from the recycle bin and release their storage space.

To empty the database recycle bin

- Issue the following command:

```
SQL> PURGE dba_recyclebin
```

> **Caution:** **The database recycle bin must be empty during the upgrade process to avoid possible** ORA-00600 **errors and to minimize the upgrade time.**

Configuring the Listener When Upgrading from Oracle9*i*

If you are upgrading from Oracle9*i* and a listener was not configured in the Oracle9*i* repository, then you must run Oracle Net Configuration Assistant to configure the listening protocol address and service information for the new Oracle Database 11*g* database, including a `listener.ora` file, *before* running DBUA. A new version of the listener is required for an Oracle Database 11*g* database. Previous versions of the listener are not supported for use with an Oracle Database 11*g* database. However, it is possible to use the new version of the listener with previous versions of Oracle Database.

If you are upgrading an Oracle RAC database, then you have the following options:

- Upgrade the Oracle RAC database with DBUA, which automatically migrates the listener from your old Oracle home to the new Oracle Grid Infrastructure 11*g* Release 2 home.

  > **Note:** Beginning with Oracle Database 11*g* Release 2, you must administer the listener by using the `lsnrctl` command in the Oracle Grid infrastructure home. Do not attempt to use the `lsnrctl` commands from Oracle home locations for previous releases.

- If you are upgrading from Oracle9*i* or upgrading manually without using DBUA, then run Oracle Net Configuration Assistant before upgrading the Oracle RAC database.

  This is a two-step option. You must first run Oracle Net Configuration Assistant from the old Oracle home to remove the old listener. Then you must run Oracle Net Configuration Assistant again from the Oracle home of the new Oracle Database 11*g* release to create a new listener.

  You must remove the old listener before creating a new one. If you attempt to create a new listener from the new Oracle home first, and use the same name and port as the old listener, then Oracle Net Configuration Assistant returns an error.

  > **See Also:** *Oracle Database Net Services Administrator's Guide* for complete information about using Oracle Net Configuration Assistant

Upgrading with Database Upgrade Assistant on Linux, UNIX, and Windows

DBUA provides a graphical user interface (GUI) to guide you through the upgrade of a database, or you can invoke it in silent mode, which does not present a user interface. The following topics guide you through the process of upgrading a database using Database Upgrade Assistant (DBUA).

- Considerations for Using DBUA
- About the DBUA Graphical User Interface
- Using DBUA to Upgrade the Database on Linux, UNIX, and Windows Systems
- Using DBUA in Silent Mode to Upgrade Oracle Database

Considerations for Using DBUA

- **Important:** If you terminate the upgrade, but do not restore the database, then you cannot restart DBUA. Instead, you must continue with a manual (command line) upgrade as described in "Manually Upgrading Oracle Database" on page 50.
- DBUA can be used to upgrade from earlier Oracle Database 11*g* patch releases as well as from earlier major Oracle Database releases on both Oracle RAC databases and Oracle Database single-instance databases. The procedure to upgrade patch releases is no different from the normal upgrade procedure.
- You must run the Oracle Net Configuration Assistant before running DBUA. See "Configuring the Listener When Upgrading from Oracle9i" on page 3-23.

  If you are upgrading an existing Oracle Enterprise Manager Database Control or if you are configuring a new Oracle Enterprise Manager Database Control for the new Oracle Database 11*g* release, then the listener must be running before upgrading Oracle Enterprise Manager. If DBUA detects that an Oracle Enterprise Manager upgrade or configuration is requested, and DBUA does not see a listener running, then it prompts you and starts the default listener in either of these cases.
- If you upgrade a cluster database using DBUA, then you must make sure the `CLUSTER_DATABASE` initialization parameter is set to `TRUE`.
- If the database instance is not running, then DBUA tries to start the instance with the default initialization parameter file. If that fails, then DBUA prompts you to provide the name of the correct initialization parameter file or to start the instance. If the instance is up and running, then DBUA connects to it.
- If you terminate the upgrade, but do not restore the database, then you cannot restart DBUA until you start up the existing database in `UPGRADE` mode using the new Oracle Database 11*g* server. You cannot go back to the original server unless you restore your database.

  For Oracle RAC, you cannot re-run DBUA once you terminate the upgrade. If you need to re-run the upgrade, then you must run DBUA from the restored backup of the database you are upgrading.
- If you restore your database manually (not using DBUA), then remove the `Welcome_`*`SID`*`.txt` file, which is located in the *ORACLE_HOME*/cfgtoollogs/dbua/logs/ directory, before starting DBUA. The presence of this file indicates to DBUA that this is a re-run operation.

About the DBUA Graphical User Interface

If you installed Oracle Database 11*g* Release 2 (11.2) and specified that you are upgrading an existing database, then DBUA starts automatically. However, if you did not specify that you are upgrading an existing database, then you can start DBUA independently after installation is complete.

While the upgrade is in process, DBUA shows the upgrade progress for each component. DBUA writes detailed trace and log files and produces a complete HTML report for later reference. To enhance security, DBUA automatically locks new user accounts in the upgraded database. DBUA then proceeds to create new configuration files (parameter and listener files) in the new Oracle home.

DBUA does not begin the upgrade until all of the pre-upgrade steps are completed.

> **See Also:** "Using the Pre-Upgrade Information Tool" on page 10

These topics provide additional information about DBUA:

- Checks Performed by Oracle DBUA
- Upgrade Scripts Invoked by Oracle DBUA

Checks Performed by Oracle DBUA

DBUA performs the following checks before the upgrade:

- Invalid user accounts or roles
- Invalid data types or invalid objects
- Desupported character sets
- Adequate resources, including rollback segments, tablespaces, and free disk space
- Missing SQL scripts needed for the upgrade
- Listener running (if Oracle Enterprise Manager Database Control upgrade or configuration is requested)
- Oracle Database software linked with Database Vault option

> **Note:** If Database Vault is enabled, then DBUA returns an error asking you to disable Database Vault before upgrading. See "About Upgrading Databases That Use Oracle Database Vault" on page 3-8.

Upgrade Scripts Invoked by Oracle DBUA

During the upgrade, DBUA automatically modifies or creates new required tablespaces and invokes the appropriate upgrade scripts, as follows:

- If the datafiles are auto extensible and have enough disk space to grow, then DBUA continues with the upgrade.
- If the datafiles are not autoextensible, then DBUA prompts you and makes the files auto extensible.
- If the tablespaces are auto extensible and the MAXSIZE initialization parameter needs adjustment, then DBUA prompts for the same and adjusts the MAXSIZE parameter.

- If there is not enough disk space to grow, then DBUA prompts you to create space (by adding more datafiles). DBUA does not add new datafiles because DBUA cannot determine where to create the files.

Optionally, DBUA backs up all necessary files.

Using DBUA to Upgrade the Database on Linux, UNIX, and Windows Systems

Complete the following steps to upgrade a database using the DBUA graphical user interface. These steps are performed from within the new Oracle home where the latest database software has been installed as described in "Installing the New Oracle Database Software for the Upgrade" on page 8 .

> **Important: If you terminate the upgrade, but do not restore the database, then you cannot restart DBUA. Instead, you must continue with a manual (command line) upgrade as described in "Manually Upgrading Oracle Database" on page 50.**

To upgrade a database using DBUA on Linux, UNIX, and Windows

1. Start DBUA from the Oracle home where the new database software has been installed:

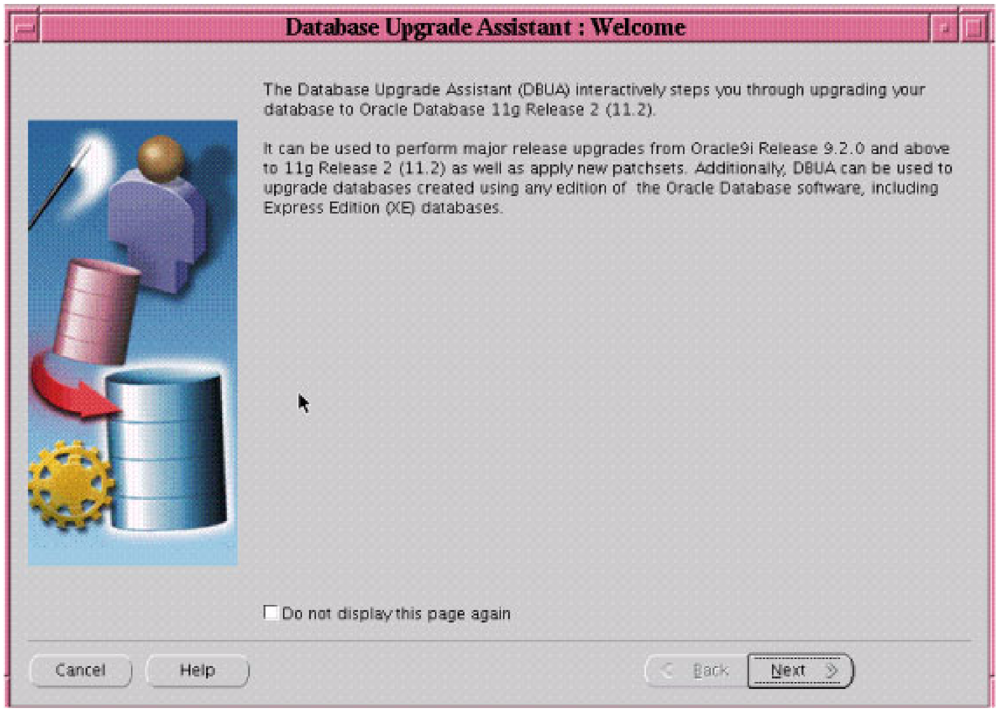
   - On Linux or UNIX platforms, enter the following command at a system prompt in the new Oracle Database 11*g* Release 2 (11.2) home:

     ```
     dbua
     ```

     > **Note:** The `dbua` executable is usually located in the *ORACLE_HOME*/bin directory.

   - On Windows operating systems, select **Start** > **Programs** > **Oracle -** ***HOME_NAME*** > **Configuration and Migration Tools** > **Database Upgrade Assistant**.

   The DBUA Welcome screen displays.



Database Upgrade Assistant : Welcome

The Database Upgrade Assistant (DBUA) interactively steps you through upgrading your database to Oracle Database 11g Release 2 (11.2).

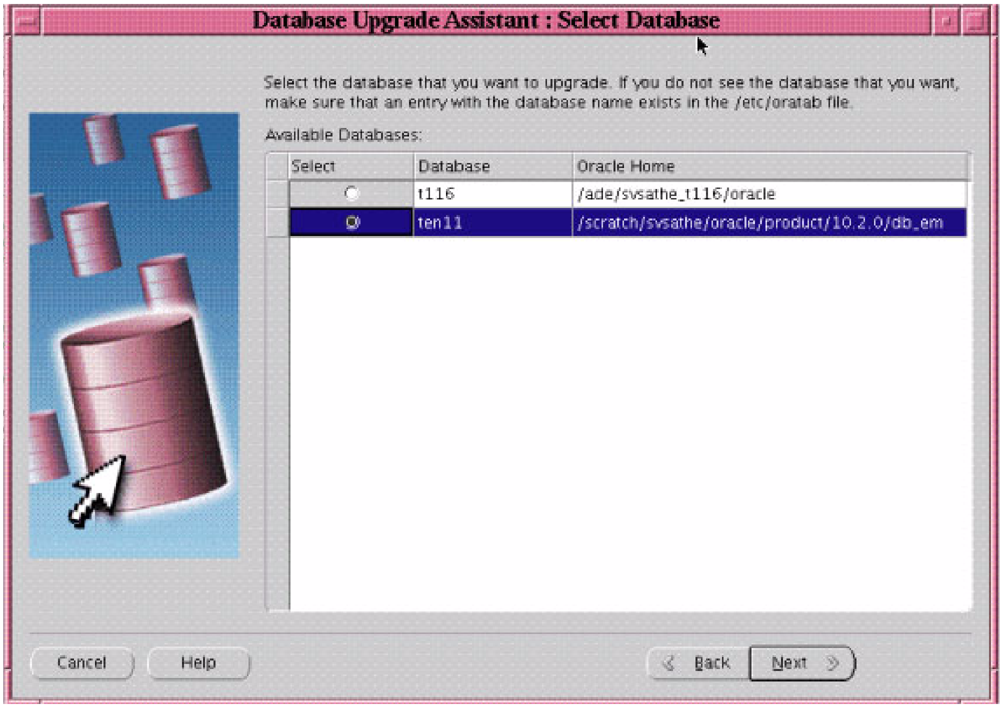
It can be used to perform major release upgrades from Oracle9i Release 9.2.0 and above to 11g Release 2 (11.2) as well as apply new patchsets. Additionally, DBUA can be used to upgrade databases created using any edition of the Oracle Database software, including Express Edition (XE) databases.

□ Do not display this page again

2. If you need help at any screen or want to consult more documentation about DBUA, then click **Help** to open the online help.

   Click **Next**.

   The Select Database page appears, listing the databases available for upgrade.



Database Upgrade Assistant : Select Database

Select the database that you want to upgrade. If you do not see the database that you want, make sure that an entry with the database name exists in the /etc/oratab file.

Available Databases:

Select	Database	Oracle Home
○	t116	/ade/svsathe_t116/oracle
●	ten11	/scratch/svsathe/oracle/product/10.2.0/db_em

3. Select the database you want to upgrade to Oracle Database 11*g* Release 2 (11.2).

You can select only one database at a time. If you do not see the database that you want, then make sure an entry with the database name exists in the `oratab` file in the etc directory.

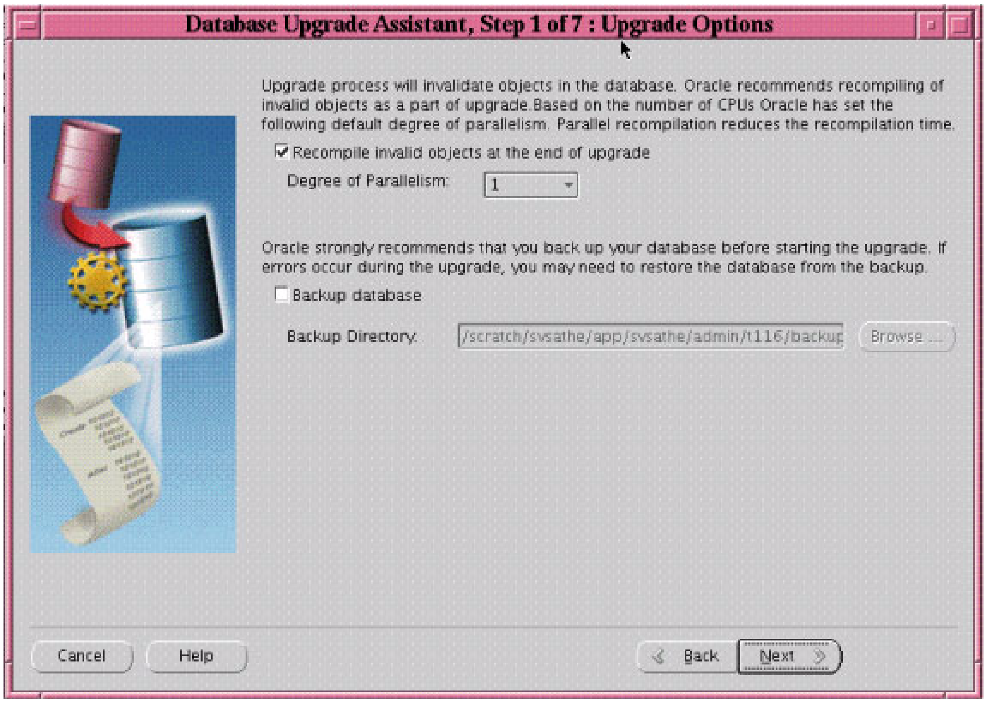
If you are running DBUA from a user account that does not have `SYSDBA` privileges, then you must enter the user name and password credentials to enable `SYSDBA` privileges for the selected database.

Click **Next**.

DBUA analyzes the database, performing pre-upgrade checks and displaying warnings as necessary. Examples of DBUA database checks include:

- Redo log files whose size is less than 4 MB. If such files are found, then DBUA gives the option to drop/create new redo log files.
- Obsolete or deprecated initialization parameters.

When DBUA finishes its checks, the Upgrade Options screen displays.



Database Upgrade Assistant, Step 1 of 7 : Upgrade Options

Upgrade process will invalidate objects in the database. Oracle recommends recompiling of invalid objects as a part of upgrade.Based on the number of CPUs Oracle has set the following default degree of parallelism. Parallel recompilation reduces the recompilation time.

☑ Recompile invalid objects at the end of upgrade

Degree of Parallelism: 1

Oracle strongly recommends that you back up your database before starting the upgrade. If errors occur during the upgrade, you may need to restore the database from the backup.

□ Backup database

Backup Directory: /scratch/svsathe/app/svsathe/admin/t116/backup

4. The Upgrade Options screens enables you to set the following options:

Recompile invalid objects at the end of upgrade

Select **Recompile invalid objects at the end of upgrade** if you want DBUA to recompile all invalid PL/SQL modules after the upgrade is complete.

When you upgrade a database to Oracle Database 11*g* Release 2 (11.2), many of the PL/SQL modules in your database become invalid. As a result, all existing PL/SQL modules in an `INVALID` state must be recompiled, such as packages, procedures, types, and so on.

By default, Oracle Database recompiles invalid PL/SQL modules as they are used. The time it takes to recompile the module can result in poor performance when you begin to use your newly upgraded database.

To eliminate these performance issues, select **Recompile invalid objects at the end of upgrade**. When you select this option, DBUA recompiles all the invalid

PL/SQL modules immediately after the upgrade is performed. This ensures that you do not experience any performance issues later, as you begin using your newly upgraded database.

Note: Selecting **Recompile invalid objects at the end of upgrade** is equivalent to running the `utlrp.sql` script, located in the *ORACLE_HOME*/`rdbms/admin` directory, which is used to recompile stored PL/SQL and Java code.

The task of recompiling all the invalid PL/SQL modules in your database can take a significant amount of time and increase the time it takes to complete your database upgrade. If you have multiple CPUs, then you can reduce the time it takes to perform this task by taking advantage of parallel processing on your available CPUs. If you have multiple CPUs available, then DBUA automatically adds an additional section to the Recompile Invalid Objects screen and automatically determines the number of CPUs you have available.

Degree of Parallelism

DBUA also provides a recommended degree of parallelism, which determines how many parallel processes are used to recompile your invalid PL/SQL modules. Specifically, DBUA sets the degree of parallelism to one less than the number of CPUs you have available. For example, if you have three CPUs available for processing, then DBUA selects 2 from the Degree of Parallelism menu. You can adjust this default value by selecting a new value from the Degree of Parallelism menu.

Backup Database

Select **Backup database** if you want DBUA to back up your database for you.

Important: Oracle strongly recommends that you back up your database before starting the upgrade. If errors occur during the upgrade, then you might be required to restore the database from the backup.

If you use DBUA to back up your database, then it makes a copy of all your database files in the directory you specify in the Backup Directory field. DBUA performs this cold backup automatically after it shuts down the database and before it begins performing the upgrade procedure. The cold backup does not compress your database files, and the backup directory must be a valid file system path. You cannot specify a raw device for the cold backup files.

In addition, DBUA creates a batch file in the specified directory. You can use this batch file to restore the database files:

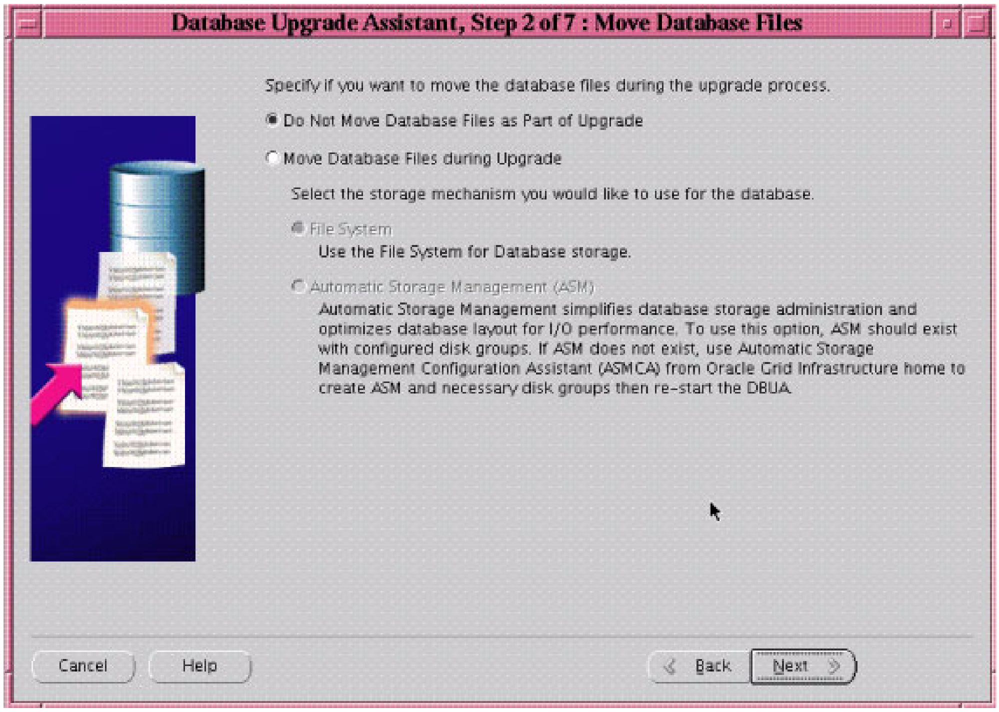
- On Windows operating systems, the file is called `db_name_restore.bat`.
- On Linux or UNIX platforms, the file is called `db_name_restore.sh`.

If you choose not to use DBUA for your backup, then Oracle assumes you have backed up your database using your own backup procedures.

Note: DBUA does not back up Oracle ASM databases. You must manually back them up on your own.

Click **Next**.

If you are upgrading a single-instance database or Oracle Express Edition (XE), then the Move Database Files screen appears. If you are upgrading an Oracle Real Application Clusters database, then the Move Database Files screen does not display.



Database Upgrade Assistant, Step 2 of 7 : Move Database Files

Specify if you want to move the database files during the upgrade process.

● Do Not Move Database Files as Part of Upgrade

○ Move Database Files during Upgrade

Select the storage mechanism you would like to use for the database.

● File System

Use the File System for Database storage.

○ Automatic Storage Management (ASM)

Automatic Storage Management simplifies database storage administration and optimizes database layout for I/O performance. To use this option, ASM should exist with configured disk groups. If ASM does not exist, use Automatic Storage Management Configuration Assistant (ASMCA) from Oracle Grid Infrastructure home to create ASM and necessary disk groups then re-start the DBUA.

5. Select one of the following options:

   - **Do Not Move Database Files as Part of Upgrade**
   - **Move Database Files during Upgrade**

   If you choose to move database files, then you must also make one of the following selections:

   - **File System**

     Your database files are moved to the host file system.

   - **Oracle Automatic Storage Management (Oracle ASM)**

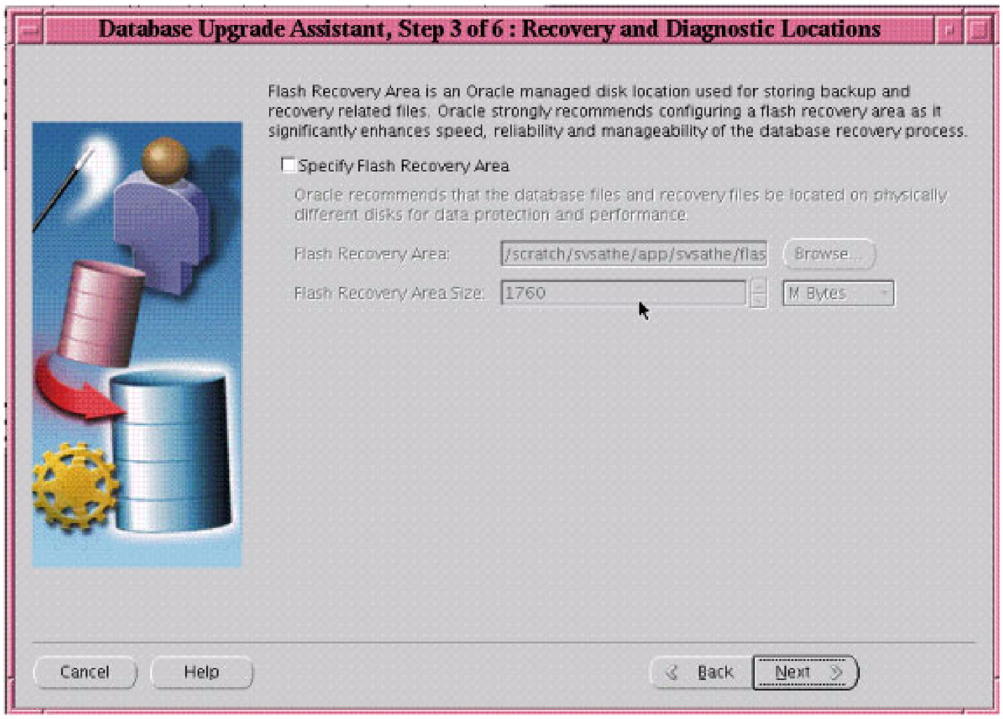
     Your database files are moved to Oracle ASM storage, which must currently exist on your system. If you do not have an Oracle ASM instance, then you can create one using Automatic Storage Management Configuration Assistant (ASMCA) from the Oracle Grid Infrastructure home and then restart DBUA.

     See Also:

     - *Oracle Grid Infrastructure Installation Guide* for information about installing and configuring Oracle ASM
     - "Managing Oracle ASM Instances With ASM Configuration Assistant" in *Oracle Database Storage Administrator's Guide*

     Click **Next**.

The Recovery and Diagnostic Locations screen displays, where you can designate a flash recovery area for your database.



Database Upgrade Assistant, Step 3 of 6 : Recovery and Diagnostic Locations

Flash Recovery Area is an Oracle managed disk location used for storing backup and recovery related files. Oracle strongly recommends configuring a flash recovery area as it significantly enhances speed, reliability and manageability of the database recovery process.

□ Specify Flash Recovery Area

Oracle recommends that the database files and recovery files be located on physically different disks for data protection and performance.

Flash Recovery Area: /scratch/svsathe/app/svsathe/flas

Flash Recovery Area Size: 1760 M Bytes

Flash recovery area is an Oracle managed disk location used for storing backup and recovery related files. Oracle strongly recommends configuring a flash recovery area, because it significantly enhances speed, reliability, and manageability of the database recovery process. The location of the flash recovery area is also used by Enterprise Manager if you enable local management and daily backups on the Management Options screen.

If you selected the **Move Database Files during Upgrade** option in step 5, or if an Oracle Express Edition database is being upgraded to Oracle Enterprise Edition, then you must configure a flash recovery area. If a flash recovery area is currently configured, then the current settings are retained but the screen displays to allow you to override these values.

6. Do one of the following:

   - Accept the default flash recovery area location
   - Enter the full path to a different location in the **Flash Recovery Area** field
   - Click **Browse** and select a different flash recovery area location

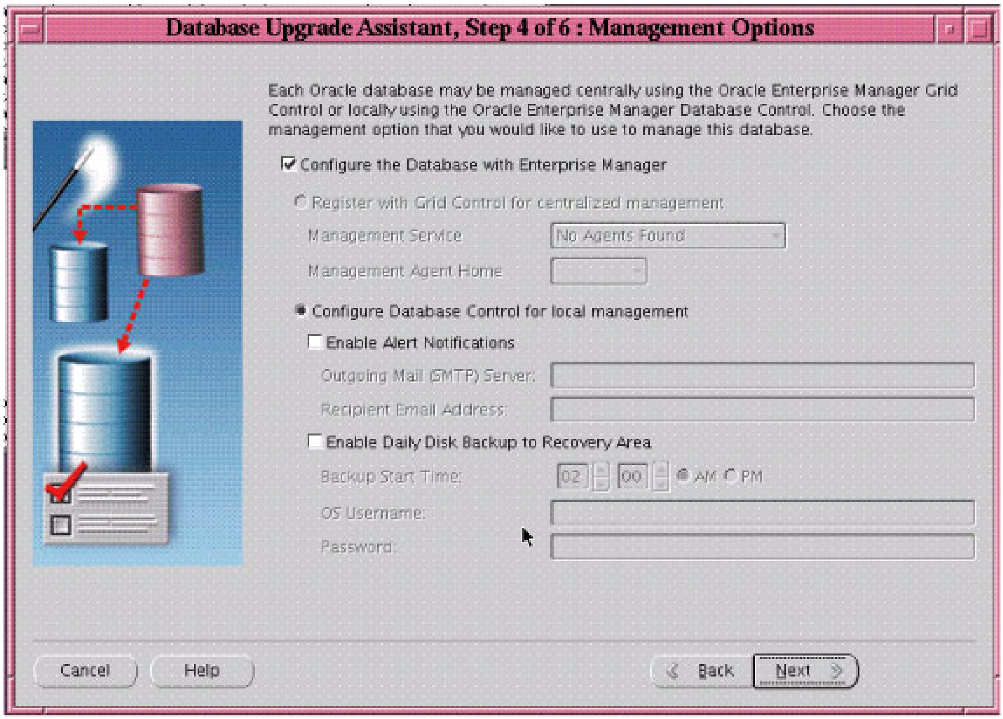
   Do one of the following:

   - Accept the default size for the flash recovery area
   - Enter a different value in the **Flash Recovery Area Size** field
   - Click the up or down arrows to adjust the flash recovery area size

   See Also: *Oracle Database 2 Day DBA* for more information about configuring the flash recovery area

   Click **Next**.

If no other database is currently being monitored with Enterprise Manager, then the Management Options screen displays.



Database Upgrade Assistant, Step 4 of 6 : Management Options

Each Oracle database may be managed centrally using the Oracle Enterprise Manager Grid Control or locally using the Oracle Enterprise Manager Database Control. Choose the management option that you would like to use to manage this database.

☑ Configure the Database with Enterprise Manager

○ Register with Grid Control for centralized management

Management Service: No Agents Found

Management Agent Home:

● Configure Database Control for local management

☐ Enable Alert Notifications

Outgoing Mail (SMTP) Server:

Recipient Email Address:

☐ Enable Daily Disk Backup to Recovery Area

Backup Start Time: 02 00 ● AM ○ PM

OS Username:

Password:

At the Management Options screen, you have the option of setting up your database so it can be managed with Enterprise Manager. Enterprise Manager provides Web-based management tools for managing individual database instances, as well as central management tools for managing your entire Oracle environment, including multiple databases, hosts, application servers, and other components of your network.

Before you can register the database with Oracle Enterprise Manager Grid Control, an Oracle Enterprise Manager Agent must be configured on the host computer.

7. To set up your database to be managed with Enterprise Manager, select **Configure the Database with Enterprise Manager** and then select one of the following options:

   - **Register with Grid Control for centralized management**

     If you select this option, then you must also select an Oracle Management Service from the **Management Service** drop-down list. When you run DBUA, it checks to see if the Oracle Management Agent has been installed on the host computer. If no Oracle Management Agent is found, then the Grid Control option is not available.

     When you finish installing the Oracle Database software, the database is automatically available as a managed target within Oracle Enterprise Manager Grid Control.

   - **Configure Oracle Enterprise Manager Database Control for local management**

     If you are not centrally managing your Oracle environment, then you can still use Enterprise Manager to manage your database. When you install Oracle Database, you automatically install Oracle Enterprise Manager Database

Control, which provides Web-based features for monitoring and administering the single-instance or cluster database you are installing.

Do the following steps if you want the `SYSMAN` user (the default Super Administrator and owner of the Management Repository schema) to receive E-mail notification when a metric for a specified condition reaches a critical or warning threshold:

– Select **Enable Alert Notifications**

– Enter the name of the e-mail server you want to use for notifications in the **Outgoing Mail (SMTP) Server** field

– Enter the e-mail address of the `SYSMAN` user in the **Recipient Email Address** field

For example, Enterprise Manager can send an e-mail when a target goes down or when there are database space usage problems. E-mail notifications are enabled immediately upon installation.

Do the following steps to use the Oracle suggested backup strategy to back up your entire database with a minimum amount of configuration:

– Select **Enable Daily Disk Backup to Recovery Area**

– Enter a start time in the **Backup Start Time** field

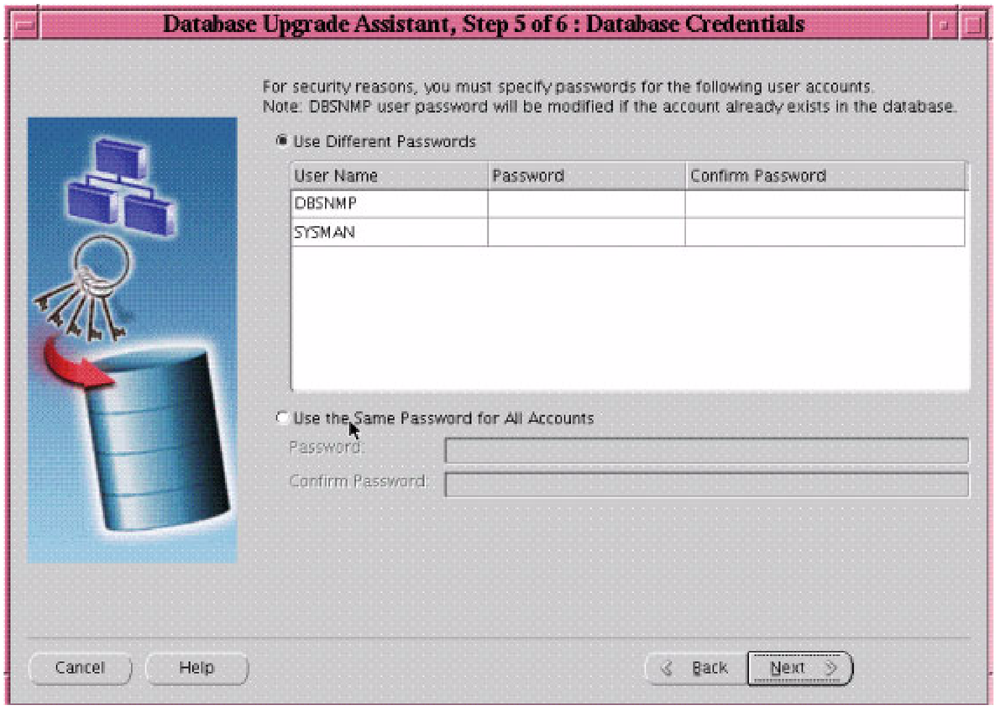
– Enter host credentials in the **OS Username** and **Password** fields

If you select the Enable Daily Disk Backup to Recovery Area option, then Enterprise Manager is configured to back up your database immediately after you finish upgrading Oracle Database. Enterprise Manager backs up the database to the flash recovery area. Later, you can use Enterprise Manager to customize your backup strategy further.

On Windows the user whose credentials you enter for the backup must be granted the `Logon as a batch job` privilege in the Local Security Policies of Windows. If the chosen user does not have this privilege, then the backup job fails.

Click **Next**.

The Database Credentials screen displays.



Database Upgrade Assistant, Step 5 of 6 : Database Credentials

For security reasons, you must specify passwords for the following user accounts.
Note: DBSNMP user password will be modified if the account already exists in the database.

● Use Different Passwords

User Name	Password	Confirm Password
DBSNMP		
SYSMAN		

○ Use the Same Password for All Accounts

Password:

Confirm Password:

8. Select one of the following options:

   - **Use Different Passwords**

     If you choose to use different passwords, then you must enter a password in the **Password** and **Confirm Password** columns for each account in the table.

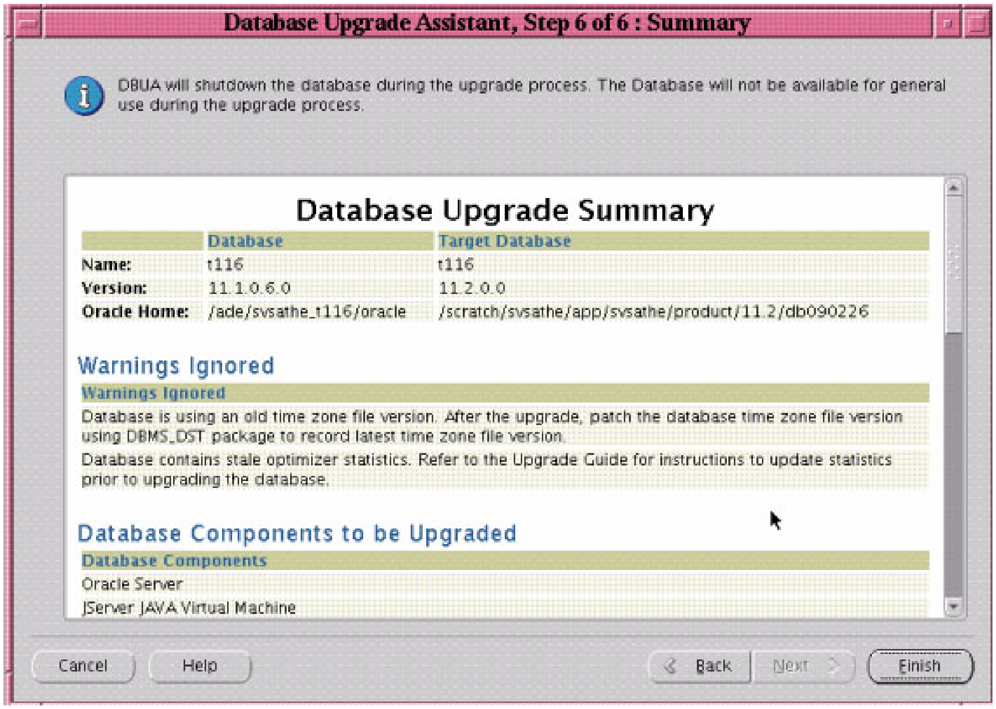
   - **Use the Same Password for All Accounts**

     If you choose to use the same password, then you must enter that password in the **Password** and **Confirm Password** fields.

     Note: If the default Oracle Database 11*g* Release 2 (11.2) security settings are in place, then new passwords must be at least eight characters, and passwords such as `welcome` and `oracle` are not allowed. See *Oracle Database Security Guide* for more information.

   Click **Next**.

   The Summary screen appears.



Database Upgrade Assistant, Step 6 of 6 : Summary

DBUA will shutdown the database during the upgrade process. The Database will not be available for general use during the upgrade process.

Database Upgrade Summary

	Database	Target Database
Name:	t116	t116
Version:	11.1.0.6.0	11.2.0.0
Oracle Home:	/ade/svsathe_t116/oracle	/scratch/svsathe/app/svsathe/product/11.2/db090226

Warnings Ignored

Warnings Ignored
Database is using an old time zone file version. After the upgrade, patch the database time zone file version using DBMS_DST package to record latest time zone file version.
Database contains stale optimizer statistics. Refer to the Upgrade Guide for instructions to update statistics prior to upgrading the database.

Database Components to be Upgraded

Database Components
Oracle Server
JServer JAVA Virtual Machine

9. The Summary screen shows the following information about the upgrade before it starts:

   - Name, version, and Oracle home of the old and new databases
   - Database backup location, available space, and space required
   - Warnings ignored
   - Database components to be upgraded
   - Initialization parameters changes
   - Database files location
   - Listener registration

   See Also: "Setting the COMPATIBLE Initialization Parameter" on page 4-14 for information about setting the `COMPATIBLE` initialization parameter after the upgrade

   Check all of the specifications. Then do one of the following:

   - Click **Back** if anything is incorrect until you reach the screen where you can correct it.
   - Click **Finish** if everything is correct.

   The Progress screen displays and DBUA begins the upgrade.

10. After the upgrade has completed, the following message is displayed on the Progress screen:

    ```
    Upgrade is complete. Click "OK" to see the results of the upgrade.
    ```

11. The Upgrade Results screen displays a description of the original and upgraded databases and changes made to the initialization parameters. The screen also

shows the directory where various log files are stored after the upgrade. You can examine these log files to obtain more details about the upgrade process.

Note: An HTML version of the Upgrade Results is also saved in the log files directory.

a. Click **Configure Database Passwords** to display the Password Management dialog box.

   The Password Management dialog box enables you to change the default password for a user after you upgrade the database. For security reasons, all users are locked except for the following users:

   – `SYS`

   – `SYSTEM`

   If you have enabled Local Management with Enterprise Manager, then the `SYSMAN` and `DBSNMP` accounts are also unlocked. These accounts provide Enterprise Manager with access to the database so it can gather monitoring data and so you can perform administration tasks with Enterprise Manager.

   If you have enabled Central Management with Enterprise Manager, then the `DBSNMP` account is unlocked, as well as the `SYS` and `SYSTEM` user accounts.

   Note: To prevent unauthorized use of the database, Oracle recommends that you change all user passwords immediately after you upgrade your database.

   If the default Oracle Database 11*g* Release 2 (11.2) security settings are in place, then passwords must be at least eight characters, and passwords such as `welcome` and `oracle` are not allowed. See *Oracle Database Security Guide* for more information.

b. Click **Restore Database** if you are not satisfied with the upgrade results.

   Depending on the method you used to back up your database, the Restore operation performs one of two tasks:

   – If you used DBUA to back up your database, then clicking Restore restores the original database and the original database settings from the backup.

   – If you used your own backup procedure to back up the database, then clicking Restore restores only the original database settings. To restore the database itself, you must restore the backup you created with your own backup utilities.

If you are satisfied with the upgrade results, then click **Exit** to quit DBUA and use your newly upgraded database.

a. DBUA modifies the `SID_DESC` entry for the upgraded database in the Oracle Database `listener.ora` file in one of the following ways:

   A simple case: Suppose the old `listener.ora` has the following SID_DESC entry:

```
...
   (SID_DESC =
      (SID_NAME = ORCL)
   )
```

```
...
```

If the database name is SAL, the domain name is COM, and the Oracle home is `/oracle/product/9.2`, then the assistant adds the following entry:

```
...
   (SID_DESC =
       (GLOBAL_DBNAME = sal.com)
          (ORACLE_HOME = /oracle/product/9.2)
          (SID_NAME = SAL)
    )
...
```

A more complicated case: Suppose the old `listener.ora` has the following SID_DESC entry:

```
...
   (SID_DESC =
       (GLOBAL_DBNAME = an_entry)
       (SID_NAME = ORCL)
    )
...
```

If *an_entry* does not match the `GLOBAL_DBNAME` of the migrated database, and if the database name is SAL, the domain name is COM, and the Oracle home is `/oracle/product/9.2`, then the assistant adds the following entry:

```
...
   (SID_DESC =
       (GLOBAL_DBNAME = sal.com)
          (ORACLE_HOME = /oracle/product/9.2)
          (SID_NAME = SAL)
    )
...
```

This entry is the same as the entry in the simple case, but DBUA also adds the entry *an_entry* to the `SERVICE_NAMES` parameter in the initialization parameter file. Therefore, DBUA changes the `SERVICE_NAMES` parameter to the following:

```
SERVICE_NAMES = sal.com, an_entry
```

b. DBUA removes the entry of the upgraded database from the old `listener.ora` file.

c. DBUA reloads the `listener.ora` file in both the old and new Oracle Database environments.

12. >>The following steps were deleted, either because the screens no longer appear or because they have been replaced with rewritten steps covering the same screens, as per IOUG presentation by Ravi Pattabhi 3/26/07. CBF 3/30/07

Specify the attributes for the `SYSAUX` tablespace, which is added automatically to all new Oracle Database 11*g* Release 1 (11.1) databases you create.

> **See Also:** *Oracle Database Administrator's Guide* for more information about the `SYSAUX` tablespace

Many of the attributes of the `SYSAUX` tablespace are set automatically and cannot be modified. For example, the `SYSAUX` tablespace is set to use Automatic

Segment-Space Management. However, you can specify the location of the data file, the default size of the SYSAUX tablespace, and its autoextend attributes.

> **Note:** If you specify an existing datafile for the SYSAUX tablespace, then you must select Reuse Existing File Name. Otherwise, DBUA alerts you to the fact that the file currently exists.

Click **Next**.

The Recompiling Invalid Objects screen appears.

13. Secure your database with passwords for the Enterprise Manager accounts. You can set a single password, which is applied to each of the listed Enterprise Manager user accounts, or enhance the security of the accounts by providing unique passwords for each user.

    Click **Next**.

    The Recovery Configuration screen appears.

14. Specify a flash recovery area and enable archiving. When you are managing your database, it is important to configure the database so you can recover your data in the event of a system failure.

    The Flash Recovery Area can be used to recover data that would otherwise be lost during a system failure. This location is also used by Enterprise Manager if you enabled local management and daily backups on the Management Options screen.

    Click **Next**.

    The Network Configuration screen appears.

15. At the Upgrade Operations page, do one of the following:

    - Select the option to upgrade only the Oracle ASM instance
    - Select the option to upgrade the database

      If you choose to upgrade the database, and the database is using Oracle ASM, then DBUA asks if you want to upgrade the Oracle ASM instance along with the database. Oracle recommends that you upgrade the database and Oracle ASM in separate DBUA sessions, in separate Oracle homes.

    Click **Next**.

    The Selecting a Database Instance screen appears.

16. At the **Changes in Default Behavior** screen, DBUA displays some changes in the behavior of Oracle Database 11*g* Release 1 (11.1) from that of previous releases. In some cases the default values of some initialization parameters have changed. In other cases some new behavior/requirement has been introduced that might affect current scripts or applications.

17. Complete the procedures described in Chapter 4, "Post-Upgrade Tasks for Oracle Database".

Optionally Performing an In-Place Upgrade (Into the Same Oracle Home)

Oracle recommends when upgrading to Oracle Database 11*g* R2 (11.2) that you perform an out-of-place mode upgrade, which installs the new software into a new Oracle home. This means that you provide a different Oracle home location for the new upgrade. However, there may be cases where you need to perform an in-place upgrade, which maintains the same Oracle home. This section provides the information on how to keep the existing Oracle home.

The considerations and options for performing in-place upgrades are provided in these topics:

- Known Issue When Starting an In-Place Upgrade
- Performing an In-Place Upgrade for Single-Instance Oracle Database
- Performing an In-Place Upgrade for an Oracle RAC Database
- In-Place Upgrade on Windows

Known Issue When Starting an In-Place Upgrade

When performing an in-place upgrade, which uses the same Oracle home location, an error messages appears stating that the installer detects Oracle Database software in the location that you specified.

Message: The installer has detected that the software location you have specified contains Oracle Database software release 11.2.0.1. Oracle recommends that when upgrading, you perform an out-of-place installation of the software into a new Oracle home and then upgrade the database using the new software binaries.

Cause: The installer has detected that the software location you have specified contains Oracle Database software release 11.2.0.1.

Action: Either perform an in-place upgrade as described in this section, or perform an out-of-place upgrade, into a new Oracle home, as described in "Installing the New Oracle Database Software for the Upgrade".

Performing an In-Place Upgrade for Single-Instance Oracle Database

If you must perform an in-place upgrade, then the procedure is to back up the current installation, move it to a different location, and then install the new software in the old location, as described in the following procedure.

> **WARNING: Oracle strongly recommends that you do not upgrade Oracle Database to release 11.2.0.2 or 11.2.0.3 using the existing Oracle home. This procedure is provided only if for some reason you have no alternative but to use the existing Oracle home.**

To perform an in-place upgrade for a single-instance database

1. Obtain the Oracle Database release 11.2.0.3 software from My Oracle Support at `https://support.oracle.com/`.
2. Back up the configuration data by backing up the following directories:
   - `ORACLE_HOME/dbs`
   - `ORACLE_HOME/network/admin`
   - `ORACLE_HOME/owb/bin/admin`
   - `ORACLE_HOME/`*`hostname_dbname`*
   - `ORACLE_HOME/oc4j/j2ee/OC4J_DBConsole_`*`hostname_dbname`*
3. Depending on which release you are upgrading from, in the software location for the release 11.2.0.1 or 11.2.0.2 software, *detach* the `ORACLE_HOME` with the following command:

```
ORACLE_HOME/oui/bin/runInstaller -detachHome ORACLE_HOME=11.2.x.x.x software
location
```

4. Rename or move this `ORACLE_HOME` software directory to a temporary name:

   ```
   mv ORACLE_HOME ORACLE_HOME.backup
   ```

5. Start OUI and select release 11.2.0.3. **Software Only**.
6. For **Location**, select the same location as used for release 11.2.0.1 or 11.2.0.2 (depending on which release you are upgrading from).
7. After the installation completes, restore the configuration data for the old Oracle home. Restore the backed up configuration data files (from the backups you made of `/dbs` and `network/admin`), and also restore the following directories: `/`*`hostname_dbname`*, `/owb/bin/admin`, and `/oc4j/j2ee/OC4J_DBConsole_`*`hostname_dbname`*. Specify the actual name for *`hostname_dbname`* where *hostname* is the actual hostname and *dbname* is the actual database name.
8. Run DBUA from `ORACLE_HOME/bin directory` and select the 11.2.0.1.0 or 11.2.0.2.0 database instance to perform the upgrade to 11.2.0.3.0.

   See Also:

   - *Oracle Database Installation Guide* for your operating system
   - *Oracle Database Installation Guide for Linux*
   -
   - *Oracle Universal Installer and OPatch User's Guide for Windows and UNIX*

Performing an In-Place Upgrade for an Oracle RAC Database

Performing an in-place upgrade of an Oracle RAC database is similar to the procedure described in "Performing an In-Place Upgrade for Single-Instance Oracle Database" on page 39. You must back up the current Oracle home and installation, move it to a different location, and then install the new software in the old location, as described in the following steps.

WARNING: Oracle strongly recommends that you do not upgrade Oracle Database to release 11.2.0.2 or 11.2.0.3 using the existing Oracle home. This procedure is provided only if for some reason you have no alternative but to use the existing Oracle home.

To perform an in-place upgrade for Oracle RAC Database instances

1. Back up the configuration data by backing up the following directories on all cluster nodes:

   - `ORACLE_HOME/dbs`
   - `ORACLE_HOME/network/admin`
   - `ORACLE_HOME/owb/bin/admin`
   - `ORACLE_HOME/`*`hostname_dbname`*
   - `ORACLE_HOME/oc4j/j2ee/OC4J_DBConsole_`*`hostname_dbname`*

2. Run the following command on each of the nodes to detach the old Oracle RAC `ORACLE_HOME`. For example:

```
ORACLE_HOME/oui/bin/runInstaller -detachHome ORACLE_HOME=11.2.0.1.0 software
location
```

3. Rename this Oracle RAC `ORACLE_HOME` directory on all the nodes to a temporary name.
4. Install release 11.2.0.3 **Software Only** on all nodes.
5. On all nodes, restore into the 11.2.0.3 `ORACLE_HOME` software location the backed up configuration data files (from the backups you made of `/dbs` and `network/admin`), and also restore the following directories: `/`*`hostname_dbname`*, `/owb/bin/admin`, and `/oc4j/j2ee/OC4J_DBConsole_`*`hostname_dbname`*. Specify the actual name for *`hostname_dbname`*.
6. Run DBUA from the 11.2.0.3 `ORACLE_HOME/bin` directory on the local node and select the 11.2.0.1.0 Oracle RAC database instance to upgrade it to release 11.2.0.3.0.

In-Place Upgrade on Windows

Oracle recommends that you perform the out-of-place upgrade into a new Oracle home. This procedure is provided only if for some reason you must use the old Oracle home.

The following topics discuss how to perform an upgrade into an existing Oracle home on Windows platforms:

- Considerations for In-Place Upgrade of Oracle Database on Windows Platforms
- Performing an In-Place Upgrade of Oracle Database on Windows

Important:

- Oracle recommends that you perform the out-of-place upgrade into a new Oracle home, which is described in "Upgrading with Database Upgrade Assistant on Linux, UNIX, and Windows" on page 24
- The installer on Windows is `setup.exe` instead of `runInstaller`

Considerations for In-Place Upgrade of Oracle Database on Windows Platforms

On Windows, you must shut down all services that invoke or lock Oracle files. To do this, set the services to **disable** in Windows service manager. A reboot may be required after you disable some services in order to free their associated locked objects. On Windows platforms, `ORADIM` manages Windows Services (operating system processes). DBCA manages Oracle database instances and database services.

See Also: *Oracle Database Platform Guide for Microsoft Windows*

Performing an In-Place Upgrade of Oracle Database on Windows

This procedure is provided only if you have no alternative but to use the old Oracle home when upgrading to the new Oracle Database 11*g* Release 2 (11.2) release. This is not the perferred method for upgrading Oracle Database to release 11.2. If you must perform an in-place upgrade, then you must perform the steps listed in this section. The procedure describes how to backup the current installation, move the current installation to a new location, and then install the new software into the old location.

WARNING: Oracle strongly recommends that you do not upgrade Oracle Database to release 11.2.0.2 or 11.2.0.3 using the existing Oracle home. This procedure is provided only if for some reason you have no alternative but to use the existing Oracle home.

To perform an in-place upgrade on Windows

1. Back up the configuration data by backing up the following directories on the single node or all cluster nodes:
   - `ORACLE_HOME/dbs`
   - `ORACLE_HOME/database`
   - `ORACLE_HOME/network/admin`
   - `ORACLE_HOME/owb/bin/admin`
   - `ORACLE_HOME/`*`hostname_dbname`*
   - `ORACLE_HOME/oc4j/j2ee/OC4J_DBConsole_`*`hostname_dbname`*
2. Skip this step if you are upgrading a single instance database. If you are upgrading an Oracle RAC database, then installation of the software for Oracle Database 11*g* Release 2 (11.2) requires that you upgrade Oracle Clusterware first. See "Considerations for Upgrading Oracle Clusterware and Oracle ASM Instances" on page 3
3. Shut down the instances running on `ORACLE_HOME`:

   ```
   SQL> SHUTDOWN IMMEDIATE
   ```

   Note: On an Oracle RAC Database, you must shut down the database instances on all nodes, and stop all interfering services on all nodes in the cluster.

4. On all instances, stop the Oracle services that may interfere with this Oracle home. For example:

   ```
   C:\oracle\bin\oradim -shutdown
   OracleService%ORACLE_SID%
   ```

   where *`OracleService`* is the name of the service, and *`ORACLE_SID`* is the SID of the database instance.

   Oracle services that may need to be stopped include:
   - the TNSListener
   - HTTP server
   - Oracle Management Server
   - OLAP Agent
   - Intelligent Agent

See Also: My Oracle Support at `https://support.oracle.com`:

- Note ID 1291682.1, "In-Place Upgrade to 11.2.0.2 on Windows Platform" for information on 11.2.0.2 and 11.2.0.3
- Note ID 294350.1, "Write Errors, or Files In Use, or NoServicesForProcessException During Patch Installation" for information on files that are locked by interfering processes
- *Oracle Database Platform Guide for Microsoft Windows* for information on using ORADIM to administer a database instance

---

Note: The list of services to stop is not inclusive. There may be Windows-specific and vendor-specific services running that need to be stopped.

---

5. Run the following command on each of the nodes to detach this Oracle RAC `ORACLE_HOME`:

   ```
   ORACLE_HOME/oui/bin/setup -detachHome ORACLE_HOME=this software location
   ```

6. Rename this Oracle RAC `ORACLE_HOME` directory on all the nodes to a temporary name.
7. Install release 11.2.0.3 **Software Only** on all nodes.
8. If needed, use ORADIM to create a new instance:

   ```
   C:\oracle\bin\oradim -NEW -SID ORACLE_SID
   ```

   where *`ORACLE_SID`* is the SID of the database instance. This will enable the instance to be listed with DBUA for upgrading.

   See Also: *Oracle Database Platform Guide for Microsoft Windows* for information on creating an instance with ORADIM

9. On all nodes, restore into this `ORACLE_HOME` software location the backed up configuration data files (from the backups you made of `/dbs`, `/database`, and `network/admin`), and also restore the following directories: `/`*`hostname_dbname`*, `/owb/bin/admin`, and `/oc4j/j2ee/OC4J_DBConsole_`*`hostname_dbname`*. Specify the actual name for *`hostname_dbname`*.
10. Run DBUA from this `ORACLE_HOME/bin` directory on the local node. If you are upgrading an Oracle RAC database, then select the 11.2.0.1.0 or 11.2.0.2.0 Oracle RAC database instance to upgrade to release 11.2.0.3.0.

    See Also: *Oracle Database Platform Guide for Microsoft Windows* for information on postinstallation tasks for the database

Using DBUA in Silent Mode to Upgrade Oracle Database

When invoked with the `-silent` command line option, DBUA operates in silent mode. In silent mode, DBUA does not present a user interface. It also writes any messages (including information, errors, and warnings) to a log file in *ORACLE_HOME*`/cfgtoollogs/dbua/SID/upgrade`*n*, where *n* is the number of upgrades that DBUA has performed as of this upgrade.

To upgrade a database with DBUA in silent mode

- Issue the following command

  ```
  dbua -silent -sid ORCL &
  ```

where the database is named `ORCL` in this example.

Oracle DBUA Command Line Options for Silent Mode

Database Upgrade Assistant (DBUA) supports command line options when run in silent mode. Table 3–3 describes the various options and the corresponding parameters that are supported by DBUA.

> **Note:** If the default Oracle Database 11*g* Release 2 (11.2) security settings are in place, then passwords must be at least eight characters, and passwords such as `welcome` and `oracle` are not allowed. See *Oracle Database Security Guide* for more information.

Table 3–3 DBUA Command Line Options

Option	Description
`-silent`	Specifies that DBUA should operate in silent mode.
`-sid` *`SID`*	Specifies the system identifier (SID) of the database to upgrade
`-oracleHome` *`home_name`*	Specifies the Oracle Database home directory of the database to upgrade
`-oracleBase` *`base_name`*	Specifies the Oracle Database base directory of the database to upgrade
`-diagnosticDest` *`diagnostic_destination`*	Specifies the default location to store Oracle trace and diagnostic files. It replaces the initialization parameter settings for background dump destination and user dump destination from earlier releases.
`-sysDBAUserName` *`SYSDBA_user`*	Specifies a user with `SYSDBA` privileges.
`-sysDBAPassword` *`SYSDBA_pwd`*	Specifies the password for *SYSDBA_user*.
`-autoextendFiles`	Autoextends database files during the upgrade. Data files revert to their original autoextend settings after the upgrade.
`-newGlobalDbName` *`db_name`*	Specifies a new global database name. This option applies only if you are moving data files or upgrading an Oracle XE database.
`-newSid` *`new_SID`*	Specifies a new system identifier (SID) of the database to upgrade. This option applies only if you are moving data files or upgrading an Oracle XE database.
`-upgradeTimezone`	Upgrades the timezone file version for the database.
`-generateMapFile`	Applies only if you are moving data files or upgrading an Oracle XE database. If you specify this option, then DBUA only generates a database map file in the log location and then exits.

Table 3–3 (Cont.) DBUA Command Line Options

Option	Description
`-useASM`	Applies only if you are moving data files or upgrading an Oracle XE database. If the database to upgrade has an Oracle ASM instance, then this option tells DBUA to use it for the upgrade.
`-commonFileLocation` *`common_files`*	Specifies a common location to store database files. This option applies only if you are moving data files or upgrading an Oracle XE database.
`-omfLocation` *`omf_area`*	Specifies a database area for Oracle Managed Files. This option applies only if you are moving data files or upgrading an Oracle XE database.
`-databaseMapFile` *`map_file_name`*	Specifies the full name of the map file to map database files. This option applies only if you are moving data files or upgrading an Oracle XE database.
`-newRecoveryArea` *`recover_area`*	Specifies the recovery area for a database that is moved during upgrade. This option applies only if you are moving data files or upgrading an Oracle XE database.
`-newRecoveryAreaSize` *`recover_size`*	Specifies the recovery area size (MB) for a database that is moved during upgrade. This option applies only if you are moving data files or upgrading an Oracle XE database.
`-apexAdminPassword` *`apex_pwd`*	Specifies the password for the Application Express Administrator.
`-disableUpgradeScriptLogging`	Disables the detailed log generation for running SQL scripts during the upgrade process. This is enabled by default. To enable log generation, do not specify this option.
`-backupLocation` *`directory`*	Specifies a directory to back up your database before the upgrade starts
`-sysauxTablespace`	Creates a new `SYSAUX` tablespace
`-datafileName` *`name`*	Specifies the complete path of the data file for creating the new `SYSAUX` tablespace
`-datafileSize` *`size`*	Specifies the size of the data file for creating the new `SYSAUX` tablespace {500M, 10G}
`-datafileSizeNext` *`size`*	Specifies the next extent for the data file
`-datafileSizeMax` *`size`*	Specifies the maximum size of the data file
`-postUpgradeScripts` *`script`* `[,` *`script`* `] ...`	Specifies a comma-delimited list of SQL scripts. Specify complete path names. The scripts are executed at the end of the upgrade.
`-initParam` *`parameter=value`* `[,` *`parameter=value`* `] ...`	Specifies a comma-delimited list of initialization parameter values of the form *name=value*
`-disableArchiveLogMode`	Turns off archiving and flashback logging for the duration of the upgrade.
`-recompile_invalid_objects true|false`	When you specify `TRUE` for this option, DBUA recompiles all invalid PL/SQL modules immediately after the upgrade is performed.
`-degree_of_parallelism` *`cpu_number`*	Specifies the number of CPUs to be used for parallel recompilation.

Table 3–3 (Cont.) DBUA Command Line Options

Option	Description
`-emConfiguration {CENTRAL|LOCAL|ALL|NOBACKUP|NOEMAIL|NONE}`	Specifies Oracle Enterprise Manager management options: ■ `CENTRAL` - Database is centrally managed by Oracle Enterprise Manager. ■ `LOCAL` - Database is locally managed by Oracle Enterprise Manager. ■ `ALL` - This option configures Oracle Enterprise Manager Database Control, enables daily backups, and enables e-mail notifications ■ `NOBACKUP` - Automatic daily backups of the database are not enabled. ■ `NOEMAIL` - E-mail notifications are not enabled. ■ `NONE` - Database is not managed by Enterprise Manager.
`-dbsnmpPassword` *`password`*	Specifies the `DBSNMP` user password. This option applies only if you are configuring Oracle Enterprise Manager.
`-sysmanPassword` *`password`*	Specifies the `SYSMAN` user password. This option applies only if you are configuring Oracle Enterprise Manager.
`-asmPassword` *`password`*	Specifies the `SYS` password for the Oracle ASM instance. This option applies only if you are configuring Oracle Enterprise Manager.
`-hostUserName` *`hostname`*	Specifies the host user name for the Oracle Enterprise Manager backup job. This option applies only if you are configuring Oracle Enterprise Manager.
`-hostUserPassword` *`password`*	Specifies the host user password for the Oracle Enterprise Manager backup job. This option applies only if you are configuring Oracle Enterprise Manager.
`-backupSchedule` *`hh`*`:`*`mm`*	Specifies the daily backup schedule in the form *hh*:*mm* (hours and minutes). This option applies only if you are configuring Oracle Enterprise Manager.
`-smtpServer` *`server_name`*	Specifies the outgoing mail (SMTP) server for E-mail notifications. This option applies only if you are configuring Oracle Enterprise Manager.
`-emailAddress` *`address`*	Specifies the E-mail address for E-mail notifications. This option applies only if you are configuring Oracle Enterprise Manager.
`-centralAgent` *`location`*	Specifies the Oracle Enterprise Manager central agent location. This option applies only if you are configuring Oracle Enterprise Manager.
`-recoveryAreaDestination` *`directory`*	Specifies the destination directory for all recovery files. This option applies only if you are moving data files, upgrading an Oracle XE database, or configuring Oracle Enterprise Manager.

Table 3–3 (Cont.) DBUA Command Line Options

Option	Description
`[-localRacSid SID`	Specifies the local SID of the cluster database if the cluster database is not registered in the Oracle Cluster Registry
`-h | -help`	Displays help for DBUA

DBUA Command Line Syntax for Silent Mode

You can specify all valid options from the command line using the following syntax:

```
dbua [ -silent ] [ -sid SID ] [-oracleHome home_name] [-oracleBase base_name]
[-diagnosticDest diagnostic_destination]
[-sysDBAUserName SYSDBA_user] [-sysDBAPassword SYSDBA_pwd]
[-upgradeASM] [-autoextendFiles] [-newGlobalDbName db_name] [-newSid new_SID]
[-generateMapFile] [-useASM] [-commonFileLocation common_files]
[-omfLocation omf_area] [-databaseMapFile map_file_name]
[-newRecoveryArea recover_area] [-newRecoveryAreaSize recover_size]
[-apexAdminPassword apex_pwd] [-disableUpgradeScriptLogging ]
[-backupLocation directory]
[-sysauxTablespace -datafileName name -datafileSize size
  -datafileSizeNext size -datafileSizeMax size]
[-postUpgradeScripts script [, script ] ... ]
[-initParam parameter=value [, parameter=value ] ... ]
[-disableArchiveLogMode] [-recompile_invalid_objects true | false]
[-degree_of_parallelism cpu_number]
[-emConfiguration {CENTRAL|LOCAL|ALL|NOBACKUP|NOEMAIL|NONE}
  -dbsnmpPassword password -sysmanPassword password -asmPassword password
  -hostUserName hostname -hostUserPassword password -backupSchedule hh:mm
  [-smtpServer server_name -emailAddress address]
  [-centralAgent location] [-localRacSid SID]]
[-recoveryAreaDestination directory] [-h|-help]
```

Upgrading Oracle Database Manually

There may be cases where you need to upgrade Oracle Database manually instead of using DBUA. After installing the new Oracle Database software as described in "Installing the New Oracle Database Software for the Upgrade" on page 8, backing up your database, and preparing the new Oracle home, you are ready to proceed with a manual upgrade. The procedure for manually upgrading your database assumes that you have previously run the Pre-Upgrade Information Tool as described in "Using the Pre-Upgrade Information Tool" on page 3-10.

The following topics guide you through the process of performing a manual upgrade.

- Backing Up Oracle Database for a Manual Upgrade
- Preparing the New Oracle Home for a Manual Upgrade
- Manually Upgrading Oracle Database

Backing Up Oracle Database for a Manual Upgrade

After running the Pre-Upgrade Information Tool and cleanly shutting down the instance, Oracle recommends that you back up the database as described in this section. If you encounter problems with the upgrade and wish to abandon the upgrade completely, then you must restore the database from this backup. Therefore, Oracle recommends that you back up your database now as a precaution.

To perform a full backup of the database to be upgraded

1. Sign on to RMAN:

   ```
   rman "target / nocatalog"
   ```

2. Issue the following RMAN commands:

   ```
   RUN
   {
       ALLOCATE CHANNEL chan_name TYPE DISK;
       BACKUP DATABASE FORMAT 'some_backup_directory%U' TAG before_upgrade;
       BACKUP CURRENT CONTROLFILE FORMAT 'controlfile location and name';
   }
   ```

   > **See Also:** *Oracle Database Backup and Recovery User's Guide*

Preparing the New Oracle Home for a Manual Upgrade

After backing up the database to be upgraded, prepare the new Oracle home in a new location. Do this for any release of Oracle Database for which you are upgrading, whether the database is release 11.2 or earlier. Starting with Oracle Database 11*g* Release 2 (11.2.0.3), you install the Oracle Grid Infrastructure and Oracle Database software into a new Oracle home instead of applying a patch set to the existing Oracle home.

> **See Also:** "Backing Up Oracle Database for a Manual Upgrade" on page 47

To prepare the new Oracle home

1. Copy configuration files from the Oracle home of the database being upgraded to the new Oracle Database 11*g* Release 2 (11.2) Oracle home:

   a. If your parameter file resides within the old environment's Oracle home, then copy it to the new Oracle home. By default, Oracle looks for the parameter file in the `ORACLE_HOME/dbs` directory on Linux or UNIX platforms and in the `ORACLE_HOME\database` directory on Windows operating systems. The parameter file can reside anywhere you wish, but it should not reside in the old environment's Oracle home after you upgrade to Oracle Database 11*g* Release 2 (11.2).

   > **Note:** It might be necessary to create a text initialization parameter file (`PFILE`) from the server parameter file (`SPFILE`) so that you can edit the initialization parameters.

   b. If your parameter file resides within an Oracle ASM instance, then back up the parameter file using the following command:

   ```
   CREATE pfile FROM spfile;
   ```

   If you must downgrade the database and your `SPFILE` resided within Oracle ASM, then the parameter file must be restored before the downgrade.

   c. If your parameter file is a text-based initialization parameter file with either an `IFILE` (include file) or a `SPFILE` (server parameter file) entry, and the file specified in the `IFILE` or `SPFILE` entry resides within the old environment's Oracle home, then copy the file specified by the `IFILE` or `SPFILE` entry to the new Oracle home. The file specified in the `IFILE` or `SPFILE` entry contains additional initialization parameters.

d. If you have a password file that resides within the old environment's Oracle home, then move or copy the password file to the Oracle Database 11*g* Release 2 (11.2) Oracle home.

   The name and location of the password file are operating system-specific. On Linux or UNIX platforms, the default password file is `orapw`*`sid`*, located in the `ORACLE_HOME/dbs` directory. On Windows operating systems, the default password file is `pwd`*`sid`*`.ora`, located in the `ORACLE_HOME\database` directory. In both cases, *`sid`* is your Oracle instance ID.

e. If you are upgrading a cluster database and your `init`*`db_name`*`.ora` file resides within the old environment's Oracle home, then move or copy the `init`*`db_name`*`.ora` file to the new Oracle home.

   Note: If you are upgrading a cluster database, then perform this step on all nodes in which this cluster database has instances configured.

2. If you are upgrading an Oracle Enterprise Edition database, you must copy the Enterprise Manager Database Console directory from the old Oracle Home into the new Oracle Home. Note that DBUA automatically copies and upgrades the Enterprise Manager Database Console directory.

3. Adjust your parameter file in Oracle Database 11*g* Release 2 (11.2) by completing the following steps:

   a. Remove obsolete initialization parameters and adjust deprecated initialization parameters. Certain parameters are obsolete in Oracle Database 11*g* Release 2 (11.2), while other parameters have become deprecated. Remove all obsolete parameters from any parameter file that starts an Oracle Database 11*g* Release 2 (11.2) instance. Obsolete parameters might cause errors in Oracle Database 11*g* Release 2 (11.2). 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.

      See Also: Appendix A, "Behavior Changes After Upgrading Oracle Database" for a list of initialization parameters that have been deprecated or have become obsolete, and for information on initialization parameters that have changed in ways that raise compatibility issues between different releases of Oracle Database software

   b. Make sure the `COMPATIBLE` initialization parameter is properly set for Oracle Database 11*g* Release 2 (11.2). The Pre-Upgrade Information Tool displays a warning in the Database section if `COMPATIBLE` is not properly set.

   c. Adjust the values of the initialization parameters to at least the minimum values indicated by the Pre-Upgrade Information Tool.

   d. Make sure all path names in the parameter file are fully specified. You should not have relative path names in the parameter file.

e. If the parameter file contains an IFILE entry, then change the IFILE entry in the parameter file to point to the new location of the include file that you specified in Step 1. c. Then, edit the file specified in the IFILE entry in the same way that you edited the parameter file in Step a through Step d.

f. If you are upgrading a cluster database, then modify the init*db_name*.ora file in the same way that you modified the parameter file.

Make sure you save all of the files you modified after making these adjustments.

> **Note:** If you are upgrading a cluster database, then perform this step on all nodes in which this cluster database has instances configured.

4. If you are upgrading a cluster database, then set the CLUSTER_DATABASE initialization parameter to false. After the upgrade, you must set this initialization parameter back to TRUE.

Manually Upgrading Oracle Database

After installing the new Oracle Database software as described in "Installing the New Oracle Database Software for the Upgrade" on page 8 and preparing the new Oracle home as described in "Preparing the New Oracle Home for a Manual Upgrade" on page 48, you are ready to proceed with the manual upgrade.

To manually upgrade the database

1. Shut down the instance:

```
SQL> SHUTDOWN IMMEDIATE
```

2. If your operating system is Windows, then complete the following steps:

   a. Stop the OracleService*SID* Oracle service of the database you are upgrading, where *SID* is the instance name. For example, if your *SID* is ORCL, then enter the following at a command prompt:

   ```
   C:\> NET STOP OracleServiceORCL
   ```

   b. Delete the Oracle service at a command prompt using ORADIM.

   If your *SID* is ORCL, then enter the following command:

   ```
   C:\> ORADIM -DELETE -SID ORCL
   ```

   c. Create the Oracle Database 11*g* Release 2 (11.2) service at a command prompt using the ORADIM command of the new Oracle Database release:

   ```
   C:\> ORADIM -NEW -SID SID -SYSPWD PASSWORD -MAXUSERS USERS
        -STARTMODE AUTO -PFILE ORACLE_HOME\DATABASE\INITSID.ORA
   ```

   This syntax includes the following variables:

Variable	Description
SID	The same SID name as the SID of the database you are upgrading.

Variable	Description
PASSWORD	The password for the new Oracle Database 11*g* Release 2 (11.2) database instance. This is the password for the user connected with SYSDBA privileges. The -SYSPWD option is not required. If you do not specify it, then operating system authentication is used, and no password is required. If the default Oracle Database 11*g* Release 2 (11.2) security settings are in place, then passwords must be at least eight characters, and passwords such as welcome and oracle are not allowed. See *Oracle Database Security Guide* for more information.
USERS	The maximum number of users who can be granted SYSDBA and SYSOPER privileges.
ORACLE_HOME	The Oracle Database 11*g* Release 2 (11.2) Oracle home directory. Ensure that you specify the full path name with the -PFILE option, including the drive letter of the Oracle home directory.

For example, if your SID is ORCL, your password (SYSPWD) is TWxy5791, the maximum number of users (MAXUSERS) is 10, and the ORACLE_HOME directory is C:\ORACLE\PRODUCT\11.2.0\DB, then enter the following command:

```
C:\> ORADIM -NEW -SID ORCL -SYSPWD TWxy5791 -MAXUSERS 10
-STARTMODE AUTO -PFILE C:\ORACLE\PRODUCT\11.2.0\DB\DATABASE\INITORCL.ORA
```

ORADIM writes a log file to the ORACLE_HOME\database directory.

3. If your operating system is Linux or UNIX, then perform the following checks:

   a. Your ORACLE_SID is set correctly

   b. The oratab file points to your Oracle Database 11*g* Release 2 (11.2) Oracle home

   c. The following environment variables point to the Oracle Database 11*g* Release 2 (11.2) directories:

      – ORACLE_HOME

      – PATH

   d. Any scripts that clients use to set the ORACLE_HOME value must point to the new Oracle home.

Note: If you are upgrading a cluster database, then perform these checks on all nodes on which this cluster database has instances configured.

See Also: *Oracle Database Installation Guide* for your operating system-specific Oracle Database procedures for information about setting other important environment variables on your operating system

4. If there is a different user and group for Grid Infrastructure and Oracle Database, then run the setasmgidwrap script, which is located in the GRID_HOME/bin directory, against the new ORACLE_HOME/bin/oracle directory with the -o option as shown in the following example:

```
$GRID_HOME/bin/setasmgidwrap o=ORACLE_HOME/bin/oracle
```

5. Log in to the system as the owner of the Oracle Database 11*g* Release 2 (11.2) Oracle home directory.
6. At a system prompt, change to the `ORACLE_HOME/rdbms/admin` directory.
7. Start SQL*Plus.
8. Connect to the database instance as a user with `SYSDBA` privileges.
9. Start the instance by issuing the following command:

   ```
   SQL> STARTUP UPGRADE
   ```

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

   You might be required to use the `PFILE` option to specify the location of your initialization parameter file.

   Once the database is started in upgrade mode, only queries on fixed views execute without errors until after the `catupgrd.sql` script is run. Before running `catupgrd.sql`, queries on any other view or the use of PL/SQL returns an error.

   The following are common errors that might occur when attempting to start the new Oracle Database 11*g* Release 2 (11.2) database. Some of these errors are written to the alert log and not to your session. If you receive any of these errors, then issue the `SHUTDOWN ABORT` command to shut down the database and correct the problem.

   - `ORA-00401: the value for parameter compatible is not supported by this release`

     The `COMPATIBLE` initialization parameter is set to a value less than `10.0.0`.

   - `ORA-39701: database must be mounted EXCLUSIVE for UPGRADE or DOWNGRADE`

     The `CLUSTER_DATABASE` initialization parameter is set to `TRUE` instead of `FALSE`.

   - `ORA-39700: database must be opened with UPGRADE option`

     The `STARTUP` command was issued without the `UPGRADE` keyword.

   - `ORA-00336: log file size xxxx blocks is less than minimum 8192 blocks`

     A redo log file size is less than 4 MB:

   If errors appear listing obsolete initialization parameters, then make a note of the obsolete initialization parameters and continue with the upgrade. Remove the obsolete initialization parameters the next time you shut down the database.

10. Perform this step only if you are upgrading from Oracle9*i* Release 2 (9.2). Otherwise, skip to the next step.

    Create a `SYSAUX` tablespace. In the new Oracle Database 11*g* release, the `SYSAUX` tablespace is used to consolidate data from a number of tablespaces that were separate in previous releases.

    The `SYSAUX` tablespace must be created with the following mandatory attributes:

- ONLINE
- PERMANENT
- READ WRITE
- EXTENT MANAGEMENT LOCAL
- SEGMENT SPACE MANAGEMENT AUTO

Table 3–4 can be used to determine an optimal size for the SYSAUX tablespace. Also, the Pre-Upgrade Information Tool provides an estimate of the minimum required size for the SYSAUX tablespace under the SYSAUX Tablespace section.

Table 3–4 Guidelines for Sizing the SYSAUX Tablespace

Factor	Small	Medium	Large
Number of CPUs	2	8	32
Number of concurrently active sessions	5	20	100
Number of user objects (tables and indexes)	500	5,000	50,000
Estimated SYSAUX size at steady state with default configuration	500 MB	2 GB	5 GB

The following SQL statement would create a 500 MB SYSAUX tablespace for the database:

```
SQL> CREATE TABLESPACE sysaux DATAFILE 'sysaux01.dbf'
         SIZE 500M REUSE
         EXTENT MANAGEMENT LOCAL
         SEGMENT SPACE MANAGEMENT AUTO
         ONLINE;
```

> **See Also:** *Oracle Database Administrator's Guide* for more information about the SYSAUX tablespace

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

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

12. 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 11*g* Release 2 (11.2) environment.

The upgrade script creates and alters certain data dictionary tables. It also upgrades or installs the following database components in the new Oracle Database 11*g* Release 2 (11.2) database:

- Oracle Database Catalog Views
- Oracle Database Packages and Types
- JServer JAVA Virtual Machine
- Oracle Database Java Packages
- Oracle XDK
- Oracle Real Application Clusters

- Oracle Workspace Manager
- Oracle Multimedia
- Oracle XML Database
- OLAP Analytic Workspace
- Oracle OLAP API
- OLAP Catalog
- Oracle Text
- Spatial
- Oracle Data Mining
- Oracle Label Security
- Messaging Gateway
- Oracle Expression Filter
- Oracle Rules Manager
- Oracle Enterprise Manager Repository
- Oracle Database Vault
- Oracle Application Express

Note: If you did not run the Pre-Upgrade Information Tool, the `catupgrd.sql` script terminates with one of the following errors:

```
ORA-00942: table or view does not exist
ORA-00904: "TZ_VERSION": invalid identifier
ORA-01722: invalid number
```

If you receive any of these errors, issue the `SHUTDOWN ABORT` statement, revert to the original Oracle home directory, and run the Pre-Upgrade Information Tool (`utlu112i.sql`) as described in "Using the Pre-Upgrade Information Tool" on page 3-10.

13. 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 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.

14. If you encountered a message listing obsolete initialization parameters when you started the database in Step 9, then remove the obsolete initialization parameters from the parameter file before restarting. If necessary, convert the `SPFILE` to a `PFILE` so you can edit the file to delete parameters. See the chapter about managing initialization parameters using a server parameter file in *Oracle Database Administrator's Guide*.

15. Run `utlu112s.sql`, the Post-Upgrade Status Tool, which provides a summary of the upgrade at the end of the spool log. Starting with release 11.2.0.3, ou can run `utlu112s.sql` any time after completing the upgrade. See "About the Post-Upgrade Status Tool" on page 56 for more information.

Run `utlu112s.sql` to display the results of the upgrade as follows:

```
SQL> @utlu112s.sql
```

If the Post-Upgrade Status Tool returns errors or shows components that are not `VALID` or not the most recent release, then see "Troubleshooting the Upgrade of Oracle Database" on page 57 for more information.

16. Run `catuppst.sql` to perform upgrade actions that do not require the database to be in `UPGRADE` mode:

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

This may generate the following informational messages:

```
Generating apply and rollback scripts...
Check the following file for errors:
Apply script: .*
Rollback script: .*
Executing script file...
Updating registry...
Check the following log file for errors:

Generating apply and rollback scripts...
Check the following file for errors:
.../cfgtoollogs/catbundle/catbundle_PSU_*.log
Apply script:
.../rdbms/admin/catbundle_PSU_*_APPLY.sql
Rollback script:
.../rdbms/admin/catbundle_PSU_*_ROLLBACK.sql
Executing script file...

Updating registry...
Check the following log file for errors:
.../cfgtoollogs/catbundle/catbundle_PSU_*.log
```

17. Run `utlrp.sql` to recompile any remaining stored PL/SQL and Java code.

```
SQL> @utlrp.sql
```

Verify that all expected packages and classes are valid:

```
SQL> SELECT count(*) FROM dba_invalid_objects;
SQL> SELECT distinct object_name FROM dba_invalid_objects;
```

> **Note:** If the pre-upgrade information tool detected `INVALID` objects and populated the `registry$sys_inv_objs` and `registry$nonsys_inv_objs` tables, then execute *ORACLE_HOME*/`rdbms/admin/utluiobj.sql` to display only those objects which are newly invalid because of the upgrade process. The `utluiobj.sql` script only displays objects that are now `INVALID` but which were `VALID` before the upgrade.

18. Exit SQL*Plus.

19. If you are upgrading a cluster database from releases 10.2, 11.1, or 11.2, then upgrade the database configuration in Oracle Clusterware using the following command:

```
$ srvctl upgrade database -d db-unique-name -o oraclehome
```

where *db-unique-name* is the database name assigned to it (not the instance name), and *oraclehome* is the Oracle home location in which the database is being upgraded.

Your database is now upgraded to the new Oracle Database 11*g* release. You are ready to complete the procedures described in Chapter 4, "Post-Upgrade Tasks for Oracle Database".

Note: Oracle Warehouse Builder components are not upgraded as part of the catupgrd.sql script and, therefore, its version will not be updated during the upgrade process.

Caution: If you retain the old Oracle software, then never start the upgraded database with the old software. Only start the database with the executables in the new Oracle Database installation. Also, before you remove the old Oracle environment, make sure you relocate any data files in that environment to the new Oracle Database environment. See the *Oracle Database Administrator's Guide* for information about relocating data files.

About the Post-Upgrade Status Tool

The Post-Upgrade Status Tool, which is the utlu112s.sql script, 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. The utlu112s.sql script can be run any time after you run catupgrd.sql, but *not* after running utlrp.sql.

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

```
Oracle Database 11.2 Post-Upgrade Status Tool           07-12-2011 12:36:25
.
Component                               Current      Version     Elapsed Time
Name                                    Status       Number      HH:MM:SS
.
Oracle Server
.                                         VALID      11.2.0.3.0  00:04:47
JServer JAVA Virtual Machine
.                                         VALID      11.2.0.3.0  00:01:05
Oracle Workspace Manager
.                                         VALID      11.2.0.3.0  00:00:18
OLAP Analytic Workspace
.                                         VALID      11.2.0.3.0  00:00:07
OLAP Catalog
.                                         VALID      11.2.0.3.0  00:00:21
Oracle OLAP API
.                                         VALID      11.2.0.3.0  00:00:12
Oracle Label Security
.                                         VALID      11.2.0.3.0  00:00:02
Oracle Enterprise Manager
.                                         VALID      11.2.0.3.0  00:00:50
Oracle XDK
.                                         VALID      11.2.0.3.0  00:00:19
Oracle Text
.                                         VALID      11.2.0.3.0  00:00:11
Oracle XML Database
```

```
.                                         VALID      11.2.0.3.0  00:01:04
Oracle Database Java Packages
.                                         VALID      11.2.0.3.0  00:00:07
Oracle Multimedia
.                                         VALID      11.2.0.3.0  00:01:25
Spatial
.                                         VALID      11.2.0.3.0  00:00:50
Oracle Expression Filter
.                                         VALID      11.2.0.3.0  00:00:04
Oracle Rules Manager
.                                         VALID      11.2.0.3.0  00:00:02
Oracle Application Express
.                                         VALID     3.2.1.00.10
Oracle Database Vault
.                                         VALID      11.2.0.3.0  00:00:02
Gathering Statistics
.                                                                00:00:36
Total Upgrade Time: 00:12:30
```

Troubleshooting the Upgrade of Oracle Database

Oracle provides troubleshooting tips and workarounds for problems and errors that may occur during the upgrade process for Oracle Database. Also, be sure to check *Oracle Database Readme*.

This section contains the following topics:

- Pre-Upgrade Oracle Home Removal on Oracle RAC
- Resource Limits and Oracle Database Upgrade
- COMPATIBLE Parameter Not Set
- Edition Session Startup Error
- Manual Workaround for ORA-01408
- Running the DBMS_DST Package After Upgrade Can Result in ORA-01822
- DBUA May Mark Invalid Components with an X Before Entire Upgrade is Done
- Understanding Component Status
- Rerunning the Upgrade for Oracle Database
- Cancelling the Upgrade for Oracle Database

Pre-Upgrade Oracle Home Removal on Oracle RAC

You may need to keep the old Oracle Home for a while after a successful upgrade. However, after you have tested the newly upgraded Oracle database, you will need to remove the old Oracle Home.

As the clusterware user on all nodes, enter the following SQL*Plus commands.

- Detach *OLD_HOME* (where *OLD_HOME* is your pre-upgrade Oracle Home:

  ```
  $OLD_HOME/oui/bin/runInstaller -detachHome -silent -local
  ```

- Confirm *OLD_HOME* is removed from central inventory:

  ```
  $OLD_HOME/OPatch/opatch lsinventory -all
  ```

- Remove files in *OLD_HOME* manually:

```
/bin/rm -rf $OLD_HOME
```

Resource Limits and Oracle Database Upgrade

If you run out of resources, such as disk space or memory, during the upgrade of Oracle Database, then increase the resource allocation. After increasing the resource allocation, you should perform a `SHUTDOWN ABORT` and restart the instance (in `UPGRADE` mode) before rerunning the `catupgrd.sql` script or restarting DBUA.

The resources that generally require increases for a new Oracle Database release are as follows:

- `SYSTEM` and `SYSAUX` tablespaces

  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
  ```

  To avoid these errors, set `AUTOEXTEND ON MAXSIZE UNLIMITED` for the `SYSTEM` and `SYSAUX` tablespaces.

- Shared memory

  You might require larger shared memory pool sizes in some cases. The error message indicates which shared memory initialization parameter must be increased.

  ```
  ORA-04031: unable to allocate string bytes of shared memory
  ("string","string","string","string")
  ```

  > **See Also:** *Oracle Database Reference* for information about shared memory initialization parameters.

- Rollback segments/undo tablespace

  If you are using rollback segments, then you must have a single large (100 MB) `PUBLIC` rollback segment online while the upgrade scripts are being run. Smaller public rollback segments should be taken offline during the upgrade. Typically you get the following error if your rollback segment size is insufficient:

  ```
  ORA-01562: failed to extend rollback segment number string
  ```

  If you are using an undo tablespace, then be sure it is at least 400 MB.

- Flash Recovery Area

  If you are using a Flash Recovery Area and it fills up during the upgrade, then the following error appears in the alert log, followed by suggestions for recovering from the problem:

  ```
  ORA-19815: WARNING: db_recovery_file_dest_size of string bytes is 98.99%
  used, and has string remaining bytes available.
  ```

Identify the root cause of the problem and take appropriate actions to proceed with the upgrade. To avoid issues during the upgrade, increase the amount of space available in your Flash Recovery Area before starting the upgrade.

COMPATIBLE Parameter Not Set

If you see error ORA-00723, "Initialization parameter COMPATIBLE must be explicitly set," when attempting to open a database in the new Oracle home in upgrade mode, then check that COMPATIBLE is set in the pfile or spfile.

> **See Also:** "Setting the COMPATIBLE Initialization Parameter" on page 4-14 for information about setting the COMPATIBLE initialization parameter

Edition Session Startup Error

If an upgrade script or command running in SQL*Plus set the EDITION parameter, then Oracle Database cannot start properly afterwards and error SP2-1540: "Oracle Database cannot startup in an Edition session" is thrown. To avoid this problem, after running catugrd.sql or any SQL*Plus session where this parameter is changed, exit the SQL*Plus session and restart the instance in a different session.

Manual Workaround for ORA-01408

The ORA-01408 error is a known problem with Oracle Application databases. The workaround addresses the following scenario:

```
SQL> Create Index system.repcat$_audit_column_f2_idx ON
2  system.repcat$_audit_column(base_sname,base_oname,base_conflict_type_id,
3  base_reference_name)
4  /
system.repcat$_audit_column(base_sname,base_oname,base_conflict_type_id,
                          *
ERROR at line 2:
ORA-01408: such column list already indexed
```

The workaround is to drop the REPCAT$_AUDIT_COLUMN_IDX1 index and rerun the upgrade as described in "Rerunning the Upgrade for Oracle Database" on page 3-60.

Running the DBMS_DST Package After Upgrade Can Result in ORA-01822

Running the DBMS_DST package after upgrading to Oracle Database 11*g* Release 2 (11.2) can result in the ORA-01882: time zone region not found error.

This error is returned if the user sets the time zone file version incorrectly, which results in the region IDs of several time zone regions being stored incorrectly in the database. For example:

```
ERROR at line 1:
@  ORA-01882: time zone region not found
@  ORA-06512: at "SYS.DBMS_DST", line 113
@  ORA-06512: at "SYS.DBMS_DST", line 1101
@  ORA-06512: at line 1
```

To fix this problem, update the time zone version as described in "About Warnings for TIMESTAMP WITH TIME ZONE Data Type" on page 18 and rerun the upgrade as described in "Rerunning the Upgrade for Oracle Database" on page 3-60.

Understanding Component Status

The Post-Upgrade Status Tool should report 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.

See Also: "About the Post-Upgrade Status Tool" on page 56

DBUA May Mark Invalid Components with an X Before Entire Upgrade is Done

DBUA may mark invalid components with an X on the Upgrade Progress window as the upgrade is progressing. An invalid component could be re-validated after a recompile is performed. Please check the final component status on the Upgrade Result page in DBUA after utlrp.sql recompiles have been run.

See Also: "Understanding Component Status" on page 60

Rerunning the Upgrade for Oracle Database

You can rerun the upgrade with the catupgrd.sql script as described in the following steps.

To rerun the upgrade

1. Shut down the database as follows:

```
SQL> SHUTDOWN IMMEDIATE
```

2. Restart the database in UPGRADE mode:

```
SQL> STARTUP UPGRADE
```

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

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

4. Rerun `catupgrd.sql`:

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

   > **Note:** You can rerun the `catupgrd.sql` script as many times as necessary. The first time you run the script, there should be no error messages returned. If you rerun the script, then the `ORA-00001` message is displayed. You can safely ignore this message.

5. Rerun `utlu112s.sql`:

   ```
   SQL> @utlu112s.sql
   ```

Cancelling the Upgrade for Oracle Database

If you completed the steps in "Backing Up Oracle Database for a Manual Upgrade" on page 47 to back up your database, then the easiest way to cancel the upgrade is to restore that backup as described in the following procedure.

To cancel the upgrade by restoring the previous backup

1. Log in to the system as the owner of the Oracle home directory of the previous release.
2. Sign on to RMAN:

   ```
   rman "target / nocatalog"
   ```

3. Issue the following RMAN commands:

   ```
   STARTUP NOMOUNT
   RUN
   {
       RESTORE CONTROLFILE FROM 'save_controlfile_location';
       ALTER DATABASE MOUNT;
       RESTORE DATABASE FROM TAG before_upgrade
       ALTER DATABASE OPEN RESETLOGS;
   }
   ```

About Upgrading an Oracle ASM Instance

The recommended practice for upgrading Oracle ASM is to upgrade an Oracle ASM instance with the Oracle Universal Installer (OUI) executable file that is located in the Oracle Grid Infrastructure home directory. OUI automatically defaults to upgrade mode when it detects an Oracle ASM instance at an earlier release level.

Oracle ASM Configuration Assistant enables you to upgrade an existing Oracle ASM instance to the current software level and upgrade an older Oracle ASM instance to the latest Oracle Grid Infrastructure home.

You can upgrade an Oracle ASM instance to an Oracle Restart 11*g* release 2 (11.2) configuration. The recommended practice is to upgrade an Oracle ASM instance with Oracle Universal Installer (OUI).

You can also perform a rolling upgrade to clustered Oracle ASM instances in environments running Oracle Database 11*g* or later.

See Also:

- *Oracle Database Storage Administrator's Guide* for more information about upgrading and Oracle ASM instance with OUI
- *Oracle Database Storage Administrator's Guide* for more information about upgrading an Oracle ASM instance with Oracle ASM Configuration Assistant
- *Oracle Database Storage Administrator's Guide* for more information about upgrading an Oracle ASM instance in an Oracle Restart configuration with OUI
- *Oracle Grid Infrastructure Installation Guide* for step-by-step instructions about performing a rolling upgrade of Oracle ASM

4

Post-Upgrade Tasks for Oracle Database

After performing the procedures for upgrading Oracle Database, you must complete required tasks and consider recommendations for the new release.

The chapter contains the following topics:

- Required Tasks to Complete After Upgrading Oracle Database
- Recommended Tasks to Complete After Upgrading Oracle Database
- Tasks to Complete Only After Manually Upgrading Oracle Database
- Required Tasks After Oracle Grid Infrastructure Upgrades
- Required Tasks After Oracle ASM Upgrades
- Recommended Tasks After Oracle ASM Upgrades

Required Tasks to Complete After Upgrading Oracle Database

After you upgrade Oracle Database, regardless of whether you perform the upgrade manually, or upgrade automatically by using Database Upgrade Assistant (DBUA), you must complete any required tasks that are specified for your environment. You must also consider important information about your environment. The following topics contain the required procedures and information:

- Setting Environment Variables on Linux and UNIX Systems After Manual Upgrades
- Upgrading the Recovery Catalog After Upgrading Oracle Database
- Upgrading the Time Zone File Version After Upgrading Oracle Database
- Upgrading Statistics Tables Created by the DBMS_STATS Package After Upgrading Oracle Database
- Upgrading Externally Authenticated SSL Users After Upgrading Oracle Database
- Installing Oracle Text Supplied Knowledge Bases After Upgrading Oracle Database
- Updating Your Oracle Application Express Configuration After Upgrading Oracle Database
- Configuring Fine-Grained Access to External Network Services After Upgrading Oracle Database
- Enabling Oracle Database Vault and Revoking the DV_PATCH_ADMIN Role After Upgrading Oracle Database

Setting Environment Variables on Linux and UNIX Systems After Manual Upgrades

If your operating system is Linux or UNIX, and if you performed a manual upgrade of Oracle Database, then you must ensure that certain environment variables point to the directories of the new Oracle Database release. Note that DBUA automatically makes necessary changes to environment variables. Additionally, if you are upgrading a cluster database, then perform these checks on all nodes on which the cluster database has instances configured.

Confirm that the following environment variables point to the directories of the new Oracle home:

- `ORACLE_HOME`
- `PATH`

See Also:

- *Oracle Database Administrator's Guide* for procedures
- Your operating system-specific Oracle Database installation documents for information about setting other important environment variables on your operating system

Setting oratab and Scripts to Point to the New Oracle Home After Upgrading Oracle Database

After you upgrade Oracle Database to the new release, you must ensure that your `oratab` file and any client scripts that set the value of `ORACLE_HOME` point to the new Oracle home that is created for the new Oracle Database 11*g* release. Although DBUA automatically points `oratab` to the new Oracle home, client scripts must be checked no matter which method you use to upgrade.

See Also: *Oracle Database Administrator's Guide*

Upgrading the Recovery Catalog After Upgrading Oracle Database

For complete information about upgrading the recovery catalog and the `UPGRADE CATALOG` command, see *Oracle Database Backup and Recovery User's Guide* for the topic that describes the procedures.

Upgrading the Time Zone File Version After Upgrading Oracle Database

If the Pre-Upgrade Information Tool instructed you to upgrade the time zone files after completing the database upgrade, then use the `DBMS_DST` PL/SQL package to upgrade the time zone file.

See Also:

- "About Warnings for TIMESTAMP WITH TIME ZONE Data Type" on page 3-18
- *Oracle Database Globalization Support Guide* and follow the procedure in "Steps to Upgrade Time Zone File and Timestamp with Time Zone Data"

Upgrading Statistics Tables Created by the DBMS_STATS Package After Upgrading Oracle Database

If you created statistics tables using the DBMS_STATS.CREATE_STAT_TABLE procedure, then upgrade these tables by running the following procedure:

```
EXECUTE DBMS_STATS.UPGRADE_STAT_TABLE('scott', 'stat_table');
```

In the example, SCOTT is the owner of the statistics table and STAT_TABLE is the name of the statistics table. Perform this procedure for each statistics table.

Upgrading Externally Authenticated SSL Users After Upgrading Oracle Database

If you are upgrading from Oracle9*i* Release 2 (9.2) or Oracle Database 10*g* Release 1 (10.1), and you are using externally authenticated SSL users, then you must run the SSL external users conversion (extusrupgrade) script to upgrade those users. The script has the following syntax:

```
ORACLE_HOME/rdbms/bin/extusrupgrade --dbconnectstring
<hostname:port_no:sid> --dbuser <db admin> --dbuserpassword
<password> -a
```

Note: If you are upgrading from Oracle Database 10*g* Release 2 (10.2) or higher, then you are not required to run this command.

See Also: for more information on the extusrupgrade script

Installing Oracle Text Supplied Knowledge Bases After Upgrading Oracle Database

The Oracle Text supplied knowledge bases are part of the companion Oracle Database 11*g* release products and are not immediately available after an upgrade to the new Oracle Database 11*g* release. Any Oracle Text features dependent on the supplied knowledge bases which were available before the upgrade do not function after the upgrade. To re-enable such features, you must install the Oracle Text supplied knowledge bases from the installation media.

After an upgrade, all user extensions to the Oracle Text supplied knowledge bases must be regenerated. These changes affect all databases installed in the given Oracle home.

See Also:

- *Oracle Text Application Developer's Guide* for information about Oracle Text-supplied knowledge bases
- The postinstallation tasks section of your platform-specific *Oracle Database Installation Guide* for companion products

Updating Your Oracle Application Express Configuration After Upgrading Oracle Database

If your database originally included Application Express Version 3.2 or higher, then there is no additional configuration necessary after upgrading to the new Oracle Database 11*g* release.

If your database was not an Oracle Express Edition (XE) database, but contained an earlier version of Application Express (HTML DB), then the latest version is automatically installed during the upgrade. You must complete a series of postinstallation steps to configure Application Express for use with the new Oracle Database 11*g* release.

> **See Also:** *Oracle Application Express Installation Guide* for postinstallation tasks for Application Express

If your database is an Oracle Express Edition (XE) database, then it contains an earlier version of Application Express, which is tailored for the XE environment. Review the OTN document describing the differences between Oracle XE and Oracle Application Express at the following URL:

http://www.oracle.com/technetwork/developer-tools/apex/overview/index.html

The database administration features available with the XE edition of Application Express are not available in version 3.2, but Oracle Enterprise Manager Database Control can, optionally, be installed to provide a graphical interface for database administration.

Configuring Fine-Grained Access to External Network Services After Upgrading Oracle Database

Oracle Database 11*g* includes fine-grained access control to the UTL_TCP, UTL_SMTP, UTL_MAIL, UTL_HTTP, or UTL_INADDR packages using Oracle XML DB. If you have applications that use these packages, then you must install Oracle XML DB if it is not installed. You must also configure network *access control lists* (ACLs) in the database before these packages can work as they did in earlier releases.

The following example first looks for any ACL currently assigned to *host_name*. If one is found, then the example grants *user_name* the CONNECT privilege in the ACL, only if that user does not have this privilege. If no ACL exists for *host_name*, then the example creates a new ACL called *ACL_name*, grants the CONNECT privilege to *user_name*, and assigns the ACL to *host_name*.

```
DECLARE
  acl_path  VARCHAR2(4000);
BEGIN
  SELECT acl INTO acl_path FROM dba_network_acls
   WHERE host = 'host_name' AND lower_port IS NULL AND upper_port IS NULL;
IF DBMS_NETWORK_ACL_ADMIN.CHECK_PRIVILEGE(acl_path,
                                         'user_name','connect') IS NULL THEN
    DBMS_NETWORK_ACL_ADMIN.ADD_PRIVILEGE(acl_path,
                                         'user_name', TRUE, 'connect');
END IF;
EXCEPTION
  WHEN no_data_found THEN
    DBMS_NETWORK_ACL_ADMIN.CREATE_ACL('ACL_name.xml',
      'ACL description', 'user_name', TRUE, 'connect');
    DBMS_NETWORK_ACL_ADMIN.ASSIGN_ACL('ACL_name.xml','host_name');
END;
COMMIT;
```

> **Note:** The transaction must be committed for the changes to take effect.

See Also: *Oracle Database Security Guide* for more complicated situations, such as connecting some users to host A and other users to host B

Enabling Oracle Database Vault and Revoking the DV_PATCH_ADMIN Role After Upgrading Oracle Database

If you use Oracle Database Vault, then you were instructed to disable it before upgrading your database. You must now:

- Enable Database Vault.
- Revoke the Database Vault `DV_PATCH_ADMIN` role for the `SYS` account.

See Also:

- "About Upgrading Databases That Use Oracle Database Vault" on page 3-8
- The appendix about "Disabling and Enabling Oracle Database Vault" in *Oracle Database Vault Administrator's Guide*
- The section about "Manually Deploying Oracle Database Vault Administrator" in *Oracle Database Vault Administrator's Guide*
- The appendix about "Post-installation Oracle Database Vault Procedures" in *Oracle Database Vault Administrator's Guide*

Recommended Tasks to Complete After Upgrading Oracle Database

Performing the following tasks is recommended, but not required, after you have upgraded your database. These tasks are recommended regardless of whether you performed the upgrade manually or by using DBUA:

- Recommended Tasks to Perform After All Database Upgrades
- Recommended Tasks After Upgrading an Oracle Database 10g Release 1 (10.1) Database
- Recommended Tasks After Upgrading an Oracle Express Edition Database
- Recommended Tasks After Upgrading an Oracle RAC Database

Recommended Tasks to Perform After All Database Upgrades

Performing the following tasks is recommended, but not required, after you have upgraded your database.

- Back Up the Database
- Reset Passwords to Enforce Case-Sensitivity
- Understand Changes with Oracle Grid Infrastructure
- Understand Oracle ASM and Oracle Grid Infrastructure Installation and Upgrade
- Add New Features as Appropriate
- Develop New Administrative Procedures as Needed
- Set Threshold Values for Tablespace Alerts
- Migrate From Rollback Segments to Automatic Undo Mode

- Configure Oracle Data Guard Broker
- Migrate Tables from the LONG Data Type to the LOB Data Type
- Test the Upgraded Production Database

Back Up the Database

Make sure you perform a full backup of the production database.

> **See Also:** *Oracle Database Backup and Recovery User's Guide* for details about backing up a database

Reset Passwords to Enforce Case-Sensitivity

Starting with Oracle Database 11*g* 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 take advantage of enforced case-sensitive passwords, you must reset the passwords of existing users during the database upgrade procedure. For new database instances, there are no additional tasks or management requirements. For upgraded databases, each user password must be reset with an `ALTER USER` statement.

Alternatively, you can change the default to make the password verifier case insensitive. For regular users, set the initialization parameter `sec_case_sensitive_logon` to `false`:

```
sql> alter system set sec_case_sensitive_logon=false;
```

For `sysdba` and `sysoper` users, you can generate a new `orapw` file using the new command line switch `ignorecase`.

> **Note:** If the default Oracle Database release 11*g* security settings are in place, then passwords must be at least eight characters, and passwords such as `welcome` and `oracle` are not allowed. See *Oracle Database Security Guide* for more information.

> **See Also:** *Oracle Database Security Guide*

Understand Changes with Oracle Grid Infrastructure

Oracle Clusterware 11*g* release 2 and Oracle ASM 11*g* release 2 are both part of an Oracle Grid Infrastructure installation.

If Oracle Grid Infrastructure is installed for a single server, then it is deployed as an Oracle Restart installation with Oracle ASM. If Oracle Grid Infrastructure is installed for a cluster, then it is deployed as an Oracle Clusterware installation with Oracle ASM.

Oracle Restart enhances the availability of Oracle Database in a single-instance environment. If you install Oracle Restart, and there is a temporary failure of any part of the Oracle Database software stack, including the database, listener, and Oracle ASM instance, Oracle Restart automatically restarts the failed component. In addition, Oracle Restart starts all these components when the database host computer is restarted. The components are started in the proper order, taking into consideration the dependencies among components.

Oracle Clusterware is portable cluster software that allows clustering of single servers so that they cooperate as a single system. Oracle Clusterware also provides the required infrastructure for Oracle RAC. In addition, Oracle Clusterware enables the protection of any Oracle application or any other application within a cluster. In any case Oracle Clusterware is the intelligence in those systems that ensures required cooperation between the cluster nodes.

Understand Oracle ASM and Oracle Grid Infrastructure Installation and Upgrade

In earlier releases, Oracle ASM was installed as part of the Oracle Database installation. With Oracle Database 11*g* release 2 (11.2), Oracle ASM is installed when you install the grid infrastructure components and shares an Oracle home with Oracle Clusterware when installed in a cluster such as with Oracle RAC or with Oracle Restart on a standalone server.

If you have an existing Oracle ASM instance, you can either upgrade it during the installation of Oracle Grid Infrastructure, or you can upgrade it after the installation. However, be aware that several Oracle ASM features are disabled until you upgrade Oracle ASM, and Oracle Clusterware management of Oracle ASM does not function correctly until Oracle ASM is upgraded, because Oracle Clusterware only manages Oracle ASM when it is running in the grid infrastructure home. For this reason, Oracle recommends that if you do not upgrade Oracle ASM at the same time as you upgrade Oracle Clusterware, then you should upgrade Oracle ASM immediately afterward.

You can upgrade an Oracle ASM instance using Oracle ASM Configuration Assistant (ASMCA).

See Also:

- "Required Tasks After Oracle Grid Infrastructure Upgrades" on page 17
- "Preparing to Upgrade Oracle ASM" on page 18
- "Upgrading Oracle ASM" on page 19
- "Recommended Tasks After Oracle ASM Upgrades" on page 22
- "Tasks to Complete Only After Manually Upgrading Oracle Database" on page 12

In earlier releases, you could use Database Upgrade Assistant (DBUA) to upgrade either an Oracle Database, or Oracle ASM. That is no longer the case. You can only use DBUA to upgrade an Oracle Database instance. Use Oracle ASM Configuration Assistant (ASMCA) to upgrade Oracle ASM.

Add New Features as Appropriate

Oracle Database New Features Guide describes many of the new features available in the new Oracle Database 11*g* release. Determine which of these new features can benefit the database and applications; then, develop a plan for using these features.

It is not necessary to make any immediate changes to begin using your new Oracle Database software. You might prefer to introduce these enhancements into your database and corresponding applications gradually.

Chapter 5, "Upgrading Applications After Upgrading Oracle Database" describes ways to enhance your applications so that you can take advantage of the features of the new Oracle Database 11*g* release. However, before you implement new features, test your applications and successfully run them with the upgraded database.

Develop New Administrative Procedures as Needed

After familiarizing yourself with the features of the new Oracle Database 11*g* release, review your database administration scripts and procedures to determine whether any changes are necessary.

Coordinate your changes to the database with the changes that are necessary for each application. For example, by enabling integrity constraints in the database, you might be able to remove some data checking from your applications.

Set Threshold Values for Tablespace Alerts

An upgraded Oracle Database 11*g* Release 1 (11.1) database has the Tablespace Alerts disabled (the thresholds are set to null). Tablespaces in the database that are candidates for monitoring must be identified and the appropriate threshold values set.

The default threshold values (for a newly created Oracle Database 11*g* Release 1 (11.1) database) are:

- 85% full warning
- 97% full critical

Migrate From Rollback Segments to Automatic Undo Mode

This section describes the steps to migrate the database that is being upgraded from using rollback segments (manual undo management) to automatic undo management.

Starting with Oracle Database 11*g* Release 1 (11.1), automatic undo management is the default undo space management mode. The `UNDO_MANAGEMENT` initialization parameter specifies which undo space management mode the system should use, as follows:

- If `UNDO_MANAGEMENT=AUTO` (or if `UNDO_MANAGEMENT` is not set), then the database instance starts in automatic undo management mode.

  A null `UNDO_MANAGEMENT` initialization parameter defaults to automatic undo management mode in Oracle Database 11*g* Release 1 (11.1), but it defaults to manual undo management mode in earlier releases. You must therefore use caution when upgrading a previous release to Oracle Database 11*g*.

- If `UNDO_MANAGEMENT=MANUAL`, then undo space is allocated externally as rollback segments.

If you are currently using rollback segments to manage undo space, then Oracle recommends that you migrate your Oracle Database 11*g* Release 1 (11.1) database to automatic undo management. This change requires that you first create an undo tablespace before opening a newly upgraded database to use automatic undo management. The required size of undo tablespace depends upon the system workload and Flashback requirements.

To migrate to automatic undo management, perform the following steps:

1. Set `UNDO_MANAGEMENT=MANUAL`.
2. Start the instance again and run through a standard business cycle to obtain a representative workload. Doing this to assess the workload and compute the size of the undo tablespace required for automatic undo management.
3. After the standard business cycle completes, run the following function to collect the undo tablespace size and help with the sizing of the undo tablespace (DBA privileges are required to run this function):

   ```
   DECLARE
   ```

```
   utbsiz_in_MB NUMBER;
BEGIN
   utbsiz_in_MB := DBMS_UNDO_ADV.RBU_MIGRATION;
end;
/
```

This function runs a PL/SQL procedure that provides information on how to size your new undo tablespace based on the configuration and usage of the rollback segments in your system. The function returns the sizing information directly.

4. Create an undo tablespace of the required size and turn on the automatic undo management by setting UNDO_MANAGEMENT=AUTO or by removing the parameter.
5. For Oracle RAC configurations, repeat these steps on all instances.

Configure Oracle Data Guard Broker

The Data Guard broker property LocalListenerAddress has been deprecated as of release 11.2.0.1 due to changes with broker communication and the redo transport setting.

The broker property InitialConnectIdentifier has been changed to DGConnectIdentifier. The value of DGConnectIdentifier is used for all Data Guard network traffic, all of the time. If you are upgrading an Oracle Database release 10*g* configuration, which requires you to first upgrade to Oracle Database 11*g* Release 1 (11.1), the value that exists for InitialConnectIdentifier is retained as the new value for DGConnectIdentifier for the database. When upgrading an Oracle RAC database, the database administrator must ensure that the value for the InitialConnectIdentifier property reaches all instances.

Migrate Tables from the LONG Data Type to the LOB Data Type

LOB data types (BFILE, BLOB, CLOB, and NCLOB) can provide many advantages over LONG data types. See *Oracle Database Concepts* for information about the differences between LONG and LOB data types.

In Oracle9*i* Release 1 (9.0.1) and later, you can use the ALTER TABLE statement to change the data type of a LONG column to CLOB and that of a LONG RAW column to BLOB.

In the following example, the LONG column named long_col in table long_tab is changed to data type CLOB:

```
SQL> ALTER TABLE Long_tab MODIFY ( long_col CLOB );
```

After using this method to change LONG columns to LOBs, all the existing constraints and triggers on the table are still usable. However, all the indexes, including Domain indexes and Functional indexes, on all columns of the table become unusable and must be rebuilt using an ALTER INDEX...REBUILD statement. Also, the Domain indexes on the LONG column must be dropped before changing the LONG column to a LOB.

> **See Also:** *Oracle Database SecureFiles and Large Objects Developer's Guide* for information about modifying applications to use LOB data

Test the Upgraded Production Database

If you upgraded a test database to the new Oracle Database 11*g* release and then tested it, then you can now repeat those tests on the production database that you upgraded to the new Oracle Database 11*g* release. Compare the results, noting anomalies. Repeat the test upgrade as many times as necessary.

Test the newly upgraded production database with existing applications to verify that they operate properly with a new Oracle database. You also might test enhanced functions by adding available Oracle Database features. However, first ensure that the applications operate in the same manner as they did before the upgrade.

> **See Also:** Chapter 5, "Upgrading Applications After Upgrading Oracle Database" for more information on using applications with Oracle Database

Recommended Tasks After Upgrading an Oracle Database 10*g* Release 1 (10.1) Database

Performing the following tasks is recommended, but not required, after you have upgraded from Oracle Database 10*g* Release 1 (10.1) or Oracle Database 10*g* Release 2 (10.2).

- Upgrade Change Data Capture
- Configure Secure HTTP
- Provide Anonymous Access to XML DB Repository Data Through HTTP

Upgrade Change Data Capture

Starting with Oracle Database 10*g* Release 2 (10.2), Asynchronous Change Data Capture (CDC) no longer requires the same operating system for source and target databases. This feature enables a heterogeneous CDC setup with different operating systems and Oracle Database releases, enabling asynchronous CDC to leverage any existing Oracle9*i* Release 2 (9.2) system as a source.

> **See Also:** *Oracle Database Data Warehousing Guide* for complete information on upgrading an Oracle9*i* Release 2 (9.2) or Oracle Database 10*g* Release 1 (10.1) database to the new Oracle Database 11*g* release with Change Data Capture, and supported configurations for the Distributed HotLog mode of Change Data Capture

Configure Secure HTTP

To configure HTTPS access to Oracle XML DB, follow the steps in this section to provide correct configuration information.

When a database is upgraded to Oracle Database 10*g* Release 2 (10.2) or later, the XML schema for the Oracle XML DB configuration file is automatically upgraded so that the Oracle XML DB configuration file (located at `/xdbconfig.xml` in the repository) can have two additional elements, *`http2-port`* and *`http2-protocol`*. These elements are not added to the Oracle XML DB configuration file by default during an upgrade. If you need support for HTTPS, then you must edit the configuration file to add these two new elements (see the XML schema for their exact locations), and to set the value of *`http2-protocol`* to `tcps`. The value of *`http2-port`* should be different from the value of *`http-port`*.

In addition to specifying the parameters *`http2-port`* and *`http2-protocol`* in the Oracle XML DB configuration file, you must configure the database and the listener to enable Oracle XML DB to use HTTPS. Additionally, if the steps in the following procedure were not performed before the upgrade, then you must perform them after the upgrade.

To enable Oracle XML DB to use HTTPS

1. Enable the HTTP listener and the database to use SSL
2. Enable launching of a TCPS dispatcher

For more information on how to do this, see *Oracle XML DB Developer's Guide*.

> **Note:** If Oracle XML DB is not installed on the system, then you must install it during the upgrade procedure. Oracle XML DB is required to properly maintain the access control lists (ACLs).

Provide Anonymous Access to XML DB Repository Data Through HTTP

If anonymous access to XML DB repository data through HTTP is not required, then you are not required to perform this step. If anonymous access to XML DB repository data through HTTP is required, then you must provide correct configuration information, as described in this section. The administrator must carefully consider whether anonymous access is to be allowed, given the inherent security risks.

When a database is upgraded to Oracle Database 10*g* Release 2 (10.2) or later, the XML schema for the Oracle XML DB configuration file (located at `/xdbconfig.xml` in the repository) is automatically upgraded so that it can have an additional element, `allow-repository-anonymous-access`. This element is of Boolean type which means it can have a value of `true` or `false`. You can use this element to disallow unauthenticated access to your Oracle XML DB Repository data through HTTP even if you unlock the `ANONYMOUS` user account. The `allow-repository-anonymous-access` element is not added to the Oracle XML DB configuration file by default during an upgrade, but when this element is missing, it is interpreted as `false`.

Therefore, anonymous access to XML DB repository data through HTTP is disabled when you upgrade to Oracle Database 10*g* Release 2 (10.2) or later. If you want to have anonymous access to XML DB repository data through HTTP, then you must change the configuration file to set this new element to `true`, in addition to unlocking the `ANONYMOUS` user account.

> **Caution:** There is an inherent security risk associated with allowing unauthenticated access to the repository.

> **See Also:** *Oracle XML DB Developer's Guide* for more information about the `allow-repository-anonymous-access` element and configuring Oracle XML DB

Recommended Tasks After Upgrading an Oracle Express Edition Database

An Oracle Express Edition database contains only a subset of the components available in a Standard Edition or Enterprise Edition database. After upgrading to the new Oracle Database 11*g* release, you can use the Database Configuration Assistant to install additional components into your database. If you did not install Enterprise Manager Database Control during the DBUA upgrade, then you can install it, along with any other components you would like to have in the database.

Recommended Tasks After Upgrading an Oracle RAC Database

Oracle Real Application Clusters (Oracle RAC) 11*g* Release 2 (11.2) introduces the Single Client Access Name (SCAN). The SCAN is a single name that resolves to three IP addresses in the public network. When an earlier release of an Oracle RAC database is upgraded to 11g release 2 (11.2), it is registered with SCAN listeners as remote listeners, and also continues to register with all node listeners. You can configure clients to use SCANs, or continue to use the node listeners. If you migrate all of your client connections to use SCANs, you can then remove the node listeners from the `REMOTE_LISTENERS` parameter. However, you cannot remove the listeners themselves, because only node listeners can create dedicated servers for the database.

> **See Also:** *Oracle Clusterware Administration and Deployment Guide* for more information on the Single Client Access Name (SCAN)

Tasks to Complete Only After Manually Upgrading Oracle Database

If you are performing a manual upgrade of Oracle Database rather than using DBUA, then you must perform additional tasks after your database is upgraded.

- Change Passwords for Oracle Supplied Accounts
- Create Password File with ORAPWD
- Migrate Your Initialization Parameter File to a Server Parameter File
- Upgrade Oracle Text
- Upgrade the Oracle Clusterware Configuration
- Adjust the Initialization Parameter File for the New Release
- Configure Enterprise Manager
- Set CLUSTER_DATABASE Initialization Parameter For Oracle RAC

Change Passwords for Oracle Supplied Accounts

Depending on the release from which you upgraded, there might be new Oracle supplied accounts. Oracle recommends that you lock all Oracle supplied accounts except for `SYS` and `SYSTEM`, and expire their passwords, thus requiring new passwords to be specified when the accounts are unlocked.

> **Note:** If the default Oracle Database 11*g* security settings are in place, then passwords must be at least eight characters, and passwords such as `welcome` and `oracle` are not allowed. See *Oracle Database Security Guide* for more information.

You can view the status of all accounts by issuing the following SQL statement:

```
SQL> SELECT username, account_status
         FROM dba_users
         ORDER BY username;
```

To lock and expire passwords, issue the following SQL statement:

```
SQL> ALTER USER username PASSWORD EXPIRE ACCOUNT LOCK;
```

Create Password File with ORAPWD

If the `REMOTE_LOGIN_PASSWORDFILE` initialization parameter is set to either `exclusive` or `shared`, create a password file with `ORAPWD`.

> **See Also:** *Oracle Database Administrator's Guide* for more information about creating password files

Migrate Your Initialization Parameter File to a Server Parameter File

If you are currently using a traditional initialization parameter file, then perform the following steps to migrate to a server parameter file:

1. If the initialization parameter file is located on a client computer, then transfer the file from the client computer to the server computer.

> **Note:** If you are using Oracle RAC, then you must combine all of your instance-specific initialization parameter files into a single initialization parameter file. Instructions for doing this, and other actions unique to using a server parameter file for cluster databases, are discussed in:
>
> - *Oracle Real Application Clusters Administration and Deployment Guide*
> - The *Oracle Real Application Clusters Installation Guide* for your operating system

2. 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.
3. Start up the instance using the newly-created server parameter file.

> **See Also:**
>
> - *Oracle Database Administrator's Guide* for more information about creating server parameter files
> - *Oracle Database SQL Language Reference* for information about the `CREATE SPFILE` statement

Upgrade Oracle Text

After an upgrade to the new Oracle Database 11*g* release, copy the following files from the previous Oracle home to the new Oracle home:

- Stemming user-dictionary files
- User-modified `KOREAN_MORPH_LEXER` dictionary files
- `USER_FILTER` executables

These files affect all databases installed in the given Oracle home.

You can obtain a list of these files by doing the following:

1. Looking at `$ORACLE_HOME/ctx/admin/ctxf102.txt`
2. Executing `$ORACLE_HOME/ctx/admin/ctxf102.sql` as database user `SYS`, `SYSTEM`, or `CTXSYS`

If your Oracle Text index uses KOREAN_LEXER which was deprecated in Oracle 9*i* and desupported in Oracle Database 10*g* Release 2 (10.2), see Note 300172.1 on My Oracle Support for further information on manual migration from KOREAN_LEXER to KOREAN_MORPH_LEXER.

See Also:

- *Oracle Text Reference* for more information about these files
- *Oracle Text Application Developer's Guide* for information about upgrading your applications from previous releases of Oracle Text
- My Oracle Support at https://support.oracle.com

Upgrade the Oracle Clusterware Configuration

If you are using Oracle Clusterware, then you must upgrade the Oracle Clusterware keys for the database.

Run srvctl for release 11.2.0.3 to upgrade the database. For example:

```
%11.2.0.3_ORACLE_HOME/bin/srvctl upgrade database -d <name> -o 11.2.0.3_ORACLE_
HOME
```

Caution: By default, any named user may create a server pool. To restrict the operating system users that have this privilege, Oracle strongly recommends that you add specific users to the CRS Administrators list.

See Also: *Oracle Clusterware Administration and Deployment Guide* for more information about adding users to the CRS Administrators list

Adjust the Initialization Parameter File for the New Release

Each release of Oracle Database introduces new initialization parameters, deprecates some initialization parameters, and makes some initialization parameters obsolete. You must adjust the parameter file to account for these changes and to take advantage of new initialization parameters that might be beneficial to your system. Additionally, when you perform a manual upgrade without using DBUA, the tnsnames.ora file is not automatically populated with new configuration information and settings. Therefore, you may need to manually update tnsnames.ora and adjust local_listener and remote_listener parameter references if these need to be resolved.

See Also:

- The "What's New in Oracle Database Reference" section of *Oracle Database Reference* for a list of the new initialization parameters in the new Oracle Database 11*g* release, and for information about each parameter
- Appendix A, "Behavior Changes After Upgrading Oracle Database" for lists of obsolete and deprecated initialization parameters in the new Oracle Database 11*g* release

Setting the COMPATIBLE Initialization Parameter

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 release, set the COMPATIBLE initialization parameter based on the compatibility level you want for your new database.

Complete the following steps to set the COMPATIBLE initialization parameter to a higher value:

1. Perform a backup of your database before you raise the COMPATIBLE initialization parameter (optional).

   Raising the COMPATIBLE initialization parameter might cause your database to become incompatible with earlier releases of Oracle Database, and a backup ensures that you can return to the earlier release if necessary.

   See Also: *Oracle Database Backup and Recovery User's Guide* for more information about performing a backup

2. If you are using a server parameter file, then complete the following steps:

   a. Update the server parameter file to set or change the value of the COMPATIBLE initialization parameter.

   For example, to set the COMPATIBLE initialization parameter to 11.0.0, enter the following statement:

   ```
   SQL> ALTER SYSTEM SET COMPATIBLE = '11.0.0' SCOPE=SPFILE;
   ```

   b. Shut down and restart the instance.

   Note: When upgrading systems with HARD-compliant storage (Hardware Assisted Resilient Data), consider the following:

   - If the COMPATIBLE parameter is set to a release number earlier than 11.0.0, then you cannot locate the server parameter file (SPFILE) on HARD storage.
   - If the COMPATIBLE parameter is set to 11.0.0, then you can optionally locate the server parameter file on HARD storage.

   Because the default SPFILE location (*ORACLE_HOME*/dbs) might not be on a HARD-compliant storage system, it is likely you must provide a parameter file that specifies the location of the SPFILE.

   See Also: *Oracle Database High Availability Overview* or *Oracle Database Concepts* for more information on HARD storage

3. If you are using an initialization parameter file, then complete the following steps:

   a. Shut down the instance if it is running:

   ```
   SQL> SHUTDOWN IMMEDIATE
   ```

   b. Edit the initialization parameter file to set or change the value of the COMPATIBLE initialization parameter.

   For example, to set the COMPATIBLE initialization parameter to 11.0.0, enter the following in the initialization parameter file:

   ```
   COMPATIBLE = 11.0.0
   ```

   c. Start the instance using STARTUP.

> **Note:** If you are using an ASM disk group, then the diskgroup's compatibility attribute must match or be lower than that of the database compatibility parameter in `init.ora`.

Configuring tnsnames.ora and Listener Parameters

After performing a manual upgrade, you may need to adjust `local_listener` and `remote_listener` parameter references if they need to be resolved in `tnsnames.ora`. DBUA handles changes to network naming and listeners during automatic upgrades, but during a manual upgrade, `tnsnames.ora` is not changed, nor are the listeners.

See Also:

- Local Naming Parameters (tnsnames.ora) in *Oracle Database Net Services Reference*
- "Configuring the tnsnames.ora File After Installation" in *Oracle Database Net Services Administrator's Guide*
- "Configuring and Administering Oracle Net Listener" in *Oracle Database Net Services Administrator's Guide* for information on registering information with a local listener and a remote listener
- "Net Service Names (tnsnames.ora File)" in *Oracle Real Application Clusters Installation Guide* for Windows
- "Net Service Names (tnsnames.ora File)" in *Oracle Real Application Clusters Installation Guide for Linux and UNIX*

Configure Enterprise Manager

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

If your database is being managed by Oracle Enterprise Manager Database Control or Oracle Enterprise Manager Grid Control, then use the following command to update the configuration:

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

You must run this from the Oracle home of the new Oracle Database 11*g* release. When prompted, provide the Oracle home from which the configuration is being upgraded.

You can also configure Enterprise Manager using DBCA. Select the **Configure Database Options** option, and then select the **Enterprise Manager Repository** option.

See Also: *Oracle Enterprise Manager Advanced Configuration*

Set CLUSTER_DATABASE Initialization Parameter For Oracle RAC

For upgrades of Oracle RAC databases, in "Preparing the New Oracle Home for a Manual Upgrade" on page 3-48, you were instructed to set the `CLUSTER_DATABASE` initialization parameter to `false` before upgrading a cluster database. Now that the upgrade is finished, you must set this parameter to `true`.

Required Tasks After Oracle Grid Infrastructure Upgrades

Oracle ASM release 11.2 and later are included as part of an Oracle Grid Infrastructure installation.

If you upgrade Oracle Clusterware and Oracle ASM for a cluster, then Oracle Clusterware and Oracle ASM are both located in the same home, which is referred to as *grid home*. You can have one installation owner that owns all Oracle software installations, or you can use role-allocated owners, in which case you use a separate software owner for the grid infrastructure installation, and separate software owners for one or more Oracle Database installations.

The following tasks are required after an upgrade from Oracle ASM, performed as a separate installation procedure, to an Oracle ASM installation as part of the Oracle Grid Infrastructure:

- Using Environment Variables for Grid Infrastructure Installations
- Upgrading An Earlier Release of Oracle ASM to Oracle Grid Infrastructure

> **See Also:** *Oracle Grid Infrastructure Installation Guide* for your platform for more information about role-allocated installation owners

Using Environment Variables for Grid Infrastructure Installations

If your operating system is Linux or UNIX, then you may need to change environment variable settings after performing an upgrade.

If you use a single Oracle installation owner for all installations, then be aware that you should change environment variables such as `ORACLE_HOME` either to an Oracle Database home, or to the grid home, depending on whether you are administering an Oracle Database instance as part of database administration, or administering an Oracle ASM instance as part of storage administration.

If you use role-allocated Oracle installation owners, so that you have a separate owner for the Oracle Grid Infrastructure (Oracle Clusterware and Oracle ASM) software, then set the following environment variables for the grid infrastructure installation owner so that they point to the directories of the Oracle ASM home in the grid infrastructure home:

- `ORACLE_HOME`
- `PATH`

Also, check that your `oratab` file and any client scripts for Oracle ASM that set the value of `ORACLE_HOME` point to the Oracle ASM home in the grid infrastructure home.

> **Note:** If you are upgrading a clustered Oracle ASM installation to an Oracle Grid Infrastructure for a cluster installation, then perform these checks on all cluster member nodes.

> **See Also:** Your operating system-specific Oracle Database installation documents for information about setting other important environment variables on your operating system.

Upgrading An Earlier Release of Oracle ASM to Oracle Grid Infrastructure

In earlier releases, Oracle ASM was installed as part of the Oracle Database installation. With Oracle Database 11*g* Release 2 (11.2), Oracle ASM is installed when you install the grid infrastructure components. If you install Oracle Grid Infrastructure for a cluster, then it is part of the grid home, with Oracle Clusterware. If you install Oracle Grid Infrastructure for a single server, then Oracle ASM shares an Oracle home with Oracle Restart.

If you want to upgrade an existing Oracle ASM, then you must upgrade Oracle ASM by running an Oracle Grid Infrastructure upgrade. If you do not have Oracle ASM installed and you want to use Oracle ASM as your storage option, then you must complete an Oracle Grid Infrastructure installation before you start your Oracle Database installation.

> **See Also:** *Oracle Grid Infrastructure Installation Guide* or *Oracle Database Installation Guide* for your platform

Oracle ASM Configuration Assistant (ASMCA) supports installing and configuring Oracle ASM instances, disk groups, volumes, and Oracle ASM Cluster File System (Oracle ACFS). Additionally, you can use the ASMCA command-line interface as a non-GUI utility (command name `asmca`).

You can use the `asmca` command to complete the upgrade separately. However, you must run `asmca` soon after you upgrade Oracle Clusterware, as Oracle ASM management tools such as `srvctl` do not work until Oracle ASM is upgraded.

> **Note:** For cluster upgrades, ASMCA performs a rolling upgrade only if the earlier release of Oracle ASM is either 11.1.0.6 or 11.1.0.7. Otherwise, ASMCA performs a normal upgrade, in which case ASMCA brings down all Oracle ASM instances on all nodes of the cluster, and then brings them all up in the new grid infrastructure home.

Preparing to Upgrade Oracle ASM

Note the following information if you intend to perform rolling upgrades of Oracle ASM:

- You cannot change the owner of a home as part of an upgrade. For example, if you plan to install Oracle Grid Infrastructure as the user `grid`, then the home of an existing Oracle ASM must be owned by the user `grid` before you upgrade it.
- The active release of Oracle Clusterware must be 11*g* release 2 (11.2). To determine the active release, enter the following command:

  ```
  $ crsctl query crs activeversion
  ```

> **See Also:** *Oracle Clusterware Administration and Deployment Guide* for more information about `crsctl` and adding users to the CRS Administrators list

- You can upgrade a single instance Oracle ASM installation to a clustered Oracle ASM installation. However, you can only upgrade an existing single instance Oracle ASM installation if you run the installation from the node on which the Oracle ASM installation is installed. You cannot upgrade a single instance Oracle ASM installation on a remote node.
- You must ensure that any rebalance operations on your existing Oracle ASM installation are completed before starting the upgrade process.

- During the upgrade process, you alter the Oracle ASM instances to an upgrade state. Because this upgrade state limits Oracle ASM operations, you should complete the upgrade process soon after you begin. The following are the operations allowed when an Oracle ASM instance is in the upgrade state:
  - Diskgroup mounts and dismounts
  - Opening, closing, resizing, or deleting database files
  - Recovering instances
  - Queries of fixed views and packages: Users are allowed to query fixed views and run anonymous PL/SQL blocks using fixed packages, such as `dbms_diskgroup`)

Upgrading Oracle ASM

The procedure in this section describes how to upgrade Oracle ASM using Oracle ASM Configuration Assistant (ASMCA).

To upgrade Oracle ASM

1. Log on as the installation owner of the Oracle Grid Infrastructure installation.
2. If you are upgrading on a cluster, then on the node you plan to start the upgrade, set the environment variable `ASMCA_ROLLING_UPGRADE` as `true`. For example:

   ```
   $ export ASMCA_ROLLING_UPGRADE=true
   ```

3. From the Oracle Grid Infrastructure 11*g* release 2 (11.2) home, start ASMCA. For example:

   ```
   $ cd /u01/11.2/grid/bin
   $ ./asmca
   ```

4. Select **Upgrade**.

   The Oracle ASM Configuration Assistant upgrades Oracle ASM in succession for all nodes in the cluster.

   > **See Also:** *Oracle Database Storage Administrator's Guide* for additional information about preparing an upgrade plan for Oracle ASM, and for starting, completing, and stopping Oracle ASM upgrades

Required Tasks After Oracle ASM Upgrades

This section contains the tasks that are required after an Oracle ASM Upgrade, and additional considerations.

- Set Environment Variables
- Single-Instance Oracle ASM Upgrade
- Cluster Oracle ASM Upgrade
- Additional Considerations After Oracle ASM Upgrades

Set Environment Variables

If your operating system is Linux or UNIX, then make sure that the following environment variables point to the directories of the new Oracle Database 11*g* release:

- `ORACLE_HOME`

- PATH

Also check that your oratab file and any client scripts that set the value of ORACLE_HOME point to the Oracle home of the new Oracle Database 11*g* release.

Note: The ORACLE_HOME, PATH, and oratab checks are required only if you upgrade manually. DBUA automatically points oratab to the new Oracle home. Client scripts must be checked no matter how you upgrade.

If you are upgrading a clustered Oracle ASM, then perform these checks on all nodes in which this clustered Oracle ASM has instances configured.

See Also: Your operating system-specific Oracle Database installation documents for information about setting other important environment variables on your operating system.

Single-Instance Oracle ASM Upgrade

The following procedure assumes that Oracle ASM is installed in *Oracle home 1* (OH1) and the operating system user is orauser.

To perform a single-instance upgrade of Oracle ASM

1. As orauser, upgrade Oracle ASM to release 11.2 using OUI and ASMCA. The new Oracle ASM release 11.2 runs in the grid infrastructure home. Oracle ASM should still be running as orauser.
2. As orauser, bring down the Oracle ASM instance and the listener.
3. As root, run `/etc/init.d/init.cssd stop` to stop CSS.
4. As a new user (asmuser), install 11.2 in a third Oracle home (OH3). This should be a software-only installation.
5. As root, run `localconfig reset` from OH3.
6. Update /etc/oratab so that OH3 is the Oracle home with the +ASM entry.
7. Copy listener.ora, sqlnet.ora, and tnsnames.ora from OH2.
8. Run EMCP to change the Oracle ASM instance path and connect-string role.
9. Make sure the disks are owned by asmuser and by OSASM for Oracle ASM. They should also have the 0660 permission set.
10. As asmuser, start the listener.
11. As asmuser, start Oracle ASM (connect as SYSASM).
12. Run the command, `GRANT sysasm TO sys`.

Cluster Oracle ASM Upgrade

To perform an upgrade of Oracle ASM on a cluster

1. As orauser, upgrade Oracle ASM to release 11.2 using OUI and ASMCA. The new Oracle ASM release 11.2 should be running in a new *Oracle home 2* (OH2). Oracle ASM should still be running as orauser.
2. Bring down the Oracle ASM and Listener resources from CRS home.

3. As a new user (`crs` for example), install 11.2 into a third Oracle home (`OH3`) to match that of the Grid Infrastructure home. This must be a software-only installation.

4. From CRS home, run:

```
srvctl remove listener -n node_name
srvctl add listener -n node_name -o OH3
srvctl modify asm -n node_name -i ASM_instance_name -o ORACLE_HOME_path
```

> **Caution:** By default, any named user may create a server pool. To restrict the operating system users that have this privilege, Oracle strongly recommends that you add specific users to the CRS Administrators list.
>
> **See Also:** *Oracle Clusterware Administration and Deployment Guide* for more information about adding users to the CRS Administrators list

5. Update `/etc/oratab` so that `OH3` is the Oracle home with the `+ASM` entry.

6. Copy `listener.ora`, `sqlnet.ora`, and `tnsnames.ora` from `OH2`.

7. Run EMCP to change the Oracle ASM instance path and `connect-string` role.

8. Make sure the disks are owned by `asmuser` and by OSASM for Oracle ASM. They should also have the `0660` permission set.

9. Start Oracle ASM and Listener resources from the new Oracle ASM 11*g* ORACLE_HOME or the new Oracle Database 11*g* ORACLE_HOME.

10. Run the command, `GRANT sysasm TO sys`.

If you have clustered Oracle ASM instances, then you also have the option of performing a rolling Oracle ASM upgrade. A rolling upgrade enables you to independently upgrade or patch Oracle ASM nodes without affecting database availability, thus providing greater uptime.

> **See Also:** *Oracle Database Storage Administrator's Guide* for more information on rolling Oracle ASM upgrades

Additional Considerations After Oracle ASM Upgrades

If you separate the operating system user ownership of the Oracle Grid Infrastructure binaries and the Oracle Database installation owners of one or more databases, then you must migrate the operating system user of an upgraded Oracle ASM or database home as described in "Role-Allocated Software Owners and Database Upgrade After Oracle ASM Upgrade" on page 21.

Role-Allocated Software Owners and Database Upgrade After Oracle ASM Upgrade

If you are migrating from one software binary owner (such as `oracle`) to multiple role-allocated software owner user accounts (such as `grid`, `oracle1`, `oracle2`), then change the owner of the existing Oracle ASM installation owner to the installation owner that you plan to use for the Oracle Grid Infrastructure installation.

There are three scenarios to consider:

- Keeping the Existing User as the Oracle ASM Operating System User
- Changing the Operating System User for Single-Instance Oracle ASM

- Changing the Operating System User for an Oracle RAC Database

> **See Also:** *Oracle Database Storage Administrator's Guide* for information on making an Oracle ASM disk group compatible with Oracle Database 10*g* and Oracle Database 11*g*, and for additional information about Oracle ASM upgrades

Keeping the Existing User as the Oracle ASM Operating System User

If you are using the same operating system user for your Oracle Grid Infrastructure installation that you used for your existing Oracle ASM installation, then run Oracle Universal Installer (OUI) to perform a grid infrastructure installation, and select the upgrade option. OUI automatically upgrades your existing Oracle ASM installation from the prior release to 11*g* Release 2 (11.2) in the Oracle Grid Infrastructure home.

Changing the Operating System User for Single-Instance Oracle ASM

Consider your earlier release Oracle ASM installation is installed in Oracle home 4 (`OH4`) and currently running `oracle` as the operating system user, and you want to change the Oracle ASM operating system user to `grid`. This is useful if you have two databases using Oracle ASM, and you had installed Oracle ASM with an installation owner that is the same as that for the existing databases, and you want to change the operating system installation owner of Oracle ASM to enable separate databases to run as separate operating system users, where neither Oracle Database installation owner has Oracle Grid Infrastructure binary ownership.

Changing the Operating System User for an Oracle RAC Database

There may be scenarios where you must change the operating system user for an Oracle RAC database. For example, if your earlier release database is installed in Oracle home 4 (`OH4`) and currently running `oracle` as the operating system user, then you should consider changing the Oracle ASM operating system user to `grid`. Changing the operating system user of Oracle ASM enables separate databases to run as separate operating system users, where no Oracle Database installation owner has grid infrastructure binary ownership.

Recommended Tasks After Oracle ASM Upgrades

Performing the following tasks is recommended, but not required, after you have upgraded Oracle ASM:

- Reset Oracle ASM Passwords to Enforce Case-Sensitivity
- Advance the Oracle ASM and Oracle Database Disk Group Compatibility
- Set Up Oracle ASM Preferred Read Failure Groups

You should also consider performing the following tasks, discussed earlier in this chapter:

- "Add New Features as Appropriate" on page 7
- "Develop New Administrative Procedures as Needed" on page 8

Reset Oracle ASM Passwords to Enforce Case-Sensitivity

Starting with Oracle Database 11*g* 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 take advantage of enforced case-sensitive passwords, you must reset the passwords of existing users during the database upgrade procedure. For new Oracle Oracle ASM instances, there are no additional tasks or management requirements. For upgraded Oracle ASM instances, each user password must be reset with an ALTER USER statement.

Note: If the default Oracle Database 11*g* security settings are in place, then passwords must be at least eight characters, and passwords such as welcome and oracle are not allowed. See *Oracle Database Security Guide* for more information.

See Also: *Oracle Database Security Guide*

Advance the Oracle ASM and Oracle Database Disk Group Compatibility

Starting with Oracle Database 11*g* Release 1 (11.1), you can advance the Oracle Database and the Oracle ASM disk group compatibility settings across software versions.

Caution: If you advance the COMPATIBLE.RDBMS attribute, then you *cannot* revert to the previous setting. Therefore, before advancing the COMPATIBLE.RDBMS attribute, ensure that the values for the COMPATIBLE initialization parameter for all of the databases that use the disk group are set to at least the new setting for COMPATIBLE.RDBMS before you advance the attribute value.

Advancing compatibility enables new features only available in the new release. However, doing so makes the disk group incompatible with older releases of the software. Note that advancing the on-disk compatibility is an irreversible operation.

You use the compatible.rdbms and compatible.asm attributes to specify the minimum software release required by the database instance and the Oracle ASM instance, respectively, to access the disk group. For example, the following ALTER DISKGROUP statement advances the Oracle ASM compatibility of the disk group asmdg2:

```
ALTER DISKGROUP asmdg2 SET ATTRIBUTE 'compatible.asm' = '11.1'
```

In this case, the disk group can be managed only by Oracle ASM software of release 11.1 or higher, while any database client of release 10.1 or higher can use the disk group.

See Also: *Oracle Database Storage Administrator's Guide* for complete information about disk group compatibility, and *Oracle Database SQL Language Reference* for more information about the disk group compatibility attributes on the ALTER DISKGROUP and CREATE DISKGROUP statements

Set Up Oracle ASM Preferred Read Failure Groups

Oracle ASM administrators can specify some disks to be preferred over others for read I/O operations. When Oracle ASM preferred read failure groups are defined, Oracle ASM can read from the extent that is closest to it, rather than always reading the primary copy.

See Also:

- *Oracle Clusterware Administration and Deployment Guide* for information about specifying failure groups settings in an extended cluster
- *Oracle Database Storage Administrator's Guide* for complete information about Oracle ASM preferred read failure groups, and specifying the new `ASM_PREFERRED_READ_FAILURE_GROUPS` initialization parameter to list failure group names that contain the preferred read disks for each node in a cluster
- *Oracle Database Reference* for the `ASM_PREFERRED_READ_FAILURE_GROUPS` initialization parameter

5 Upgrading Applications After Upgrading Oracle Database

Many new features and enhancements are available after upgrading to a new release of Oracle Database. To take advantage of the new features, you must upgrade applications running in the new release.

This chapter contains the following topics:

Overview of Upgrading Applications

You are not required to modify existing applications that do not use features available in the new Oracle Database 11*g* release. Existing applications running against a new Oracle Database 11*g* database function the same as they did in prior releases and achieve the same, or enhanced, performance.

Many new features and enhancements are available after upgrading to the new Oracle Database 11*g* release. Some of these features provide added features and functions, while others provide improved performance. Before you upgrade your applications, you should review these new features to decide which ones you want to use.

> **See Also:** *Oracle Database New Features Guide* for information about the features available in the new Oracle Database 11*g* release

Compatibility Issues for Applications When Upgrading

There might be compatibility issues between different releases of Oracle Database that could affect your applications. These compatibility issues result from differences in Oracle Database in various releases. Also, in each new release of Oracle Database, new Oracle reserved words might be added, changes might be made to initialization parameters, and changes might be made to the data dictionary.

When you upgrade your Oracle Database software to a new release, make sure that your applications do not use any Oracle reserved words, that your applications are compatible with the initialization parameters of the database, and that your applications are compatible with the data dictionary of the database. Finally, a new release of Oracle Database software might require certain operating system releases or the application of certain patchsets.

See Also:

- Appendix A, "Behavior Changes After Upgrading Oracle Database" for information about initialization parameter changes and data dictionary changes
- *Oracle Database SQL Language Reference* for a complete list of Oracle reserved words
- Your operating system-specific Oracle documentation for information about operating system requirements

Upgrading Precompiler and OCI Applications

The upgrade path is very similar for precompiler and OCI applications. This section guides you through your upgrade options for these applications and notes differences between precompiler and OCI applications whenever necessary.

Create a test environment before you upgrade your production environment. Your test environment should include your upgraded application and the new Oracle Database software. Also, your test environment should provide a realistic test of your application.

This section contains these topics:

- Understanding Software Upgrades and Your Client/Server Configuration
- Compatibility Rules for Applications When Upgrading Client/Server Software
- Upgrading Options for Your Precompiler and OCI Applications

See Also: *Pro*C/C++ Programmer's Guide*, *Pro*COBOL Programmer's Guide*, and *Oracle Call Interface Programmer's Guide* for more information about using these programming environments.

Understanding Software Upgrades and Your Client/Server Configuration

To understand your options for upgrading precompiler and OCI applications, you first must understand the type of software upgrade you are performing and your client/server configuration.

This section contains these topics:

- Types of Software Upgrades for Oracle Database
- Possible Client/Server Configurations

Types of Software Upgrades for Oracle Database

When you upgrade Oracle Database, you are typically upgrading to a major release or you are applying patches for a maintenance release. The types of upgrades possible for Oracle Database client and server software are as follows:

- Major Database Release Upgrade
- Database Maintenance Release Upgrade

Major Database Release Upgrade The upgrade changes the first digit of the release number. For example, upgrading from Oracle9*i* to Oracle Database 11*g* is a major database release upgrade.

Database Maintenance Release Upgrade The upgrade changes the second digit of the release number. For example, upgrading from Oracle Database 11*g* Release 1 (11.1) to Oracle Database 11*g* Release 2 (11.2) is a database maintenance release upgrade.

Possible Client/Server Configurations

Your precompiler and OCI applications run on the client in a client/server environment, where the Oracle Database server is the server. You can use one or more of the following client/server configurations in your environment:

- Different Computers
- Different Oracle Home Directories on the Same Computer
- Same Oracle Home

Different Computers The client software and the server software are on different computers, and they are connected through a network. The client and server environments are separate.

Different Oracle Home Directories on the Same Computer The client software and the server software are on the same computer, but they are installed in different Oracle home directories. Again, the client and server environments are separate.

Same Oracle Home The client software and server software are installed in the same Oracle home on the same computer. In this case, any upgrade of the server software is also an upgrade of the client software.

> **See Also:** *Oracle Database Concepts* and *Oracle Database Heterogeneous Connectivity User's Guide* for more information about client/server environments

Compatibility Rules for Applications When Upgrading Client/Server Software

This section covers compatibility rules that apply when you upgrade Oracle Database client or server software. The rules are based on the type of software upgrade you are performing and on your client/server configuration.

The following sections contain compatibility rules for the following types of upgrades:

- Upgrading the Oracle Database Server Software
- Upgrading the Oracle Database Client Software

> **Note:** This section uses the terms introduced in "Understanding Software Upgrades and Your Client/Server Configuration" on page 5-2.

Upgrading the Oracle Database Server Software

The following rules apply when you upgrade the Oracle Database server software.

- If You Do Not Change the Client Environment, Then You Are Not Required to Relink.
- Applications Can Run Against Newer or Older Oracle Database Server Releases

If You Do Not Change the Client Environment, Then You Are Not Required to Relink. If your client and server are on different computers or are in different Oracle home directories on the same computer, and you upgrade the Oracle Database server software without changing the client software, then you are not required to precompile, compile, or relink your applications. In these cases, the client software is separate from the server software and continues to function against the server.

However, if your applications are using the same Oracle home as the Oracle Database server, then your server upgrade also upgrades your client software, and you must follow the rules in "Upgrading the Oracle Database Client Software" on page 5-4.

> **Note:** It is possible to upgrade the Oracle Database server software but not install the new precompiler or OCI client software when you are using the same Oracle home for both. In this case, the client software is not upgraded. However, such a configuration is not recommended.

Applications Can Run Against Newer or Older Oracle Database Server Releases When you run a precompiler or OCI application against a database server, Oracle recommends that the release of the database server software be equal to or higher than the client software release, but this configuration is not strictly required. For example, if your client software is Oracle9*i* Release 2 (9.2.0.8), then it is recommended that your server software be Oracle9*i* Release 2 (9.2.0.8) or higher to run a precompiler application on the client against the server.

Upgrading the Oracle Database Client Software

Oracle recommends that you upgrade your client software to match the current server software. For example, if you upgrade your server to Oracle Database 11*g* Release 2 (11.2), then Oracle recommends upgrading the client software to Oracle Database 11*g* Release 2 (11.2) as well. Keeping the server and client software at the same release number ensures the maximum stability for your applications. In addition, the latest Oracle Database client software might provide added features and performance enhancements that were not available with previous releases.

The following rules apply when you upgrade the Oracle Database client software:

- Applications Can Be Linked with Newer Libraries
- Statically Linked Applications Must Always Be Relinked
- Relinking Dynamically Linked Applications

Applications Can Be Linked with Newer Libraries The code generated by precompiler applications can be linked with a release of the client library that is equal to or higher than the server release.

OCI applications can be linked with a version of the OCI runtime library that is equal to or higher than the version of the OCI library with which the application was developed.

Statically Linked Applications Must Always Be Relinked Statically linked OCI applications need to be re-linked for both major and minor releases, because the statically linked Oracle client-side library code may be incompatible with the error messages in the upgraded `ORACLE_HOME`. For example, if an error message was updated with additional parameters, then it will not be compatible with the statically-linked code.

Relinking Dynamically Linked Applications Dynamically linked OCI applications from Oracle Database 10*g* Release 1 (10.1) and later releases are upward compatible with the current release. That is, the Oracle client-side dynamic library is upward compatible with the previous version of the library. Oracle Universal Installer creates a symbolic link for the previous version of the library that resolves to the current version. Therefore, an application that is dynamically linked with the previous version of the Oracle client-side dynamic library does not need to be relinked to operate with the current version of the Oracle client-side library.

> **Note:** If the application is linked with a runtime library search path (such as `-rpath` on Linux), then the application may still run with the version of the Oracle client-side library with which it is linked. To run with the current version of the Oracle client-side library, it must be relinked.
>
> If the application is linked with the deferred option (for example, statically-linked application), it must be relinked.
>
> If the application is from a release before Oracle Database 10*g* Release 1 (10.1), then it must be relinked.

Upgrading Options for Your Precompiler and OCI Applications

You have the following options for upgrading your precompiler and OCI applications:

- Option 1: Leave the Application Unchanged. Do not relink, precompile, or compile the application, and do not change the application code. The application continues to work against the new Oracle Database 11*g* release.
- Option 2: Precompile or Compile the Application Using the New Software. Application code must be changed if any APIs are deprecated, desupported, or changed.
- Option 3: Change the Application Code to Use New Oracle Database 11g Features. Subsequently, precompile or compile and then relink the code.

These options are listed in order of increasing difficulty and increasing potential benefits. That is, Option 1 is the least difficult option, but it offers the least potential benefits, while Option 3 is the most difficult option, but it offers the most potential benefits. These options are discussed in the following sections.

Option 1: Leave the Application Unchanged

This option requires that the Oracle home environment of the application is not upgraded. You can leave the application unchanged, and it continues to work with the new Oracle Database 11*g* server. The major advantage to this option is that it is simple and easy. In addition, this option requires the least amount of administration, because you are not required to upgrade any of your client computers. If you have a large number of client computers, then avoiding the administrative costs of upgrading all of them can become very important.

The major disadvantage to this option is that your application cannot use the features that are available in the new Oracle Database 11*g* release. In addition, your application cannot leverage some of the possible performance benefits of the new Oracle Database 11*g* release.

Option 2: Precompile or Compile the Application Using the New Software

When upgrading from a maintenance release to the new Oracle Database 11*g* software, you must precompile or compile the application with the new software after making necessary code changes to account for APIs that are deprecated or changed.

Recompiling is not required if you are upgrading to a minor release within Oracle Database 11*g* software.

This option requires that you install the new Oracle Database client software on each client computer. You are required to precompile or compile, and relink your application only once, regardless of the number of clients you have.

By recompiling, you perform a syntax check of your application code. Some problems in the application code that were not detected by previous releases of the Oracle software might emerge when you precompile or compile with the new Oracle software. Therefore, precompiling and compiling with the new software often helps you detect and correct problems in the application code that might have gone unnoticed before.

Also, recompiling affords maximum stability for your application, because you are sure that it works with the new Oracle software. Further, your environment is ready for new development using the latest tools and features available. In addition, you might benefit from performance improvements that are available with the new Oracle software only after you recompile and relink.

Option 3: Change the Application Code to Use New Oracle Database 11*g* Features

You can make code changes to your application to take advantage of new Oracle Database 11*g* features. This option is the most difficult, but it can provide the most potential benefits. You gain all of the advantages described in "Option 2: Precompile or Compile the Application Using the New Software" on page 5. In addition, you also benefit from changes to your application that might leverage performance and scalability benefits available with the new Oracle Database 11*g* release. Further, you can add new features to your application that are available only with the new Oracle Database 11*g* release.

Become familiar with the features of the new Oracle Database 11*g* release by reading *Oracle Database New Features Guide*. Also, consult the Oracle documentation for your development environment so that you understand how to implement the features you want to use. For the precompilers, see *Pro*C/C++ Programmer's Guide* and *Pro*COBOL Programmer's Guide*. For OCI, see *Oracle Call Interface Programmer's Guide*.

When you have decided on the new features you want to use, change the code of your application to use these features. Follow the appropriate instructions in the following sections based on your development environment:

- Changing Precompiler Applications
- Changing OCI Applications

Changing Precompiler Applications Complete the following steps to change your precompiler application to use features of the new Oracle Database 11*g* release:

1. If you want to take advantage of features of the new Oracle Database 11*g* release, then incorporate the code for these new features into the existing application.
2. Precompile the application using the Oracle precompiler.
3. Compile the application.
4. Relink the application with the runtime library of the new Oracle Database 11*g* release, `SQLLIB`, which is included with the precompiler.

Changing OCI Applications Complete the following steps to change your OCI application to use features of the new Oracle Database 11*g* release:

1. Incorporate OCI calls of the new Oracle Database 11*g* release into the existing application.
2. Compile the application.
3. Relink the application with the runtime library of the new Oracle Database 11*g* release.

Upgrading SQL*Plus Scripts and PL/SQL

To use features and functions of the new Oracle Database 11*g* release, change existing SQL scripts to use the syntax of the new Oracle Database 11*g* release. Existing SQL scripts run unchanged on the new Oracle Database 11*g* release, and require no modification, if they do not use features and functions of the new Oracle Database 11*g* release.

Note that improved error checking in the new Oracle Database 11*g* release might now identify errors at compile time rather than at run time.

Related topic:

- Change to Evaluation of Numeric Literals

Change to Evaluation of Numeric Literals

Evaluation of numeric literals has changed such that at least one of the constants in a numeric computation with literals must be a decimal specified to the 10th place. This is because Oracle Database releases after Oracle Database 10*g* Release 1 (10.1) use `INTEGER` arithmetic (approximately 9 significant digits) for some expressions whereas Oracle9*i* Release 2 (9.2) used `NUMBER` arithmetic (approximately 38 significant digits).

Therefore, if you are dealing with results of greater than 9 significant digits, then one of the literals should be in decimal format to prevent numeric overflow errors. For example, in Oracle Database 10*g*, the computation of `v1` in the following example causes a numeric overflow error:

```
DECLARE
  v1 NUMBER(38);
BEGIN
  v1 := 256*256*256*256;
  DBMS_OUTPUT.PUT_LINE(v1);
END;
/
```

The solution to the error is to specify one of the numeric literals as a decimal (256.0), as follows:

```
DECLARE
  v1 NUMBER(38);
BEGIN
  v1 := 256*256*256*256.0;
  DBMS_OUTPUT.PUT_LINE(v1);
END;
/
```

See Also:

- The "What's New in SQL*Plus" section in the *SQL*Plus User's Guide and Reference* to learn about new features in SQL*Plus
- *Oracle Database SQL Language Reference* for more information about upgrading SQL scripts

Upgrading Oracle Forms or Oracle Developer Applications

Forms applications run the same on Oracle9*i*, Oracle Database 10*g*, and Oracle Database 11*g*. However, review the new features described in *Oracle Database New Features Guide* to determine whether any of the features of the new Oracle Database 11*g* release would be beneficial to your applications or might otherwise affect them. Information about the ways in which the features of the new Oracle Database 11*g* release interact with forms and developer applications is provided in the Oracle Developer documentation set. Also, the Oracle Developer documentation for your operating system contains instructions for upgrading your forms or developer applications.

Note: New releases of Oracle Developer might introduce new reserved words that are specific to Oracle Developer. Code changes might be required if your application uses any of these new reserved words.

6

Downgrading Oracle Database to an Earlier Release

Oracle Database can only be downgraded to the release from which you upgraded. For example, if you upgrade from Oracle Database 10*g* Release 1 (10.1.0.5) to Oracle Database 11*g* Release 2 (11.2), then you cannot subsequently downgrade to Oracle Database 10*g* Release 2 (10.2); you can only downgrade to Oracle Database 10*g* Release 1 (10.1.0.5).

This chapter contains the following topics:

> **See Also:** *Oracle Database Installation Guide* for your operating system for discussions of downgrading that are operating system-specific

Supported Releases for Downgrading Oracle Database

You can downgrade both major releases and patchset releases, based on the original release from which the database was upgraded. Major release downgrades are supported back to Oracle Database 11*g* Release 1 (11.1.0.6), Oracle Database 10*g* Release 2 (10.2.0.2), and Oracle Database 10*g* Release 1 (10.1.0.5). Patchset downgrades are supported back to all prior 11.2 patch releases.

> **Note:** You cannot downgrade a database that was upgraded from Oracle Database Express Edition.

If you have Oracle Database Vault installed with your Oracle Database 11*g* Release 2 (11.2) database, then you can downgrade to release 11.1.0.6 and release 11.1.0.7 only.

If you have Messaging Gateway or Workspace Manager in your database, then be aware that neither of them are part of Oracle Database patchsets earlier than release 10.2.0.4. Therefore, you must separately apply all relevant patches to the release 10.2.0.3 or release 10.1.0.5 Oracle home before downgrading.

Downgrade is not supported for Oracle Enterprise Manager. However, if you save your Oracle Enterprise Manager Database Control files and data before upgrading your database, then you can restore the old version of Database Control after downgrading the database.

See Also:

- "Saving Database Control Files and Data with the emdwgrd Utility" on page 3-20
- "Restoring Oracle Enterprise Manager after Downgrading Oracle Database" on page 6-10

Check for Incompatibilities Between Oracle Database Releases

Check the compatibility level of your release of Oracle Database to see if the database might have incompatibilities that prevent you from downgrading. If the compatibility level of your Oracle Database 11*g* Release 2 (11.2) database is `11.2.0` or higher, then you cannot downgrade.

See Also: "Checking the Compatibility Level of Oracle Database" on page 1-9

If you are downgrading to Oracle Database 11*g* Release 1 (11.1), then the `COMPATIBLE` initialization parameter must be set to `11.0.0` or lower.

If you are downgrading to Oracle Database 10*g* Release 2 (10.2), then the `COMPATIBLE` initialization parameter must be set to `10.2.0` or lower.

If you are downgrading to Oracle Database 10*g* Release 1 (10.1), then the `COMPATIBLE` initialization parameter must be set to `10.1.0`.

See Also: "Considerations for Downgrading Oracle Database and Compatibility" on page 1-8 and Appendix A, "Behavior Changes After Upgrading Oracle Database"

Perform a Full Backup of Oracle Database Before Downgrading

Perform a full backup of your Oracle Database 11*g* Release 2 (11.2) database before you downgrade. Follow the procedures given in *Oracle Database Backup and Recovery User's Guide*.

See Also: *Oracle Database Backup and Recovery User's Guide* for more information

Downgrading Oracle Database to an Earlier Release

You can downgrade your Oracle Database 11*g* Release 2 (11.2) database to a major release or a relevant patchset upgrade.

To downgrade the database to an earlier release

1. If you have enabled Oracle Database Vault on your database, then you must:
   - Grant the Database Vault `DV_PATCH_ADMIN` role for the `SYS` account.
   - Disable Database Vault before downgrading the database.

> **See Also:** *Oracle Database Vault Administrator's Guide* for instructions about disabling Oracle Database Vault

2. If you previously had upgraded the database and then used the DBMS_DST PL/SQL package to update the database time zone version, then you must apply the patch for the same time zone file version into the earlier release's Oracle home before downgrading.

   As an example scenario, assume that a release 10.2.0.4 database on Linux x64 using DSTv4 had been upgraded to release 11.2.0.2, and DBMS_DST was then run to update this database to DSTv14. Then, before downgrading from release 11.2.0.3 to 10.2.0.4, you need to apply on the release 10.2.0.4 side the DSTv14 patch for 10.2.0.4 for Linux x64. This ensures that your TIMESTAMP WITH TIME ZONE data is not logically corrupted during retrieval.

   To find which time zone file version your database is using, run:

   ```
   SELECT value$ FROM sys.props$ WHERE NAME = 'DST_PRIMARY_TT_VERSION';
   ```

   > **See Also:** *Oracle Database Globalization Support Guide* for information on applying time zone files

3. If you had set the ORA_TZFILE environment variable to the full path name of the timezone.dat file when you upgraded to Oracle Database 11*g* Release 2 (11.2), then you must unset it if you subsequently downgrade your database.

   Two time zone files are included in the Oracle home directory:

   - The default time zone file at

     ```
     $ORACLE_HOME/oracore/zoneinfo/timezonelrg.dat
     ```

   - A smaller time zone file at

     ```
     $ORACLE_HOME/oracore/zoneinfo/timezone.dat
     ```

   If you do not unset the ORA_TZFILE variable, then connecting to the database using the smaller time zone file might produce the following errors:

   ```
   SP2-1503: Unable to initialize Oracle call interface
   SP2-0152: ORACLE may not be functioning properly
   ```

   > **See Also:** *Oracle Database Globalization Support Guide* for more information about date and time data types and time zone support

4. If you have Oracle Application Express on your database, then you must copy the apxrelod.sql file from the Oracle Database 11*g* Release 2 (11.2) ORACLE_HOME/apex/ directory to a directory outside of the Oracle home, such as the temporary directory on your system.

   Make a note of the new location of this file.

5. If you have created objects based on fixed objects, then drop these objects to avoid possible ORA-00600 errors. You can re-create these objects after the downgrade.

6. Log in to the system as the owner of the Oracle Database 11*g* Release 2 (11.2) Oracle home directory.

7. This step is required only if Enterprise Manager Database Control is already configured for the database.

Stop Database Control, as follows:

a. Set the ORACLE_UNQNAME environment variable to the database unique name.

b. Run the following command:

```
ORACLE_HOME/bin/emctl stop dbconsole
```

If the database being downgraded is an Oracle RAC database, then perform this step on all instances.

8. If you are downgrading an Oracle RAC database to 10*g* Release 1 (10.1), then you must remove extra voting disks before shutting down the Oracle Clusterware stack.

a. To see the number of voting disks used and to list voting disk paths, run the following command:

```
Oracle_Clusterware_Home/bin/crsctl query css votedisk
```

b. Remove each additional voting disk you find in the previous step by running the following command, where *path* is the voting disk path listed in the File Name column displayed in the previous step:

```
Oracle_Clusterware_Home/bin/crsctl delete css votedisk path
```

For example:

```
Oracle_Clusterware_Home/bin/crsctl delete css votedisk
/share/cluster2/vote_cluster2.dbf
```

Do not delete the last voting disk.

9. At a system prompt, change to the *ORACLE_HOME*/rdbms/admin directory.

> **Note:** If you are downgrading a cluster database, then shut down the instance completely and change the CLUSTER_DATABASE initialization parameter to FALSE. After the downgrade, you must set this parameter back to TRUE.

10. Using SQL*Plus, connect to the database instance as a user with SYSDBA privileges.

11. Start up the instance in DOWNGRADE mode:

```
SQL> STARTUP DOWNGRADE
```

You might be required to use the PFILE option to specify the location of your initialization parameter file.

12. If you have Enterprise Manager configured in your database, then drop the Enterprise Manager user:

```
DROP USER sysman CASCADE;
```

> **Note:** After this step, MGMT* synonyms may be invalid. Follow the guidelines in "Restoring Oracle Enterprise Manager after Downgrading Oracle Database" on page 6-10 to validate the synonyms.

13. Set the system to spool results to a log file so you can track the changes and issues:

    ```
    SQL> SPOOL downgrade.log
    ```

14. Run `catdwgrd.sql`:

    ```
    SQL> @catdwgrd.sql
    ```

    The following are notes about running the script:

    - You must use the version of the script included with Oracle Database 11*g* Release 2 (11.2).
    - You must run the script in the Oracle Database 11*g* Release 2 (11.2) environment.
    - The script downgrades all Oracle Database components in the database to the major release or Oracle Database 11*g* patch release from which you originally upgraded.

    If you encounter any problems when you run the script, or any of the scripts in the remaining steps, then correct the causes of the problems and rerun the script. You can rerun any of the scripts described in this chapter as many times as necessary.

    If the downgrade for a component fails, then an `ORA-39709` error is displayed and the SQL*Plus session terminates without downgrading the Oracle Database data dictionary. All components must be successfully downgraded before the Oracle Database data dictionary is downgraded. You must identify and fix the problem before rerunning the `catdwgrd.sql` script.

15. Turn off the spooling of script results to the log file:

    ```
    SQL> SPOOL OFF
    ```

    Then, check the spool file and verify that there were no errors generated during the downgrade. You named the spool file in Step 13 and the suggested name was `downgrade.log`. Correct any problems you find in this file and rerun the downgrade script if necessary.

    > **Note:** If you want to save the results of the first time you ran the downgrade script, then before you rerun it be sure to rename `downgrade.log` to something else.

16. Shut down the instance:

    ```
    SQL> SHUTDOWN IMMEDIATE
    ```

17. Exit SQL*Plus.

18. If your operating system is Linux or UNIX, then change the following environment variables to point to the directories of the release to which you are downgrading:

    - `ORACLE_HOME`
    - `PATH`

    You should also check that your `oratab` file and any client scripts that set the value of `ORACLE_HOME` point to the downgraded Oracle home.

See Also: *Oracle Database Installation Guide* for your operating system for information about setting other important environment variables on your operating system

19. If your operating system is Windows, then complete the following steps:

    a. Stop all Oracle services, including the `OracleService`*`SID`* Oracle service of the Oracle Database 11*g* Release 2 (11.2) database, where *`SID`* is the instance name.

    For example, if your *SID* is ORCL, then enter the following at a command prompt:

```
C:\> NET STOP OracleServiceORCL
```

See Also: *Oracle Database Net Services Administrator's Guide*

    b. Delete the Oracle service at a command prompt by issuing the `ORADIM` command. For example, if your *SID* is ORCL, then enter the following command:

```
C:\> ORADIM -DELETE -SID ORCL
```

    c. Create the Oracle service of the database that you are downgrading at a command prompt using the `ORADIM` command.

```
C:\> ORADIM -NEW -SID SID -INTPWD PASSWORD -MAXUSERS USERS
     -STARTMODE AUTO -PFILE ORACLE_HOME\DATABASE\INITSID.ORA
```

This syntax includes the following variables:

Variable	Description
`SID`	Same SID name as the SID of the database being downgraded.
`PASSWORD`	Password for the database instance. This is the password for the user connected with `SYSDBA` privileges. The `-INTPWD` option is not required. If you do not specify it, then operating system authentication is used, and no password is required.
`USERS`	Maximum number of users who can be granted `SYSDBA` and `SYSOPER` privileges.
`ORACLE_HOME`	Oracle home directory of the database to which you are downgrading. Ensure that you specify the full path name with the `-PFILE` option, including drive letter of the Oracle home directory.

For example, if you are downgrading to Oracle Database 10*g* Release 2 (10.2), if your *SID* is `ORCL`, your *PASSWORD* is `TWxy5791`, the maximum number of *USERS* is 10, and the *ORACLE_HOME* directory is C:\ORANT, then enter the following command:

```
C:\> ORADIM -NEW -SID ORCL -INTPWD TWxy5791 -MAXUSERS 10
     -STARTMODE AUTO -PFILE C:\ORANT\DATABASE\INITORCL.ORA
```

Note: You are not required to change any Windows Registry settings when downgrading a database. The `ORADIM` utility makes all necessary changes automatically.

20. Restore the configuration files (for example, parameter files, password files, and so on) of the release to which you are downgrading.

    If this is an Oracle RAC database, execute the following command to return the database to single instance mode:

    ```
    SET CLUSTER_DATABASE=FALSE
    ```

    > **Note:** If you are downgrading a cluster database, then perform this step on all nodes in which this cluster database has instances configured. Set the `CLUSTER_DATABASE` initialization parameter to `FALSE`. After the downgrade, you must set this initialization parameter back to `TRUE`.

21. At a system prompt, change to the *`ORACLE_HOME`*`/rdbms/admin` directory of the previous release.

22. Start SQL*Plus.

23. Connect to the database instance as a user with `SYSDBA` privileges.

24. Start up the instance:

    ```
    SQL> STARTUP UPGRADE
    ```

25. Set the system to spool results to a log file to track changes and issues:

    ```
    SQL> SPOOL reload.log
    ```

26. Run `catrelod.sql`:

    ```
    SQL> @catrelod.sql
    ```

    The `catrelod.sql` script reloads the appropriate version of all of the database components in the downgraded database.

27. If you are downgrading to Oracle Database 11*g* Release 1 (11.1.0.6), run the `xsrelod.sql` script:

    ```
    SQL> @xsrelod.sql
    ```

    Running the `xsrelod.sql` script avoids the following error:

    ```
    PLS-00306: wrong number or types of arguments in call
    to 'INVALIDATE_DSD_CACHE' DBMS_XS_DATA_SECURITY_EVENTS
    PL/SQL: Statement ignored
    ```

    > **Note:** The PLS-00306 error is not an issue for Oracle Database release 11.2.0.7 or later.

28. If you are downgrading to Oracle Database 10g Release 1 (10.1.0.5) and you have XDB in your database, then run the `dbmsxdbt.sql` script:

    ```
    @dbmsxdbt.sql
    ```

    Running the `dbmsxdbt.sql` script avoids the following errors when reloading `XDB.DBMS_XDBT` package:

    ```
    PLS-00201: identifier 'CTXSYS.CTX_OUTPUT' must be declared
    PLS-00201: identifier 'CTX_DDL' must be declared
    ```

29. If you are downgrading to Oracle Database 11*g* Release 1 or earlier, and you have Oracle Application Express in your database, then change to the directory to which you had copied the `apxrelod.sql` script (in step 4). Manually reload Oracle Application Express by running the `apxrelod.sql` script:

    ```
    SQL> @apxrelod.sql
    ```

    Running the `apxrelod.sql` script avoids package `APEX_030200.WWV_FLOW_HELP` being `INVALID` due to the following error:

    ```
    PLS-00201: identifier 'CTX_DDL' must be declared
    ```

30. Turn off the spooling of script results to the log file:

    ```
    SQL> SPOOL OFF
    ```

    Then, check the spool file and verify that the packages and procedures compiled successfully. You named the spool file in Step 25; the suggested name was `reload.log`. Correct any problems you find in this file and rerun the appropriate script if necessary.

31. Shut down and restart the instance for normal operation:

    ```
    SQL> SHUTDOWN IMMEDIATE
    SQL> STARTUP
    ```

    You might be required to use the `PFILE` option to specify the location of your initialization parameter file.

32. Perform this step if the database is configured for Oracle Label Security and you are downgrading to Oracle Database 10*g* Release 1 (10.1).

    a. Copy the `olstrig.sql` script from the Oracle Database 11*g* Release 2 (11.2) Oracle home to the Oracle home of the version to which the database is to be downgraded.

    b. Run `olstrig.sql` to re-create DML triggers on tables with Oracle Label Security policies.

       ```
       SQL> @olstrig.sql
       ```

    See the *Oracle Label Security Administrator's Guide* for more information.

33. Run the `utlrp.sql` script:

    ```
    SQL> @utlrp.sql
    ```

    The `utlrp.sql` script recompiles all existing PL/SQL modules that were previously in an `INVALID` state, such as packages, procedures, types, and so on.

34. Exit SQL*Plus.

35. If you are downgrading a cluster database, then you must run the following command to downgrade the Oracle Clusterware database configuration:

    ```
    $ srvctl downgrade database -d db-unique-name -o old_ORACLE_HOME -t to_relnum
    ```

    [where *`db-unique-name`* is the database name (not the instance name), *`old_ORACLE_HOME`* is the location of the old Oracle home in which the downgraded database will be running. In this example, *`to_relnum`* is the database release number to which the database is being downgraded. (For example: 11.2.0.1.0.)]

Caution: By default, any named user may create a server pool. To restrict the operating system users that have this privilege, Oracle strongly recommends that you add specific users to the CRS Administrators list.

See Also: *Oracle Clusterware Administration and Deployment Guide* for more information about adding users to the CRS Administrators list

Note: Execute this command from the current Oracle home, not from the Oracle home to which the database is being downgraded.

Your database is now downgraded.

Post-Downgrade Tasks for Oracle Database

Tasks might be required after downgrading Oracle Database. This section contains the following topics:

- Re-creating the Network Listener When Downgrading to Oracle Database Releases 10.2 or 11.1
- Restoring Oracle Enterprise Manager after Downgrading Oracle Database
  - Single-Instance Oracle Database Without Oracle ASM
  - Oracle RAC Database Without Oracle ASM
  - Single-Instance Oracle ASM Instance
  - Oracle ASM on Oracle RAC Instance
  - Single-Instance Oracle Database With Oracle ASM
  - Oracle RAC Database and Oracle ASM Instance
- Re-enabling Oracle Database Vault after Downgrading Oracle Database
- Restoration of the Configuration for Oracle Clusterware after Downgrading

Re-creating the Network Listener When Downgrading to Oracle Database Releases 10.2 or 11.1

If you are downgrading the new database to Oracle Database 10*g* or Oracle Database 11*g* Release 1, then you must recreate the listener after performing the downgrade steps. Run Oracle Net Manager to delete the old listener and create a new listener.

- To start Oracle Net Manager from the Oracle Enterprise Manager console, select Service Management from the Tools menu, and then select Oracle Net Manager.
- On Microsoft Windows, select **Programs** from the Start menu, and then select **Oracle - *HOME_NAME***. Next, select **Configuration and Migration Tools**, and then **Net Manager**.

See Also: *Oracle Database Net Services Administrator's Guide*

Restoring Oracle Enterprise Manager after Downgrading Oracle Database

This task is required only if you are downgrading in some form and Oracle Enterprise Manager is configured on the host. To restore Oracle Enterprise Manager to its previous state, you must have saved your Oracle Enterprise Manager files and data before upgrading.

> **See Also:** "Saving Database Control Files and Data with the emdwgrd Utility" on page 3-20

If this is an Oracle RAC database using Oracle Clusterware, the database must be registered with the `srvctl` before running the `emca -restore` command. This needs to be executed from the *`ORACLE_HOME`*/bin of the version to which the database is being downgraded.

> **See Also:** *Oracle Clusterware Administration and Deployment Guide* for information about `srvctl`

Run the `emca -restore` command with the appropriate options to restore Oracle Enterprise Manager Database Control or Grid Control to the old Oracle home. The options that you specify depend on whether the database being downgraded is an Oracle RAC database or an Oracle ASM database, as follows:

> **Note:** Use the Oracle Database 11*g* Release 2 (11.2) version of `emca` for this procedure. See *Oracle Database Administrator's Guide* for complete information about `emca`.

Single-Instance Oracle Database Without Oracle ASM

```
111Home/bin/emca -restore db
```

You are prompted to enter the following information:

- Oracle home for the database to be restored
- Database SID
- Listener port number

Oracle RAC Database Without Oracle ASM

```
111Home/bin/emca -restore db -cluster
```

You are prompted to enter the following information:

- Oracle home for the database to be restored
- Database unique name
- Listener port number

Single-Instance Oracle ASM Instance

```
111Home/bin/emca -restore asm
```

You are prompted to enter the following information:

- Oracle home for the database to be restored
- Oracle ASM port
- Oracle ASM SID

Oracle ASM on Oracle RAC Instance

```
111Home/bin/emca -restore asm -cluster
```

You are prompted to enter the following information:

- Oracle home for the database to be restored
- Oracle ASM port

Single-Instance Oracle Database With Oracle ASM

```
111Home/bin/emca -restore db_asm
```

You are prompted to enter the following information:

- Oracle home for the database to be restored
- Database SID
- Listener port number
- Oracle ASM port
- Oracle ASM home
- Oracle ASM SID [+ASM]

Oracle RAC Database and Oracle ASM Instance

```
111Home/bin/emca -restore db_asm -cluster
```

You are prompted to enter the following information:

- Oracle home for the database to be restored
- Database unique name
- Listener port number
- Oracle ASM port
- Oracle ASM Oracle home
- Oracle ASM SID [+ASM]

The output of `emca` varies according to the options that you specify and the values that you enter at the prompts.

For an inplace patchset upgrade, the original home backup you saved before applying the patchset must be restored. This step must be repeated on all the nodes in Oracle RAC environments. Running the `emca -restore` command is not required for an inplace patchset upgrade.

After completing the `emca -restore` procedure you are ready to restore the Oracle Enterprise Manager Database files and data with the `emdwgrd` utility. You must run the emdwgrd utility from the Oracle Database 11*g* Release 2 (11.2) home. But `ORACLE_HOME` and other environment variables must be set to point to the Oracle home from which the upgrade originally took place.

The following procedure is for Linux and UNIX. To run it on Windows, simply substitute `emdwgrd.bat` for `emdwgrd`.

Follow these steps to restore your database control files and data:

1. Set `ORACLE_HOME` to the Oracle home from which the database upgrade originally took place.

For an inplace patchset upgrade, restore the backup of the original home (with one-off patch of Bug 7131048 if upgrading from 11.1.0.6), and set `ORACLE_HOME` to the current Oracle home.

2. Set `ORACLE_SID` to the SID of the database that was upgraded and then downgraded.
3. Set `PATH`, `LD_LIBRARY_PATH` and `SHLIB_PATH` to point to the Oracle home from which the database upgrade originally took place.
4. Go to the Oracle Database 11*g* Release 2 (11.2) home:

   ```
   cd ORACLE_HOME/bin
   ```

5. Execute one of the following:

   a. For a single-instance database, run the following command, where *`SID`* is the SID of the database that was upgraded and then downgraded and *`save_directory`* is the path to the storage location you chose when saving your database control files and data:

   ```
   emdwgrd -restore -sid SID -path save_sirectory -tempTablespace TEMP
   ```

   b. For an Oracle RAC database, remote copy is required across the cluster nodes. Define an environment variable to indicate which remote copy is configured. For example:

   ```
   setenv EM_REMCP /usr/bin/scp
   ```

   Then, execute the following `restore` command:

   ```
   emdwgrd -restore -tempTablespace TEMP -cluster -sid SID10g -path save_
   directory
   ```

   If 10*g* Oracle home is on a shared device, add `-shared` to the previous command line.

6. Enter the SYS and SYSMAN passwords when prompted by `emdwgrd`.
7. On a single-instance database, the `emdwgrd` utility produces output similar to the following:

   ```
   Sat Apr 28 09:27:09 2007 - Verify EM DB Control files ... pass
   Sat Apr 28 09:27:09 2007 - Validating DB Connection to DB102 ... pass
   Sat Apr 28 09:27:19 2007 - Validating TEMP tablespace in DB102 ... pass
   Sat Apr 28 09:27:20 2007 - creating directory ... created
   Sat Apr 28 09:27:20 2007 - Stopping DB Control ... stopped
   Sat Apr 28 09:27:21 2007 - dropping sysman schema ... dropped
   Sat Apr 28 09:27:22 2007 - recreating sysman user ... recreated
   Sat Apr 28 09:27:23 2007 - Restoring DB Control files ... restored
   Sat Apr 28 09:27:23 2007 - Importing sysman schema ... imported
   Sat Apr 28 09:30:42 2007 - recompiling invalid objects ... recompiled
   Sat Apr 28 09:30:54 2007 - Starting DB Control ... started
   Sat Apr 28 09:32:37 2007 - DB Control was restored successfully.
   ```

   When `emdwgrd` finishes, Oracle Enterprise Manager Database Control has been downgraded to the old Oracle home.

   On an Oracle RAC database, the `emdwgrd` utility produces output similar to the following:

   ```
   $ /scratch/oracle/product/11.1.0/db_1/bin/emdwgrd -srcOracleHome $ORACLE_HOME
   -sid DB102 -path /scratch/rpattabh/ravi/tmp/dbcdir5 -restore -cluster
   -tempTablespace TEMP
   ```

```
Enter sys password for database DB102?
*****

Enter sysman password for database DB102?
*****

Sat Apr 28 09:27:09 2007 - Verify EM DB Control files ... pass
Sat Apr 28 09:27:09 2007 - Validating DB Connection to DB102 ... pass
Sat Apr 28 09:27:19 2007 - Validating TEMP tablespace in DB102 ... pass
Sat Apr 28 09:27:20 2007 - creating directory ... created
Sat Apr 28 09:27:20 2007 - Stopping DB Control on all Nodes
stbdq04, r101b1, /oradbnas/sangeeta/10.1.0/db, stop, 0
stbdq05, r101b2, /oradbnas/sangeeta/10.1.0/db, stop, 1

Please Execute '/tmp/racdwgrd_dbctl.sh' on Node1, Node2.

Press yes to continue when the operations are successful.
Continue (yes/no) ?
y

... stopped
Sat Apr 28 09:27:21 2007 - dropping sysman schema ... dropped
Sat Apr 28 09:27:22 2007 - recreating sysman user ... recreated
Sat Apr 28 09:27:23 2007 - Restoring DB Control files
Executing Restore directories to node Node1
Executing Restore directories to node Node2

... restored
Sat Apr 28 09:27:23 2007 - Importing sysman schema ... imported
Sat Apr 28 09:30:42 2007 - recompiling invalid objects ... recompiled
Sat Apr 28 09:32:37 2007 - DB Control was restored successfully.
Sat Apr 28 09:33:54 2007 - Starting DB Control On All nodes

Please Execute '/tmp/racdwgrd_dbctl.sh' on Node1, Node2.

Press yes to continue when the operations are successful.
Continue (yes/no) ?
y

... started
Sat Apr 28 09:38:57 2007 - Dump directory was dropped successfully.
```

Re-enabling Oracle Database Vault after Downgrading Oracle Database

If you use Oracle Database Vault, then you were instructed to disable it before downgrading your database. To use Oracle Database Vault after downgrading, you must re-enable it.

To enable Oracle Database Vault after downgrading:

1. Connect to SQL*Plus as a user who has been granted the `DV_OWNER` role.
2. Issue the following statements:

```
ALTER TRIGGER DVSYS.DV_BEFORE_DDL_TRG ENABLE;
ALTER TRIGGER DVSYS.DV_AFTER_DDL_TRG ENABLE;
```

See Also: *Oracle Database Vault Administrator's Guide*

Restoration of the Configuration for Oracle Clusterware after Downgrading

You can restore the Oracle Clusterware configuration to the state it was in before the Oracle Clusterware 11*g* release 2 (11.2) upgrade. To do this, you must restore the release from which you were upgrading. Any configuration changes you have performed during or after the 11*g* release 2 (11.2) upgrade process are removed and cannot be recovered.

See Also:

- *Oracle Grid Infrastructure Installation Guide for Linux*
- *Oracle Grid Infrastructure Installation Guide for Microsoft Windows (32-Bit), Microsoft Windows (64-Bit) on Intel Itanium, Microsoft Windows x64*
- *Oracle Clusterware Administration and Deployment Guide*
- *Oracle Database Storage Administrator's Guide*

Troubleshooting the Downgrade of Oracle Database

Oracle provides troubleshooting information for issues that may occur when downgrading Oracle Database. For the latest information on this release, also refer to *Oracle Database Readme*.

Downgrading release 11.2 of Oracle Grid Infrastructure fails

If downgrading release 11.2 of Oracle Grid Infrastructure fails, then refer to the support note "How to Proceed from Failed Upgrade to Grid Infrastructure" on Linux/Unix [ID 969254.1]" on `https://support.oracle.com`. Oracle recommends that you read the entire note before taking action.

Error raised from the 11.2.0.1 xsrelod.sql script

When downgrading Oracle Database from release 11.2.0.2 or 11.2.0.3 to release 11.2.0.1, an error may be raised from the `xsrelod.sql` script, which is included with release 11.2.0.1. This problem only occurs after downgrading to release 11.2.0.1 and then attempting to enable XML DB indexes. The error "ORA-01418: specified index does not exist" may be raised and can safely be ignored.

Error raised from the Oracle Multimedia imrelod.sql script

When downgrading Oracle Database to release 11.2.0.2, an error may be raised from the `imrelod.sql` script which is included with release 11.2.0.2. The error "ORA-20000: Oracle ORDIM component in registry is status: DOWNGRADED. Oracle ORDIM must be installed and valid prior to Oracle Multimedia upgrade, downgrade, or patch." can safely be ignored.

Downgrading Oracle Database causes invalid object CTX_FILTER_CACHE_STATISTICS

After downgrading Oracle Database, the following invalid Text object may be seen:

```
CTX_FILTER_CACHE_STATISTICS (synonym)
CTX_FILTER_CACHE_STATISTICS (view)
```

To fix this problem, in the current release of Oracle Database, after running the `catdwgrd.sql` script and before running the `catrelod.sql` script, issue the following two commands:

```
SQL> drop public synonym ctx_filter_cache_statistics;
SQL> drop view ctx_filter_cache_statistics;
```

7 Moving Data Using Oracle Data Pump

When upgrading Oracle Database, you can use the Export and Import utilities in Oracle Data Pump to move data from one database to another.

This chapter contains the following topics:

- About Data Pump Export and Import for Upgrading Oracle Database
- Export and Import Requirements for Oracle Database Upgrades
- Upgrading the Database Using Data Pump Export/Import

> **See Also:** *Oracle Database Utilities* for detailed information about Data Pump and the Export and Import utilities

About Data Pump Export and Import for Upgrading Oracle Database

The Data Pump Export (expdp) and Import (impdp) utilities have a similar look and feel to the original Export (exp) and Import (imp) utilities, but they are completely separate. The Data Pump Export and Import utilities, first released in Oracle Database 10*g* Release 1 (10.1), are high performance replacements for the original Export and Import utilities. When upgrading your database, Oracle recommends the Data Pump Export and Import utilities because these utilities provide greatly enhanced performance compared to the original Export and Import utilities.

> **See Also:**
>
> - "Oracle Data Pump Export and Import and Oracle Database Upgrade" on page 2-4 for information on when to use this upgrade method
> - *Oracle Database Utilities* for a detailed list of Data Pump features that contribute to increased performance, as well as to enhanced ease-of-use

Data Pump Export and Import offer the following advantages:

- Provides greater data and metadata filtering capability than was provided by the original Export utility. Data Pump supports filtering the metadata that is exported and imported based upon objects and object types, using `INCLUDE` and `EXCLUDE` parameters.
- Supports different modes for unloading/loading portions of the database including: full database mode, schema mode, table mode, tablespace mode, and transportable tablespace mode. (See the "Data Pump Export Modes" and "Data Pump Import Modes" sections in *Oracle Database Utilities*)

- Enables you to specify how partitioned tables should be handled during import operations, using the `PARTITION_OPTIONS` parameter.
- Provides support for the full range of data types.

> **See Also:** *Oracle Database Utilities* for an overview of Data Pump Export and Import

The original Export/Import utilities are required for the following types of database upgrades and downgrades:

- If you are upgrading from a release earlier than Oracle Database 10*g* Release 1 (10.1), then you must use the original Export/Import utilities to move the data. Likewise, if you are downgrading your database to a release earlier than Oracle Database 10*g* Release 1 (10.1), then you must use the original Export/Import utilities.
- If you must downgrade to a release earlier than Oracle Database 10*g* Release 1 (10.1), then the original Export utility continues to provide support to ensure backward compatibility.
- If you are moving tables of XMLType or tables containing XMLType columns (schema and non-schema-based) from Oracle Database 11*g* Release 1 (11.1) to Oracle Database 10*g* Release 2 (10.2).

> **See Also:** *Oracle Database Utilities* for information on moving a database from one platform to another

Export and Import Requirements for Oracle Database Upgrades

Dump files generated by the Data Pump Export utility are not compatible with dump files generated by the original Export utility. Therefore, files generated by the original Export (exp) utility cannot be imported with the Data Pump Import (impdp) utility, and vice versa.

About Export Dump Files and Upgrading

Dump files created by one of the Export utilities can be imported into all future releases of Oracle Database. For example, an Oracle8 original Export dump file can be imported by the Oracle8*i*, Oracle9*i*, Oracle Database 10*g*, and Oracle Database 11*g* original Import utility.

Table 7–1 shows which releases to use when exporting data from releases earlier than Oracle Database 11*g* Release 2 (11.2) and then importing that data into Oracle Database 11*g* Release 2 (11.2).

Table 7–1 Exporting Data From Releases Earlier Than 11.2 and Importing Into Release 11.2

Export From	Import To	Export Version to Use	Import Version to Use
Release 11.1	Release 11.2	Data Pump Export Release 11.1	Data Pump Import Release 11.2
Release 10.2	Release 11.2	Data Pump Export Release 10.2	Data Pump Import Release 11.2
Release 10.1	Release 11.2	Data Pump Export Release 10.1	Data Pump Import Release 11.2
Release 9.2	Release 11.2	Original Export Release 9.2	Original Import Release 11.2

Table 7–1 (Cont.) Exporting Data From Releases Earlier Than 11.2 and Importing Into Release 11.2

Export From	Import To	Export Version to Use	Import Version to Use
Release 8.1.7	Release 11.2	Original Export Release 8.1.7	Original Import Release 11.2
Release 8.0.6	Release 11.2	Original Export Release 8.0.6	Original Import Release 11.2
Release 7.3.4	Release 11.2	Original Export Release 7.3.4	Original Import Release 11.2

Export and Import Requirements for Downgrades

The Oracle Database release to which you downgrade can be one release older, at the most. For example, if the current database is Oracle Database 11*g* Release 1 (11.1), then Oracle Database 10*g* Release 2 (10.2) is the only release supported for the downgrade.

To obtain a downward compatible dump file:

- With Data Pump Export, use the *current* version's Data Pump Export utility, but set the `VERSION` parameter to the target older version.

  Data Pump Import cannot read dump file sets created by a database version that is newer than the current database version, unless those dump file sets were created with the `VERSION` parameter set to the version of the target database. Therefore, the best way to perform a downgrade is to use Data Pump Export with the `VERSION` parameter set to the version of the target database.

  > **See Also:** *Oracle Database Utilities* for more information about using the `VERSION` parameter

- With original Export, run an older version of Export (exp) to produce a dump file that is compatible with the database version to which you want to downgrade. Then, run the target's original Import utility.

  Original Export dump files are *not* downward compatible with the Import utilities of previous Oracle Database releases. That is, exported data *cannot* be imported by the Import utilities of previous Oracle Database releases. For example, an Oracle9*i* Export dump file cannot be imported by an Oracle8*i* Import utility, and an Oracle Database 10*g* export dump file cannot be imported by an Oracle9*i* Import utility. The following tables provide specific examples.

  > **See Also:** *Oracle Database Utilities* for more information about using different releases and versions of Export

Table 7–2 shows which releases to use when exporting data from Oracle Database 11*g* Release 2 (11.2) and then importing that data into earlier releases. Major release downgrades are supported to 11.1, 10.2, and 10.1.

Table 7–2 Exporting Data From Release 11.1 and Importing Into Earlier Releases

Export From	Import To	Export Version to Use	Import Version to Use
Release 11.2	Release 11.1	Data Pump Export Release 11.2 with `VERSION=11.1`	Data Pump Import Release 11.1
Release 11.1	Release 10.2	Data Pump Export Release 11.1 with `VERSION=10.2`	Data Pump Import Release 10.2
Release 10.2	Release 10.1	Data Pump Export Release 10.2 with `VERSION=10.1`	Data Pump Import Release 10.1

Note: When using the original Export utility, if the source database is newer than the target database, then you must run the `catexp.sql` script supplied with the previous (older) release for the export to be successful. Once the export is done, then run the `catexp.sql` script from the newer release to restore the export views. Perform the steps as follows:

1. Run the older `CATEXP.SQL` script on the database to be exported.
2. Use the older Export utility to create the dump file.
3. Use the older Import utility to import to the target database.
4. Run the newer `CATEXP.SQL` script on the exported database.

Export/Import Usage on Data Incompatible with a Previous Release

When you export data to a previous release, data that is incompatible with the previous release either is not exported at all or is exported with the loss of some features.

In general, if you must export data to a previous release, then first remove as many incompatibilities with the previous release as possible before you export the data.

Upgrading the Database Using Data Pump Export/Import

To upgrade a database using the Data Pump Export/Import utilities, complete the following steps:

1. Export data from the current database using the Export utility shipped with the current database. See the current *Oracle Database Utilities* documentation for information about using the Export utility on the current database.

   To ensure a consistent export, make sure the current database is not available for updates during and after the export. If the current database is available to users for updates after the export, then, before making the current database available, put procedures in place to copy the changes made in the current database to the new database after the import is complete.

2. Install the new Oracle Database software. Installation is operating-system specific. Installation steps for Oracle Database are provided in *Oracle Database Installation Guide* for your operating system.

3. If the new database has the same name as the current database, then shut down the current database before creating the new database.

4. Create the new database.

   See Also: *Oracle Database Administrator's Guide* for information about creating a database

5. Start SQL*Plus in the new Oracle Database environment.

6. Connect to the database instance as a user with `SYSDBA` privileges.

7. Start an Oracle Database instance using `STARTUP`.

8. Optionally, you can change the storage parameters from the source database.

   You can pre-create tablespaces, users, and tables in the new database to improve space usage by changing storage parameters. When you pre-create tables using SQL*Plus, either run the database in the original database compatibility mode or make allowances for the specific data definition conversions that occur during import. When items have been pre-created, specify one of the following options:

- `TABLE_EXISTS_ACTION=APPEND` for Data Pump Import
- `IGNORE=Y` for original Import

> **Note:** If the new database is created on the same computer as the source database, and you do not want to overwrite the source database data files, then you must pre-create the tablespaces and specify one of the following options when you import:
>
> - `REUSE_DATAFILES=N` for Data Pump Import
>
>   Optionally, consider using the `REMAP_DATAFILE, REMAP_TABLESPACE` and `REMAP_TABLE` options so that references to the old names in the dump file set are remapped to new, non-colliding names.
> - `DESTROY=N` for original Import.

9. Use the Import utility of the new database to import the objects exported from the current database. Include one of the following parameters to save the informational and error messages from the import session to a file:

   - The `LOGFILE` parameter for Data Pump Import
   - The `LOG` parameter for original Import

   See Also: *Oracle Database Utilities* for a complete description of the Data Pump Import utility.

10. After the import, check the import log file for information about which imports of which objects completed successfully and, if there were failures, which failed.

    See Also: *Oracle Database Utilities* and the Oracle Database `README.doc` file for error handling information.

11. Use further Import scenarios (see *Oracle Database Utilities*) or SQL scripts that create the database's objects to clean up incomplete imports (or possibly to start an entirely new import).

    > **Note:** If a Data Pump Export or Import job encounters a fatal error, then the job can be restarted after the condition inducing the failure is corrected. The job then continues automatically from the point of failure.

12. If changes are made to the current database after the export, then make sure those changes are propagated to the new database before making it available to users. See Step 1 for more information.
13. Complete the procedures described in Chapter 4, "Post-Upgrade Tasks for Oracle Database".

Importing a Full Database Using a Network Link

As an alternative to the procedure in "Upgrading the Database Using Data Pump Export/Import" on page 7-4, you can use the Data Pump Import utility with a database link to perform a full database import from a source database to a destination database without intermediate dump files.

> **Note:** In order to avoid interoperability errors, ensure that the appropriate patchset has been applied to the database being upgraded. See My Oracle Support at `https://support.oracle.com` to obtain the latest patchsets.
>
> You may also refer to support note ID 4511371.8, which discusses ORA-6544 and ORA-4052 errors.

Follow these steps to use a network link with impdp:

1. Ensure that the exporting user at the source database has the `EXP_FULL_DATABASE` role.

   This user must be specified when you create the database link.
2. Ensure that the importing user at the destination database has the `IMP_FULL_DATABASE` role.
3. Create and test a database link between the source and destination databases.
4. Run the following command, where *`import_user`* is the username for the importing user, and *`db_link`* is the name of the database link owned by the exporting user:

   ```
   IMPDP import_user/password NETWORK_LINK=db_link FULL=Y;
   ```
5. A log file for the import operation is written to the DATA_PUMP_DIR directory. You can discover the location of this directory by running the following command:

   ```
   SQL> select * from dba_directories where DIRECTORY_NAME like 'DATA_PUMP_DIR';
   ```

> **Note:** XML objects are not exported from the source database.

> **Note:** The import operation re-creates users on the new destination server, and the creation date for `dba_users` shows the actual import date. The expiration date is updated to be `creation_date` + `password_life_time`. The parameters for dba_users on the new server are different than the dba_users parameters on the source server.

A

Behavior Changes After Upgrading Oracle Database

Important changes in behavior exist between Oracle9*i* Release 2 (9.2), Oracle Database 10*g* Release 1 (10.1), Oracle Database 10*g* Release 2 (10.2), Oracle Database 11*g* Release 1 (11.1), and Oracle Database 11*g* Release 2 (11.2). This appendix focuses on behavior changes that require a database administrator (also referred to as a DBA) to make an informed decision to minimize the risks that may be introduced by the changes. This appendix does not describe all changed behavior or new features in the new Oracle Database 11*g* release.

This appendix includes the following topics:

- Compatibility and Interoperability Issues in Oracle Database 11g Release 2 (11.2)
- Compatibility and Interoperability Issues in Oracle Database 11g Release 1 (11.1)
- Compatibility and Interoperability Issues in Oracle Database 10g Release 2 (10.2)
- Compatibility and Interoperability Issues in Oracle Database 10g Release 1 (10.1)

See Also:

- *Oracle Database New Features Guide* for a complete list of all new features introduced in Oracle Database 11*g* release
- The "What's New in Oracle Database Reference" section of *Oracle Database Reference* for a list of new initialization parameters, new static data dictionary views, and new dynamic performance views in Oracle Database 11*g* release

Note: Some of the initialization parameters listed in this appendix are operating system-specific. See your operating system-specific Oracle documentation for more information about these initialization parameters.

Compatibility and Interoperability Issues in Oracle Database 11*g* Release 2 (11.2)

The topics in this section describe compatibility and interoperability issues introduced in Oracle Database 11*g* Release 2 (11.2) and actions you can take to prevent problems resulting from these changes.

- Deprecation of Oracle Enterprise Manager Database Control

- Deprecation of SNMP Support in Oracle Net Listener
- Changes to PL/SQL Procedures
- Deprecated XML DB Constructs
- Cursor_sharing=similar Obsolete in Oracle Database
- Planned Desupport of Change Data Capture
- Initialization Parameters Deprecated in Oracle Database 11g Release 2 (11.2)
- Initialization Parameters Obsolete in Oracle Database 11g Release 2 (11.2)
- Static Data Dictionary Views Deprecated in Oracle Database 11g Release 2 (11.2)
- Dynamic Performance Views Deprecated in Oracle Database 11g Release 2 (11.2)
- Deprecated Features in Oracle Database 11g Release 2 (11.2)
- Changes to LOG_ARCHIVE_DEST_n Parameters

Deprecation of Oracle Enterprise Manager Database Control

Oracle Enterprise Manager Database Control is deprecated in Oracle Database 11*g* Release 2 (11.2), and will be desupported in the next major release of Oracle Database. Oracle will fully support Oracle Enterprise Manager Database Control for the life of Oracle Database 11*g* Release 2 (11.2), including for all patch sets, and through the end of Extended Support.

Deprecation of SNMP Support in Oracle Net Listener

Oracle is deprecating SNMP support in Oracle Net Listener in Oracle Database 11*g* Release 2 (11.2). Oracle recommends not using SNMP in new implementations.

See Also: Doc ID 1341834.1 at `https://support.oracle.com`

Changes to PL/SQL Procedures

The following PL/SQL procedures have been moved from package `DBMS_XDB` to package `DBMS_XDB_ADMIN` in release 11.2.0.3:

- `moveXDB_tablespace`
- `rebuildHierarchicalIndex`

See Also: *Oracle Database PL/SQL Packages and Types Reference*

JOB_QUEUE_PROCESSES Parameter and Scheduling Jobs

Starting with Oracle Database 11*g* Release 2 (11.2), setting `JOB_QUEUE_PROCESSES` to `0` causes both `DBMS_SCHEDULER` and `DBMS_JOB` jobs to not run. Previously, setting `JOB_QUEUE_PROCESSES` to `0` caused `DBMS_JOB` jobs to not run, but `DBMS_SCHEDULER` jobs were unaffected and would still run. The default value is `1000`.

Note that Oracle Database overrides the job queue setting to disable scheduler jobs during upgrade mode. Therefore, there is no need to change this setting when upgrading Oracle Database.

See Also: *Oracle Database Reference* for more information on this parameter

Deprecated XML DB Constructs

The following XML DB constructs have been deprecated in release 11.2.0.3:

- PL/SQL procedure `DBMS_XDB_ADMIN.createRepositoryXMLIndex`
- PL/SQL procedure `DBMS_XDB_ADMIN.XMLIndexAddPath`
- PL/SQL procedure `DBMS_XDB_ADMIN.XMLIndexRemovePath`
- PL/SQL procedure `DBMS_XDB_ADMIN.dropRepositoryXMLIndex`
- XML schema annotation (attribute) `csx:encodingType`
- XMLIndex index on CLOB portions of hybrid `XMLType storage`

> **See Also:** *Oracle XML DB Developer's Guide*

Cursor_sharing=similar Obsolete in Oracle Database

The `cursor_sharing=similar` parameter is deprecated in Oracle Database release 11.2.0.3. Use adaptive cursor sharing instead.

> **See Also:** *Oracle Database Performance Tuning Guide* for information about adaptive cursor sharing

Planned Desupport of Change Data Capture

Oracle Change Data Capture will not be supported in a future release of Oracle Database and will be replaced with Oracle GoldenGate. Therefore, Oracle strongly recommends that you use Oracle GoldenGate for new applications.

For Oracle Database 11*g* Release 2 (11.2), Change Data Capture continues to function as in earlier releases. If you are currently using Change Data Capture, then you will be able to continue to do so for the foreseeable future. However, Change Data Capture will not be further enhanced.

> **See Also:**
>
> `http://www.oracle.com/technetwork/middleware/goldengate/overview/index.html` on Oracle Technology Network for more information about Oracle GoldenGate

Initialization Parameters Deprecated in Oracle Database 11*g* Release 2 (11.2)

A deprecated parameter behaves the same way as a regular parameter, except that a warning message is displayed at instance startup if a deprecated parameter is specified in the parameter file. In addition, all deprecated parameters are logged to the alert log at instance startup.

To get a list of all initialization parameters that are specified as deprecated for the current database, issue the following SQL statement:

```
SQL> SELECT name FROM v$parameter
         WHERE isdeprecated = 'TRUE';
```

The following initialization parameters were deprecated in Oracle Database 11*g* Release 2 (11.2):

```
ACTIVE_INSTANCE_COUNT
PARALLEL_IO_CAP_ENABLED
```

Initialization Parameters Obsolete in Oracle Database 11*g* Release 2 (11.2)

The following initialization parameters were made obsolete in Oracle Database 11*g* Release 2 (11.2).

> **Note:** An attempt to start a database using one or more obsolete initialization parameters will succeed, but a warning is returned and recorded in the alert log.

```
DRS_START
GC_FILES_TO_LOCKS
MAX_COMMIT_PROPAGATION_DELAY
PLSQL_NATIVE_LIBRARY_DIR
PLSQL_NATIVE_LIBRARY_SUBDIR_COUNT
SQL_VERSION
```

Static Data Dictionary Views Deprecated in Oracle Database 11*g* Release 2 (11.2)

These static data dictionary views were deprecated in Oracle Database 11*g* Release 2 (11.2):

`ALL_STREAMS_STMTS` (replaced by `DBA_STREAMS_STMTS`)
`ALL_STREAMS_STMT_HANDLERS` (replaced by `DBA_STREAMS_STMT_HANDLERS`)
`DBA_COMPARISON_SCAN_SUMMARY` (replaced by `DBA_COMPARISON_SCAN`)
`USER_COMPARISON_SCAN_SUMMARY` (replaced by `USER_COMPARISON_SCAN`)

Dynamic Performance Views Deprecated in Oracle Database 11*g* Release 2 (11.2)

These dynamic performance views were deprecated in Oracle Database 11*g* Release 2 (11.2).

`V$FLASH_RECOVERY_AREA_USAGE` (replaced by `V$RECOVERY_AREA_USAGE`)

Deprecated Features in Oracle Database 11*g* Release 2 (11.2)

These Oracle Database features have been deprecated in Oracle Database 11*g* Release 2 (11.2). They are supported in this release for backward compatibility. However, Oracle recommends that you migrate away from these deprecated features.

Dictionary-managed Tablespaces

Oracle recommends that you create locally managed tablespaces. Locally managed tablespaces are much more efficiently managed than dictionary-managed tablespaces.

MAX_JOB_SLAVE_PROCESSES

`MAX_JOB_SLAVE_PROCESSES` has been deprecated. Use `JOB_QUEUE_PROCESSES` instead.

Changes to LOG_ARCHIVE_DEST_*n* Parameters

Starting with Oracle Database 11*g* Release 2 (11.2), the number of supported destinations in the `LOG_ARCHIVE_DEST_`*n* and the `LOG_ARCHIVE_DEST_STATE_`*n* parameters have been increased from 10 to 31. Destinations `LOG_ARCHIVE_DEST_11` through `LOG_ARCHIVE_DEST_31` do not support the `SYNC`, `ARCH`, `LOCATION`, `MANDATORY`, `ALTERNATE`, or `DEPENDENCY` attributes, and cannot be specified as the target of the `ALTERNATE` or `DEPENDENCY` attributes.

`LOG_ARCHIVE_DEST_11` through `LOG_ARCHIVE_DEST_31` can only be used when the `COMPATIBLE` initialization parameter is set to `11.2.0` or higher.

Compatibility and Interoperability Issues in Oracle Database 11*g* Release 1 (11.1)

The following sections describe compatibility and interoperability issues introduced in Oracle Database 11*g* Release 1 (11.1) and actions you can take to prevent problems resulting from these issues.

- Initialization Parameters Deprecated in Oracle Database 11g Release 1 (11.1)
- Initialization Parameters Obsolete in Oracle Database 11g Release 1 (11.1)
- Static Data Dictionary Views with Dropped Columns in Oracle Database 11g Release 1 (11.1)
- Deprecated Features in Oracle Database 11g Release 1 (11.1)
- Automatic Maintenance Tasks Management
- New SYSASM Privilege and OSASM Group for ASM Administration
- ASM Disk Group Compatibility
- COMPUTE STATISTICS and ESTIMATE STATISTICS Clauses
- Oracle Data Mining Models and the DMSYS Schema Objects
- Oracle Data Mining Scoring Engine
- SQL Plan Management and Control of SQL Plan Baselines
- Binary XML Support for Oracle XML Database
- PL/SQL Native Compilation and Access Control for Network Utility Packages
- PL/SQL Control Parameters
- Change in WebDAV ACL Evaluation Rules in Oracle XML DB
- Summary Management and SQL Access Advisor
- Standard Edition Starter Database
- Core Dump Location
- New Default Value for UNDO_MANAGEMENT
- LOG_ARCHIVE_DEST_n Parameters
- SHARED_POOL_SIZE Parameter
- JOB_QUEUE_PROCESSES Parameter and Scheduling Jobs
- Automatic Diagnostic Repository

Initialization Parameters Deprecated in Oracle Database 11*g* Release 1 (11.1)

The following initialization parameters were deprecated in Oracle Database 11*g* Release 1 (11.1).

To get a list of all deprecated initialization parameters, issue the following SQL statement:

```
SQL> SELECT name FROM v$parameter
        WHERE isdeprecated = 'TRUE';
```

A deprecated parameter behaves the same way as a regular parameter, except that a warning message is displayed at instance startup if a deprecated parameter is specified in the parameter file. In addition, all deprecated parameters are logged to the alert log at instance startup.

`BACKGROUND_DUMP_DEST` (replaced by `DIAGNOSTIC_DEST`)
`COMMIT_WRITE`
`CURSOR_SPACE_FOR_TIME`
`INSTANCE_GROUPS`
`LOG_ARCHIVE_LOCAL_FIRST`
`PLSQL_DEBUG` (replaced by `PLSQL_OPTIMIZE_LEVEL`)
`PLSQL_V2_COMPATIBILITY`
`REMOTE_OS_AUTHENT`
`RESOURCE_MANAGER_CPU_ALLOCATION`
`STANDBY_ARCHIVE_DEST`
`TRANSACTION_LAG` attribute (of the `CQ_NOTIFICATION$_REG_INFO` object)
`USER_DUMP_DEST` (replaced by `DIAGNOSTIC_DEST`)

Initialization Parameters Obsolete in Oracle Database 11*g* Release 1 (11.1)

These initialization parameters were made obsolete in Oracle Database 11*g* Release 1 (11.1).

`DDL_WAIT_FOR_LOCKS`
`LOGMNR_MAX_PERSISTENT_SESSIONS`
`PLSQL_COMPILER_FLAGS`

> **Note:** An attempt to start a database using one or more obsolete initialization parameters will succeed, but a warning is returned and recorded in the alert log.

Static Data Dictionary Views with Dropped Columns in Oracle Database 11*g* Release 1 (11.1)

These static data dictionary view columns were dropped in Oracle Database 11*g* Release 1 (11.1):

Static Data Dictionary View	Dropped Columns
`V$DATAFILE`	`PLUGGED_IN`

Deprecated Features in Oracle Database 11*g* Release 1 (11.1)

This section lists Oracle Database features deprecated in Oracle Database 11*g* Release 1 (11.1). They are supported in this release for backward compatibility. However, Oracle recommends that you migrate away from these deprecated features.

- Oracle Ultra Search
- Java Development Kit (JDK) 1.4

  Oracle recommends that you use JDK 5.0; but JDK 1.5 is also fully supported.
- CTXXPATH index

  Oracle recommends that you use XMLIndex instead.

> **See Also:** *Oracle XML DB Developer's Guide*

Automatic Maintenance Tasks Management

Automatic Maintenance Tasks Management, a new database component in Oracle Database 11*g* Release 1 (11.1), schedules all automatic maintenance tasks in an expanded set of maintenance windows. Automatic Maintenance Tasks Management enables you to exercise finer control over maintenance task scheduling for tasks such as optimizer statistics gathering, Segment Advisor, and Automatic SQL Tuning Advisor.

Automatic Maintenance Tasks Management uses all existing maintenance windows (for example, windows that are current members of the `MAINTENANCE_WINDOW_GROUP`. Existing resource plans associated with the maintenance windows are used. However, `AUTOTASK_CONSUMER_GROUP` is replaced in the resource plans by the AutoTask Resource Subplan.

If you disable either Optimizer Statistics Gathering or Segment Advisor jobs in 10*g*, then the corresponding Automatic Maintenance Tasks Management feature is disabled after upgrading to Oracle Database 11*g* Release 1 (11.1).

The following list shows the default settings for maintenance tasks:

- Online backup is disabled
- Optimizer Statistics Gathering is on
- Segment Advisor is on
- Automatic SQL Tuning is on

All other Automatic Maintenance Tasks Management clients are enabled by default.

Although Automatic Maintenance Tasks Management is automatically enabled when upgrading to Oracle Database 11*g* Release 1 (11.1), AutoTask online backup is not enabled automatically. You must configure online backup manually, if desired, after upgrading the database. If you perform a database downgrade, then Automatic Maintenance Tasks Management reverts to the default behavior for that release.

> **See Also:** The *Oracle Database Administrator's Guide* for complete information about the Automatic Maintenance Tasks Management feature

New SYSASM Privilege and OSASM Group for ASM Administration

Oracle Database 11*g* Release 1 (11.1) introduces a new `SYSASM` privilege that is specifically intended for performing ASM administration tasks. Using the `SYSASM` privilege instead of the `SYSDBA` privilege provides a clearer division of responsibility between ASM administration and database administration.

Warning messages will appear in the ASM `alert.log` if `SYSDBA` performs disk group maintenance (`CREATE DISKGROUP`, `MOUNT/DISMOUNT`, `ADD/DROP DISK`, `ONLINE/OFFLINE DISK`, `DROP DISKGROUP`). These tasks are deprecated for `SYSDBA`; they should be performed by `SYSASM`.

`OSASM` is a new operating system (OS) group that is used exclusively for ASM. Members of the `OSASM` group can connect `AS SYSASM` using OS authentication and have full access to ASM.

This feature is described in more detail in "About Upgrading System Authentication for Oracle ASM Instances" on page 3-5.

See Also: *Oracle Database Storage Administrator's Guide* for more information about accessing ASM instances

ASM Disk Group Compatibility

Starting with Oracle Database 11*g* Release 1 (11.1), you can advance the Oracle Database and the ASM disk group compatibility settings across software versions. Using the new compatibility attributes, `compatible.rdbms` and `compatible.asm`, you can specify the minimum software version required to use the disk group for the database and the disk group for ASM, respectively.

This feature enables heterogeneous environments with disk groups from Oracle Database 10*g* Release 1 (10.1), Oracle Database 10*g* Release 2 (10.2), and Oracle Database 11*g* Release 1 (11.1). By default, both attributes are set to 10.1. You must advance these attributes to take advantage of the new features.

See Also: *Oracle Database Storage Administrator's Guide* for more information on ASM disk group compatibility

COMPUTE STATISTICS and ESTIMATE STATISTICS Clauses

In earlier releases, the `ANALYZE...COMPUTE STATISTICS` and `ANALYZE...ESTIMATE STATISTICS` clauses could be used to start or stop the collection of statistics on an index. These clauses have been made obsolete. Oracle Database 11*g* Release 1 (11.1) automatically collects statistics during index creation and rebuild. These clauses are no longer supported.

Oracle Data Mining Models and the DMSYS Schema Objects

During the upgrade to Oracle Database 11*g* Release 1 (11.1), `DMSYS` schema objects along with user models residing in user schemas are upgraded from any previous release without major constraints. Upon completion of the upgrade, the mining metadata is migrated into the `SYS` schema while the user models continue functioning with the new metadata. Oracle recommends that you drop the `DMSYS` schema after setting the `COMPATIBLE` initialization parameter to 11.0.0. In addition, the DBA will need to grant the new `CREATE MINING MODEL` privilege so that existing users can continue to build mining models.

Data mining models residing in a user schema are automatically upgraded as part of the model upgrade, which is an integral part of the Oracle Database upgrade process. Data mining model Export and Import utilities can also be used as a means of upgrading data mining models from one release to another.

During the database downgrade process, the data mining component is downgraded to a previous release. The downgrade process reloads `DMSYS` objects such as packages, types, and table objects as well as downgrading model objects residing in user schemas (if any). Objects that were created as a part of the database upgrade are removed from the `SYS` schema during the downgrade procedure. The process is transparent and does not require any user intervention.

After upgrading (and dropping the `DMSYS` schema after setting the `COMPATIBLE` initialization parameter to 11.0.0), importing models that were exported from Oracle Database 10*g* Release 1 (10.1) might have some complications due to their reference to the now nonexistent `DMSYS` schema. To handle this case, Oracle provides scripts to sufficiently (and minimally) mimic the `DMSYS` interface present in the Oracle Database 10*g* Release 1 (10.1) database so that the Import process can proceed. This is not a common occurrence because models become stale over time and users typically want to rebuild their models rather than import older ones.

Note that Data Mining is not protected by the `COMPATIBLE` initialization parameter. If `COMPATIBLE` is set at `10.1.0` or `10.2.0` while the database has been upgraded to Oracle Database 11*g* Release 1 (11.1), then all new and existing Data Mining features and functions should work. If you have built new mining models that are only available in Oracle Database 11*g* Release 1 (11.1), and subsequently decide to downgrade the database to Oracle Database 10*g* Release 2 (10.2), you will be required to drop the new mining models before downgrading.

Oracle Data Mining Scoring Engine

Starting with Oracle Database 11*g* Release 1 (11.1), the Oracle Data Mining Scoring Engine can no longer be installed.

See Also: *Oracle Data Mining Administrator's Guide*

SQL Plan Management and Control of SQL Plan Baselines

The use of stored outlines is deprecated in Oracle Database 11*g* Release 1 (11.1). Instead, you should use the SQL plan management feature that enables the optimizer to maintain a history of execution plans for a SQL statement. Using the execution plan history, the optimizer is able to detect a new plan representing a plan change for a SQL statement. When the optimizer detects a new plan, it stores the new plan and marks it for performance evaluation and uses the old (currently known good) plan. The optimizer uses the new plan only after its performance is verified to be better than that of the old plan. A SQL plan baseline consists of a set of known good plans for a SQL statement.

Migration of SQL Profiles

SQL Profiles are SQL management objects that were introduced in Oracle Database 10*g* Release 1 (10.1). These objects resided in a section of the dictionary that was defined in `SYSTEM` tablespace. The dictionary tables storing the SQL profiles are restructured to accommodate the storage of SQL plan baselines, which are also SQL management objects. Further, these dictionary tables are now defined in the `SYSAUX` tablespace.

When you upgrade from Oracle Database 10*g* Release 1 (10.1) to Oracle Database 11*g* Release 1 (11.1), the database upgrade script moves existing SQL profiles from the `SYSTEM` tablespace to the `SYSAUX` tablespace. Thus, if an Oracle Database 11*g* Release 1 (11.1) database instance is up but the `SYSAUX` tablespace is offline, then the optimizer is not able to access SQL Management objects, which can affect the performance on some of the SQL workload. In contrast, in Oracle Database 10*g* Release 1 (10.1), because SQL profiles were stored in `SYSTEM` tablespace, the unavailability of SQL profiles did not exist. Note that starting with Oracle Database 11*g* Release 1 (11.1), taking the `SYSAUX` tablespace offline can have potential SQL performance consequences.

Backward Compatibility

In Oracle Database 11*g* Release 1 (11.1):

- If a stored outline for a SQL statement is active for the user session (for example, the stored outline category matches with the user session category), then the statement is compiled using the stored outline.
- If a private outline is available for a SQL statement, then the statement is compiled using the private outline.

If a stored outline is available for a SQL statement, then the SQL Plan Management feature is not used. However, if another user session uses the same SQL statement but without an active stored outline, then the SQL plan management feature is used.

See Also:

- *Oracle Database Performance Tuning Guide* for more information about SQL Plan Management
- *Oracle Database PL/SQL Packages and Types Reference* for more information about the DBMS_SPM package

Binary XML Support for Oracle XML Database

The binary XML storage option that is new in Oracle Database 11*g* Release 1 (11.1) is available when the COMPATIBLE initialization parameter is set to 11.0.0 or higher. When you create a table or column with this storage option, the minimum compatibility requirement is checked. This also applies when storing binary XML documents directly in the XML DB repository.

When Upgrading to Oracle Database 11*g* Release 1 (11.1)

When the database is upgraded to Oracle Database 11*g* Release 1 (11.1), none of the existing user XMLType tables and instances is modified in any fashion. Existing tables can be altered and new tables can be subsequently created using the new storage format after the upgrade is completed. The XDB tables XDB$CONFIG and XDB$ACL and the corresponding XML schemas are migrated to binary XML storage when a database is upgraded to Oracle Database 11*g* Release 1 (11.1).

PL/SQL Native Compilation and Access Control for Network Utility Packages

Oracle introduced compatibility and interoperability changes in PL/SQL for Oracle Database 11*g* Release 1 (11.1).

- PL/SQL Native Compilation
- Access Control for Network Utility Packages

PL/SQL Native Compilation

Starting with Oracle Database 11*g*, PL/SQL Native Compilation does not need a C compiler. Therefore, if you presently use a C compiler only to support PL/SQL Native Compilation, you can remove it from the machine where your database is installed (and from each node in an Oracle RAC configuration).

Moreover, the output of PL/SQL Native Compilation is no longer materialized on the file system. There, the Oracle Database 10*g* initialization parameters PLSQL_Native_Library_Dir and PLSQL_Native_Library_Subdir_Count have no significance in Oracle Database 11*g*. The directories that they denoted, and the contents of these directories, can be safely deleted on completion of the upgrade process.

Further, the SPNC_COMMANDS file (in the *ORACLE_HOME*/plsql directory) is no longer needed.

Only one initialization parameter, PLSQL_Code_Type, remains for controlling PL/SQL Native Compilation. The DBA, therefore, no longer needs to have any interest in PL/SQL Native Compilation.

Access Control for Network Utility Packages

The default behavior for access control to network utility packages has been changed to disallow network operations to all nonprivileged users. This default behavior is different from, and is incompatible with, previous versions of Oracle Database.

For database users upgrading to Oracle Database 11*g* Release 1 (11.1), applications that depend on the PL/SQL network utility packages compile without any issues. However, at runtime the applications might receive exceptions when attempting to perform privileged network operations. Although you can restore the compatibility by using a wildcard to grant those privileges to perform any network operations to `PUBLIC`, Oracle strongly advises that database administrators carefully review each situation on an individual basis and grant privileges only as needed.

> **Note:** Oracle XML DB is required to properly maintain the access control lists. If Oracle XML DB is not already installed on the system, then you must install it during the upgrade procedure.

See Also: "Configuring Fine-Grained Access to External Network Services After Upgrading Oracle Database" on page 4-4

PL/SQL Control Parameters

The behavior of some Oracle parameters that control the behavior of PL/SQL have changed as of Oracle Database 11*g* Release 1 (11.1).

- If PL/SQL debug code generation mode is selected by any parameter setup, then native code generation is turned off.
- Debug code generation is on if the `PLSQL_OPTIMIZE_LEVEL <= 1`.
- `PLSQL_DEBUG` is deprecated.

  You should use `PLSQL_OPTIMIZE_LEVEL` instead. A deprecation warning is issued if `PLSQL_DEBUG` is used.
- If `PLSQL_OPTIMIZE_LEVEL <= 1`, then native code generation is turned off.
- `PLSQL_COMPILER_FLAGS` is obsolete. It has no effect any longer and draws an error message that an illegal option is being set.
- `PLSQL_V2_COMPATIBILITY` is deprecated.

Change in WebDAV ACL Evaluation Rules in Oracle XML DB

Oracle XML DB uses a security mechanism that is based on access-control lists (ACLs) to restrict access to any Oracle XML DB resource. An ACL is a list of access-control entries (ACEs) that determine which users, roles, and groups have access to a given resource.

There have been changes to the treatment of WebDAV ACL entries. Before Oracle Database 11*g* Release 1 (11.1), a `<deny>` entry always trumped any `<allow>` entry in a given ACL. Starting with Oracle Database 11*g* Release 1 (11.1), ACE order is irrelevant. The default behavior is determined only by the first `<allow>` or `<deny>` entry that is encountered. That is, the first entry determines the behavior for that principal and additional ACEs for that principal have no effect.

This change in the default behavior is different from, and is incompatible with, previous versions of Oracle Database. When upgrading to Oracle Database 11*g* Release 1 (11.1), you can get the same behavior as in previous releases by manually reordering the ACLs (if necessary). That is, if there are any ACLs that have `<allow>` followed somewhere by `<deny>`, then you should (manually) reorder the ACLs so that the `<deny>` entry occurs first.

> **See Also:** *Oracle XML DB Developer's Guide* for more information about the ACL evaluation rules

Summary Management and SQL Access Advisor

Starting with Oracle Database 10*g* Release 2 (10.2), the `DBMS_OLAP` package, which is the Summary Advisor in Summary Management, is being deprecated and has been replaced by the SQL Access Advisor.

SQL Access Advisor Tasks

Due to internal structural changes to the SQL Access Advisor repository, a database upgrade resets all existing SQL Access Advisor tasks to their initial state. This effectively deletes all recommendation information for tasks that have successfully completed before the upgrade.

After upgrade, the recommendation information can be restored by re-executing the existing SQL Access Advisor tasks.

Standard Edition Starter Database

When the Standard Edition (SE) starter database is upgraded, the following components cannot be upgraded by the SE server because they require options that are not installed in the Standard Edition:

- OLAP Catalog
- OLAP Analytic Workspace
- Oracle OLAP API

After the upgrade, these components have a `STATUS` value of `'OPTION OFF'` in the `DBA_REGISTRY` view, and there will be some invalid objects in the associated component schemas. The Database Upgrade Assistant (DBUA) shows unsuccessful upgrades for these components.

Core Dump Location

On UNIX systems, when an application program crashes due to an unhandled signal, such as a segmentation fault, a core dump file is usually generated. The system default file name for this file is `core`, and it is located in the directory in which the application is currently running.

Starting with Oracle Database 11*g* Release 1 (11.1), applications using the Oracle Call Interface (OCI) can create a subdirectory named core*_process_id*, where *process_id* is the UNIX ID of the process that crashed. The `core` file is then placed in that subdirectory instead of the location where the application is running.

In sqlnet.ora, setting `DIAG_SIGHANDLER_ENABLED = TRUE` also puts the generated `core` file in the directory named core*_process_id*.

New Default Value for UNDO_MANAGEMENT

Starting with Oracle Database 11*g* Release 1 (11.1), the default value of the `UNDO_MANAGEMENT` parameter is `AUTO` so that automatic undo management is enabled by default. You must set the parameter to MANUAL to turn off automatic undo management, if required.

The UNDO_MANAGEMENT and ROLLBACK_SEGMENTS initialization parameters have changed from basic initialization parameters to non-basic initialization parameters. Most databases should be required to have only basic parameters set to run properly and efficiently.

See Also: UNDO_MANAGEMENT in *Oracle Database Reference*

LOG_ARCHIVE_DEST_*n* Parameters

Starting with Oracle Database 11*g* Release 1 (11.1), the LOG_ARCHIVE_DEST_*n* parameter can be used to specify a local archiving destination on a database instance running Oracle Standard Edition. Previously, this parameter could only be specified on a database instance running Oracle Enterprise Edition.

SHARED_POOL_SIZE Parameter

Migration utilities for this release recommend new values for SHARED_POOL_SIZE based on the value of internal SGA overheads in the pre-upgrade environment, which you can determine by running the following query before upgrading to Oracle Database 11*g* Release 1 (11.1):

```
SQL> SELECT SUM(BYTES) FROM v$sgastat WHERE pool = 'shared pool';
```

In Oracle Database 11*g* Release 1 (11.1), the exact value of internal SGA overhead, or Startup overhead in Shared Pool, is listed in the new v$sgainfo view.

In manual SGA mode, values of SHARED_POOL_SIZE that are too small to accommodate the internal SGA overhead result in an ORA-00371 error during startup. This generated error message includes a suggested value for the SHARED_POOL_SIZE parameter. If you are using automatic shared memory management, the size of the shared pool is tuned automatically, and the ORA-00371 error is never generated.

The amount of shared pool memory allocated by Oracle Database releases before Oracle Database 10*g* Release 1 (10.1) was equal to the sum of the value of the SHARED_POOL_SIZE initialization parameter and the internal SGA overhead computed during instance startup. This overhead was based on the values of several other initialization parameters.

For example, if the SHARED_POOL_SIZE parameter is 64 megabytes and the internal SGA overhead is 12 megabytes, then the real size of shared pool in the SGA would be 76 megabytes, although the value of the SHARED_POOL_SIZE parameter would still be displayed as megabytes.

Starting with Oracle Database 10*g* Release 1 (10.1), the size of internal SGA overhead is included in the value of the SHARED_POOL_SIZE parameter. The shared pool memory allocated at startup is exactly the value of SHARED_POOL_SIZE. Therefore, this parameter must be set such that it includes both the internal SGA overhead and the desired effective value of the shared pool size.

Assuming that the internal SGA overhead remains unchanged, the effective available value of shared pool after startup would be 12 megabytes less than the value of the SHARED_POOL_SIZE parameter, or 52 megabytes. To maintain 64 megabytes for the effective value of shared pool memory, set the parameter to 76 megabytes.

JOB_QUEUE_PROCESSES Parameter New Default Setting

Beginning with Oracle Database 11*g* Release 1 (11.1), the `JOB_QUEUE_PROCESSES` parameter is changed from a basic to a non-basic initialization parameter. Most databases must only have basic parameters set to run properly and efficiently. The default value is also changed from `0` to `1000`.

Note that Oracle Database overrides the job queue setting to disable scheduler jobs during upgrade mode. Therefore, there is no need to change this setting when upgrading Oracle Database.

> **See Also:** *Oracle Database Reference* for more information on this parameter

Automatic Diagnostic Repository

The locations of alert logs and trace files are no longer set by the initialization parameters `BACKGROUND_DUMP_DEST` and `USER_DUMP_DEST`. They are now kept in the Automatic Diagnostic Repository (ADR), whose location is set the by the initialization parameter `DIAGNOSTIC_DEST`.

> **See Also:** *Oracle Database Administrator's Guide* for more information on the management of diagnostic information

Compatibility and Interoperability Issues in Oracle Database 10*g* Release 2 (10.2)

The following sections describe compatibility and interoperability issues introduced in Oracle Database 10*g* Release 2 (10.2). If you are upgrading to Oracle Database 11*g* Release 1 (11.1) from a release earlier than Oracle Database 10*g* Release 2 (10.2), then see the following sections for information about actions you can take to prevent problems resulting from these issues:

- Initialization Parameters Deprecated in Oracle Database 10g Release 2 (10.2)
- Initialization Parameters Obsolete in Oracle Database 10g Release 2 (10.2)
- Static Data Dictionary Views with Dropped Columns in Oracle Database 10g Release 2 (10.2)
- SQL
- CONNECT Role
- Time Zone Files
- New Limit for FAILED_LOGIN_ATTEMPTS

Initialization Parameters Deprecated in Oracle Database 10*g* Release 2 (10.2)

The following initialization parameters were deprecated in Oracle Database 10*g* Release 2 (10.2). To get a list of all deprecated initialization parameters, issue the following SQL statement:

```
SQL> SELECT name FROM v$parameter
        WHERE isdeprecated = 'TRUE';
```

A deprecated parameter behaves the same way as a regular parameter, except that a warning message is displayed at instance startup if a deprecated parameter is specified in the parameter file. In addition, all deprecated parameters are logged to the alert log at instance startup:

```
LOGMNR_MAX_PERSISTENT_SESSIONS
MAX_COMMIT_PROPAGATION_DELAY
REMOTE_ARCHIVE_ENABLE
SERIAL_REUSE
SQL_TRACE
```

Initialization Parameters Obsolete in Oracle Database 10*g* Release 2 (10.2)

The following initialization parameters were made obsolete in Oracle Database 10*g* Release 2 (10.2).

Note: An attempt to start a database using one or more obsolete initialization parameters will succeed, but a warning is returned and recorded in the alert log.

```
ENQUEUE_RESOURCES
```

Static Data Dictionary Views with Dropped Columns in Oracle Database 10*g* Release 2 (10.2)

The following static data dictionary view columns were dropped in Oracle Database 10*g* Release 2 (10.2):

Static Data Dictionary View	Dropped Columns
DBA_HIST_SQLBIND	CHILD_NUMBER

SQL

The behavior of date formats has changed when used with XML functions. The XML Schema standard specifies that dates and timestamps in XML data be in standard formats. Before Oracle Database 10*g* Release 2 (10.2), dates and timestamps in XML data did not follow this standard; rather, the format of dates and timestamps in generated XML was determined by the database format.

As of Oracle Database 10*g* Release 2 (10.2), the XML generation functions in Oracle XML DB produce dates and timestamps according to the XML schema standard.

See Also: *Oracle XML DB Developer's Guide* for more information

CONNECT Role

After upgrading from a release earlier than Oracle Database 10*g* Release 2 (10.2), the CONNECT role has only the CREATE SESSION privilege; the other privileges granted to the CONNECT role in earlier releases are revoked during the upgrade. For further information about this, see "Updating the CONNECT Role from Earlier Releases" on page 3-16.

Time Zone Files

The time zone files that are supplied with Oracle Database 10*g* Release 2 (10.2) have been updated from version 4 to version 8 to reflect changes in transition rules for some time zone regions. The changes might affect existing data of TIMESTAMP WITH TIME ZONE data type. For further information about this, see "About Warnings for TIMESTAMP WITH TIME ZONE Data Type" on page 3-18.

New Limit for FAILED_LOGIN_ATTEMPTS

As of Oracle Database 10*g* Release 2 (10.2), the limit for FAILED_LOGIN_ATTEMPTS for the DEFAULT profile is 10. Before Oracle Database 10*g* Release 2 (10.2), the default was UNLIMITED.

Compatibility and Interoperability Issues in Oracle Database 10*g* Release 1 (10.1)

The following sections describe compatibility and interoperability issues introduced in Oracle Database 10*g* Release 1 (10.1). If you are upgrading to Oracle Database 11*g* Release 1 (11.1) from a release earlier than Oracle Database 10*g* Release 1 (10.1), then see the following sections for information about actions you can take to prevent problems resulting from these issues:

- Initialization Parameters Deprecated in Oracle Database 10g Release 1 (10.1)
- Initialization Parameters Obsolete in Oracle Database 10g Release 1 (10.1)
- Static Data Dictionary Views Deprecated in Oracle Database 10g Release 1 (10.1)
- Static Data Dictionary Views Obsolete in Oracle Database 10g Release 1 (10.1)
- Dynamic Performance Views Deprecated in Oracle Database 10g Release 1 (10.1)
- Dynamic Performance Views Obsolete in Oracle Database 10g Release 1 (10.1)
- SQL Optimizer
- SQL Changes
- Invalid Synonyms After an Upgrade
- Manageability
- Transaction and Space
- Recovery and Data Guard
- RMAN
- CREATE DATABASE
- Oracle Real Application Clusters
- Materialized Views
- Change Data Capture
- Change in the Default Archival Processing to Remote Archive Destinations
- Limitations on NCHAR Data Types
- PL/SQL Native Compilation
- Evaluation of Numeric Literals
- Change in Behavior for SESSION_CACHED_CURSORS
- New Default Value for DB_BLOCK_SIZE
- OPTIMIZER_MAX_PERMUTATIONS and OPTIMIZER_FEATURES_ENABLE
- Change in Behavior for LOG_ARCHIVE_FORMAT
- New Default Value for PGA_AGGREGATE_TARGET
- Change in Behavior for SHARED_POOL_SIZE

- Shared Server Parameters

Initialization Parameters Deprecated in Oracle Database 10*g* Release 1 (10.1)

The following initialization parameters were deprecated in Oracle Database 10*g* Release 1 (10.1) To get a list of all deprecated initialization parameters, issue the following SQL statement:

```
SQL> SELECT name FROM v$parameter
       WHERE isdeprecated = 'TRUE';
```

A deprecated parameter behaves the same way as a regular parameter, except that a warning message is displayed at instance startup if a deprecated parameter is specified in the parameter file. In addition, all deprecated parameters are logged to the alert log at instance startup:

`BUFFER_POOL_KEEP` (replaced by `DB_KEEP_CACHE_SIZE`)
`BUFFER_POOL_RECYCLE` (replaced by `DB_RECYCLE_CACHE_SIZE`)
`GLOBAL_CONTEXT_POOL_SIZE`
`LOCK_NAME_SPACE`
`LOG_ARCHIVE_START`
`MAX_ENABLED_ROLES`
`PARALLEL_AUTOMATIC_TUNING`
`PLSQL_COMPILER_FLAGS` (replaced by `PLSQL_CODE_TYPE` and `PLSQL_DEBUG`)
`SQL_VERSION`

Initialization Parameters Obsolete in Oracle Database 10*g* Release 1 (10.1)

The following initialization parameters were made obsolete in Oracle Database 10*g* Release 1 (10.1).

> **Note:** An attempt to start a database using one or more obsolete initialization parameters will succeed, but a warning is returned and recorded in the alert log.

`DBLINK_ENCRYPT_LOGIN`
`HASH_JOIN_ENABLED`
`LOG_PARALLELISM`
`MAX_ROLLBACK_SEGMENTS`
`MTS_CIRCUITS`
`MTS_DISPATCHERS`
`MTS_LISTENER_ADDRESS`
`MTS_MAX_DISPATCHERS`
`MTS_MAX_SERVERS`
`MTS_MULTIPLE_LISTENERS`
`MTS_SERVERS`
`MTS_SERVICE`
`MTS_SESSIONS`
`OPTIMIZER_MAX_PERMUTATIONS`
`ORACLE_TRACE_COLLECTION_NAME`
`ORACLE_TRACE_COLLECTION_PATH`
`ORACLE_TRACE_COLLECTION_SIZE`
`ORACLE_TRACE_ENABLE`
`ORACLE_TRACE_FACILITY_NAME`
`ORACLE_TRACE_FACILITY_PATH`

PARTITION_VIEW_ENABLED
PLSQL_NATIVE_C_COMPILER
PLSQL_NATIVE_LINKER
PLSQL_NATIVE_MAKE_FILE_NAME
PLSQL_NATIVE_MAKE_UTILITY
ROW_LOCKING
SERIALIZABLE
TRANSACTION_AUDITING
UNDO_SUPPRESS_ERRORS

Static Data Dictionary Views Deprecated in Oracle Database 10*g* Release 1 (10.1)

The following static data dictionary views were deprecated in Oracle Database 10*g* Release 1 (10.1).

ALL_STORED_SETTINGS (replaced by ALL_PLSQL_OBJECT_SETTINGS)
DBA_STORED_SETTINGS (replaced by DBA_PLSQL_OBJECT_SETTINGS)
USER_STORED_SETTINGS (replaced by USER_PLSQL_OBJECT_SETTINGS)

Static Data Dictionary Views Obsolete in Oracle Database 10*g* Release 1 (10.1)

The following static data dictionary views were made obsolete in Oracle Database 10*g* Release 1 (10.1).

ALL_ Views	DBA_ Views	USER_ Views
ALL_SOURCE_TAB_COLUMNS	DBA_SOURCE_TAB_COLUMNS	USER_SOURCE_TAB_COLUMNS

Dynamic Performance Views Deprecated in Oracle Database 10*g* Release 1 (10.1)

The following dynamic performance views were deprecated in Oracle Database 10*g* Release 1 (10.1):

GV$CACHE
GV$CACHE_TRANSFER
GV$CLASS_CACHE_TRANSFER (replaced by GV$INSTANCE_CACHE_TRANSFER)
GV$FALSE_PING
GV$FILE_CACHE_TRANSFER (replaced by GV$INSTANCE_CACHE_TRANSFER)
GV$GC_ELEMENTS_WITH_COLLISIONS
GV$LOCK_ACTIVITY
GV$TEMP_CACHE_TRANSFER (replaced by GV$INSTANCE_CACHE_TRANSFER)
V$CACHE
V$CACHE_LOCK
V$CACHE_TRANSFER
V$CLASS_CACHE_TRANSFER (replaced by V$INSTANCE_CACHE_TRANSFER)
V$FALSE_PING
V$FILE_CACHE_TRANSFER (replaced by V$INSTANCE_CACHE_TRANSFER)
V$GC_ELEMENTS_WITH_COLLISIONS
V$LOCK_ACTIVITY
V$TEMP_CACHE_TRANSFER (replaced by V$INSTANCE_CACHE_TRANSFER)

Dynamic Performance Views Obsolete in Oracle Database 10*g* Release 1 (10.1)

The following dynamic performance views were made obsolete in Oracle Database 10*g* Release 1 (10.1):

GV$ Views	V$ Views
GV$COMPATIBILITY	V$COMPATIBILITY
GV$COMPATSEG	V$COMPATSEG
GV$MLS_PARAMETERS	V$MLS_PARAMETERS
GV$MTS	V$MTS

SQL Optimizer

This section describes compatibility and interoperability issues relating to the SQL Optimizer in Oracle Database 10*g* Release 1 (10.1).

Rule-Based Optimizer Desupported

Starting with Oracle Database 10*g* Release 1 (10.1), the cost-based optimizer (CBO) is now enabled by default. The rule-based optimizer is no longer supported in Oracle Database 10*g* Release 1 (10.1). As a result, `rule` and `choose` are no longer supported as `OPTIMIZER_MODE` initialization parameter values and a warning is displayed in the alert log if `OPTIMIZER_MODE` is set to either of these values.

> **See Also:** *Oracle Database Performance Tuning Guide* for more information about the cost-based optimizer

Optimizer Statistics

Collection of optimizer statistics is now automatically performed by default for all schemas (including `SYS`), for pre-existing databases upgraded from a release earlier than Oracle Database 10*g* Release 1 (10.1), and for newly created databases. Gathering optimizer statistics on stale objects is scheduled by default to occur daily during the maintenance window.

> **See Also:** *Oracle Database Performance Tuning Guide* for more information about optimizer statistics

COMPUTE STATISTICS Clause of CREATE INDEX

In earlier releases, the `COMPUTE STATISTICS` clause of `CREATE INDEX` could be used to start or stop the collection of statistics on an index. This clause has been deprecated. Oracle Database 10*g* Release 1 (10.1) and later releases automatically collect statistics during index creation and rebuild. This clause is supported for backward compatibility and does not cause errors.

SKIP_UNUSABLE_INDEXES

In earlier releases, `SKIP_UNUSABLE_INDEXES` was a session parameter only. In Oracle Database 10*g* Release 1 (10.1) and later releases, it is an initialization parameter and defaults to `true`. The `true` setting disables error reporting of indexes and index partitions marked `UNUSABLE`. This setting allows all operations (inserts, deletes, updates, and selects) on tables with unusable indexes or index partitions.

> **See Also:** `SKIP_UNUSABLE_INDEXES` in *Oracle Database Reference*

SQL Changes

Starting with Oracle Database 10*g* Release 1 (10.1), CLOB <-> NCLOB implicit conversion in SQL and PL/SQL is allowed.

Starting with Oracle Database 10*g* Release 1 (10.1), name resolution for synonyms has changed. If the base object of a synonym does not exist, then the SQL compiler now tries looking up `PUBLIC.base_object`.

Starting with Oracle Database 10*g* Release 1 (10.1), VPD policies are attached to synonyms rather than the base objects.

Invalid Synonyms After an Upgrade

Starting with Oracle Database 10*g* Release 1 (10.1), if a synonym (public or private) is pointing to an object that does not exist or is invalid, then the synonym is invalid after the upgrade.

Manageability

Database performance statistics are now automatically collected by the Automatic Workload Repository (AWR) database component for databases upgraded from a release earlier than Oracle Database 10*g* Release 1 (10.1) and for newly created databases. This data is stored in the `SYSAUX` tablespace, and is used by the database for automatic generation of performance recommendations.

See Also: *Oracle Database Performance Tuning Guide*

If you currently use Statspack for performance data gathering, then refer to the Statspack README `(spdoc.txt`, located in the *ORACLE_HOME*/rdbms/admin directory) for directions on using Statspack in Oracle Database 10*g* Release 1 (10.1) and later releases to avoid conflict with the AWR.

Transaction and Space

Starting with Oracle Database 10*g* Release 1 (10.1), dropped objects are now moved to the recycle bin where the space is only reused when it is needed. This allows an object to be undropped using the `FLASHBACK DROP` feature.

See Also: *Oracle Database Administrator's Guide*

Starting with Oracle Database 10*g* Release 1 (10.1), automatic tuning of undo retention is enabled by default. The `UNDO_SUPPRESS_ERRORS` initialization parameter has been deprecated. Errors generated when executing rollback segment operations while in automatic undo management mode are always suppressed.

Starting with Oracle Database 10*g* Release 1 (10.1), the default `AUTOEXTEND NEXT` size is larger for Oracle managed files (OMF).

See Also: *Oracle Database SQL Language Reference*

Recovery and Data Guard

Starting with Oracle Database 10*g* Release 1 (10.1), the `LOG_ARCHIVE_START` initialization parameter has been deprecated. Archiving is now automatically started when the database is placed in `ARCHIVELOG` mode.

Starting with Oracle Database 10*g* Release 1 (10.1), the `LOG_PARALLELISM` initialization parameter has been deprecated. Log file parallelism is now automatically enabled.

Starting with Oracle Database 10*g* Release 1 (10.1), the default value for the RECOVERY_PARALLELISM initialization parameter now defaults to allow parallel recovery.

Starting with Oracle Database 10*g* Release 1 (10.1), the default value for the parallel clause in the ALTER DATABASE RECOVER DATABASE statement has changed to PARALLEL.

See Also: *Oracle Database SQL Language Reference*

Starting with Oracle Database 10*g* Release 1 (10.1), the default buffer size for the ASYNC attribute of the LOG_ARCHIVE_DEST_*n* initialization parameter has increased from 2,048 blocks to 61,440 blocks.

Starting with Oracle Database 10*g* Release 1 (10.1), the default values of the parameters MAX_SGA and MAX_SERVERS as set by the DBMS_LOGSTDBY.APPLY_SET() procedure have changed.

See Also: *Oracle Database PL/SQL Packages and Types Reference*

Starting with Oracle Database 10*g* Release 1 (10.1), the default values for the Data Guard broker properties ApplyParallel, AsyncBlocks, and LogXptMode have changed.

See Also: *Oracle Data Guard Broker*

Starting with Oracle Database 10*g* Release 1 (10.1), the default behavior of the STARTUP SQL*Plus command and the ALTER DATABASE MOUNT and ALTER DATABASE OPEN SQL statements have changed for physical standby databases. The commands now automatically detect that the database is a physical standby and thus the STANDBY DATABASE and READ ONLY options are made default.

See Also: *Oracle Database SQL Language Reference*

RMAN

Starting with Oracle Database 10*g* Release 1 (10.1), RMAN now creates an empty file when restoring a file from backup and no backup of the file exists. RMAN backup of archived logs now automatically backs up logs that were created before the last resetlogs. Such logs were previously ignored.

Starting with Oracle Database 10*g* Release 1 (10.1), RMAN now continues to run the remaining portions of a backup or restore job when it encounters an error. RMAN now tries to restore from an alternate backup if it finds the targeted backup is corrupt.

CREATE DATABASE

In Oracle Database 10*g* Release 1 (10.1) and later releases, a SYSAUX tablespace is always created at database creation time or whenever a database is upgraded. The SYSAUX tablespace serves as an auxiliary tablespace to the SYSTEM tablespace. Because SYSAUX is the default tablespace for many Oracle features and products that previously required their own tablespaces, it reduces the number of tablespaces that a DBA must maintain.

See Also: *Oracle Database Administrator's Guide* for more information about the SYSAUX tablespace

Starting with Oracle Database 10*g* Release 1 (10.1), the minimum and default log file sizes have increased. The minimum size is now 4 MB. The default size is 50 MB, unless using Oracle managed files (OMF) in which case the default is 100 MB.

Oracle Real Application Clusters

In Oracle Database 10*g* Release 1 (10.1), there is an automated high availability (HA) framework for Oracle Real Application Clusters. The framework provides detection, recovery, restart, and notification services.

> **See Also:** *Oracle Real Application Clusters Administration and Deployment Guide* for more information

Materialized Views

Starting with Oracle Database 10*g* Release 1 (10.1), some privilege name changes have been made. The new names appear in all data dictionary views, but both the old and new names are accepted by the `GRANT` and `REVOKE` SQL statements.

- `CREATE SNAPSHOT` changed to `CREATE MATERIALIZED VIEW`
- `CREATE ANY SNAPSHOT` changed to `CREATE ANY MATERIALIZED VIEW`
- `ALTER ANY SNAPSHOT` changed to `ALTER ANY MATERIALIZED VIEW`
- `DROP ANY SNAPSHOT` changed to `DROP ANY MATERIALIZED VIEW`

Change Data Capture

Starting with Oracle Database 10*g* Release 1 (10.1), the interfaces in `DBMS_CDC_SUBSCRIBE` and `DBMS_CDC_PUBLISH` now take a subscription name parameter instead of a subscription handle.

> **See Also:** *Oracle Database PL/SQL Packages and Types Reference*

Starting with Oracle Database 10*g* Release 1 (10.1), subscriber views are now managed automatically. There is no longer any requirement to call the `DBMS_CDC_SUBSCRIBE` and `DBMS_CDC_PUBLISH` interfaces `PREPARE_SUBSCRIBER_VIEW()` and `DROP_SUBSCRIBER_VIEW()`.

Starting with Oracle Database 10*g* Release 1 (10.1), the computation of synchronous Change Data Capture's `RSID$` column has been changed to facilitate joining a subscriber view to itself in order to show both old and new values in the same row. The `RSID$` values for the `UO` and `UN` rows associated with the same update operation are now the same. To revert to the Oracle9*i* behavior where `UN RSID$` value is `UO RSID$` value + `1` for the same update operation, set event `10983` to level `4`.

Change in the Default Archival Processing to Remote Archive Destinations

Starting with Oracle Database 10*g* Release 1 (10.1), the default archival processing to remote destinations has changed so that archiver processes on the primary database completely and successfully archive the local online redo log files before transmitting the redo data to remote standby destinations. This default behavior is equivalent to setting the `LOG_ARCHIVE_LOCAL_FIRST` initialization parameter to `true`, which is also new in Oracle Database 10*g* Release 1 (10.1) and later releases. Note that this new default archival processing is relevant only when log transport services are defined to use archiver processes (ARC*n*), not the log writer process (LGWR), when the archiver processes are writing to remote destinations, and when the remote standby destination is not a mandatory destination.

Before Oracle Database 10*g* Release 1 (10.1), the default behavior was to transmit redo data to the standby destination at the same time the online redo log file was being archived to the local online redo log files. You can achieve this behavior by setting the `LOG_ARCHIVE_LOCAL_FIRST` initialization parameter to `false`. This archival processing is also relevant only when log transport services are defined to use archiver processes (ARC*n*), not the log writer process (LGWR), when the archiver processes are writing to remote destinations, and when the remote standby destination is not a mandatory destination.

The benefit of the new default behavior is that local archiving, and hence, processing on the primary database, are not affected by archival to non-mandatory, remote destinations. Because local archiving is now disassociated with remote archiving, sites that might have policies to delete archived redo log files on the primary database immediately after backing them up must make sure that the standby destinations have received the corresponding redo data before deleting the archived redo log files on the primary database. You can query the `V$ARCHIVED_LOG` view to verify that the redo data has been received on standby destinations.

Note: Any value specified for the `LOG_ARCHIVE_LOCAL_FIRST` initialization parameter is ignored for mandatory destinations (configured with the `MANDATORY` attribute of the `LOG_ARCHIVE_DEST_`*n* initialization parameters).

See Also: *Oracle Data Guard Concepts and Administration* for complete information about setting up archival to remote destinations

Limitations on NCHAR Data Types

Starting with Oracle9*i*, the `NCHAR` data types such as `NCHAR`, `NVARCHAR2`, and `NCLOB`, are limited to the Unicode character set encoding, `UTF8` and `AL16UTF16`.

PL/SQL Native Compilation

Starting with Oracle Database 10*g* Release 1 (10.1), the configuration of initialization parameters and the command setup for native compilation has been simplified. The important parameters now are `PLSQL_NATIVE_LIBRARY_DIR` and `PLSQL_NATIVE_LIBRARY_SUBDIR_COUNT`. The parameters related to the compiler, linker, and make utility have been made obsolete. Native compilation is turned on and off by a separate initialization parameter, `PLSQL_CODE_TYPE`, rather than being one of several options in the `PLSQL_COMPILER_FLAGS` parameter, which is now deprecated. The `spnc_commands` file, located in the *ORACLE_HOME*/plsql directory, contains the commands and options for compiling and linking, rather than a makefile.

See Also:

- *Oracle Database PL/SQL Language Reference* for further information about compiling PL/SQL code for native execution
- "PL/SQL Native Compilation (NCOMP) In Oracle Database 10g Rel 1" on the Oracle Technology Network (OTN):

  http://www.oracle.com/technetwork/database/features/plsql/index.html

Evaluation of Numeric Literals

Evaluation of numeric literals has changed such that at least one of the constants in a numeric computation with literals must be a decimal specified to the 10th place. This is because releases after Oracle Database 10*g* Release 1 (10.1) use `INTEGER` arithmetic (approximately 9 significant digits) for some expressions whereas Oracle9*i* Release 2 (9.2) used `NUMBER` arithmetic (approximately 38 significant digits).

Therefore, if you are dealing with results of greater than 9 significant digits, then one of the literals should be in decimal format to prevent numeric overflow errors. For example, in Oracle Database 10*g* Release 1 (10.1), the computation of `v1` in the following example causes a numeric overflow error:

```
DECLARE
  v1 NUMBER(38);
BEGIN
  v1 := 256*256*256*256;
  DBMS_OUTPUT.PUT_LINE(v1);
END;
/
```

The solution to the error is to specify one of the numeric literals as a decimal (256.0), as follows:

```
DECLARE
  v1 NUMBER(38);
BEGIN
  v1 := 256*256*256*256.0;
  DBMS_OUTPUT.PUT_LINE(v1);
END;
/
```

See Also:

- The "What’s New in SQL*Plus" section in the *SQL*Plus User’s Guide and Reference* to learn about new features in SQL*Plus
- *Oracle Database SQL Language Reference* for more information about upgrading SQL scripts

Change in Behavior for SESSION_CACHED_CURSORS

Starting with Oracle Database 10*g* Release 1 (10.1), the number of cached cursors is determined by the `SESSION_CACHED_CURSORS` initialization parameter. In previous Oracle Database releases, the number of SQL cursors cached by PL/SQL was determined by the `OPEN_CURSORS` initialization parameter.

See Also: `SESSION_CACHED_CURSORS` in *Oracle Database Reference*

New Default Value for DB_BLOCK_SIZE

Starting with Oracle Database 10*g* Release 1 (10.1), the default value of DB_BLOCK_SIZE is operating system specific, but is typically 8 KB (8192 bytes). In previous Oracle Database releases, the default value was 2 KB (2048 bytes). If DB_BLOCK_SIZE is not specified in the parameter file when upgrading from Oracle9*i* Release 2 (9.2), then you receive an error when attempting to start up your database. Add the following to your parameter file:

```
DB_BLOCK_SIZE = 2048
```

If DB_BLOCK_SIZE is specified in the parameter file, then Oracle Database uses this value instead of the default value of 8 KB.

OPTIMIZER_MAX_PERMUTATIONS and OPTIMIZER_FEATURES_ENABLE

Starting with Oracle Database 10*g*, the OPTIMIZER_MAX_PERMUTATIONS initialization parameter has been made obsolete. If you are upgrading from Oracle9*i* and have OPTIMIZER_FEATURES_ENABLE set to 8.1.7 or earlier and OPTIMIZER_MAX_PERMUTATIONS explicitly set to 2000 in the parameter file, then the release 8.1.7 default of 80000 is used when you start up the Oracle Database 11*g* Release 1 (11.1) database.

Setting OPTIMIZER_FEATURES_ENABLE to 9.0.0 or higher sets the default to 2000

Change in Behavior for LOG_ARCHIVE_FORMAT

Starting with Oracle Database 10*g* Release 1 (10.1), if the COMPATIBLE initialization parameter is set to 10.0.0 or higher, then archive log file names must contain each of the elements %s (sequence), %t (thread), and %r (resetlogs ID) to ensure that all archive log file names are unique. If the LOG_ARCHIVE_FORMAT initialization parameter is set in the parameter file, then make sure the parameter value contains the %s, %t, and %r elements.

New Default Value for PGA_AGGREGATE_TARGET

Starting with Oracle Database 10*g* Release 1 (10.1), Automatic PGA Memory Management is now enabled by default (unless PGA_AGGREGATE_TARGET is explicitly set to 0 or WORKAREA_SIZE_POLICY is explicitly set to MANUAL). PGA_AGGREGATE_TARGET defaults to 20% of the size of the SGA, unless explicitly set. Oracle recommends tuning the value of PGA_AGGREGATE_TARGET after upgrading.

See Also: *Oracle Database Performance Tuning Guide*

Change in Behavior for SHARED_POOL_SIZE

In previous releases, the amount of shared pool memory that was allocated was equal to the value of the SHARED_POOL_SIZE initialization parameter plus the amount of internal SGA overhead computed during instance startup. Starting with Oracle Database 10*g* Release 1 (10.1), the value of SHARED_POOL_SIZE must now also accommodate this shared pool overhead.

Since Oracle Database release 9.2, the shared pool can be partitioned into multiple parts. The partitions are called shared pool *subpools* and there can be up to seven subpools. Although there is no standard recommendation, you should configure the shared pool memory in a way that generates larger subpools than the default size. For example, 256M and 500M seem to work better as subpool sizes for Oracle Database releases 9*i* and 10*g*, respectively. Having adequate sizes allocated for shared pool subpools can avoid raising ORA-4031 errors.

Shared Server Parameters

Starting with Oracle Database 10*g* Release 1 (10.1), the recommended way to turn on shared server mode is to set `SHARED_SERVERS` to a value greater than `0`. This can be done at startup or dynamically after the instance is started. If shared server mode is turned off by setting `SHARED_SERVERS` to `0`, then this only affects new clients (that is, no new clients can connect in shared mode; clients that are already connected in shared mode continue to be serviced by shared servers).

In previous releases, the recommended way to turn on shared server mode was to set `DISPATCHERS`. If `SHARED_SERVERS` was changed to `0` and shared server clients were still connected, then client requests would hang.

Before Oracle Database 10*g* Release 1 (10.1), the following shared server parameters could not be changed dynamically:

- `MAX_SHARED_SERVERS`
- `MAX_DISPATCHERS`
- `SHARED_SERVER_SESSIONS`
- `CIRCUITS`

Starting with Oracle Database 10*g* Release 1 (10.1), these shared server parameters are dynamically modifiable.

New Default Value for DISPATCHERS

Starting with Oracle Database 10*g* Release 1 (10.1), the default for `DISPATCHERS` is `'(PROTOCOL=TCP)'`. `DISPATCHERS` is given this default value if it is not set or if it is set to `''` and `SHARED_SERVERS` is set to `1` or higher.

In previous releases, there was no default value for `DISPATCHERS`.

New Default Value for SHARED_SERVERS

Starting with Oracle Database 10*g* Release 1 (10.1), if `DISPATCHERS` is set such that the total number of dispatchers is equal to 0, then `SHARED_SERVERS` defaults to `0`. If `DISPATCHERS` is set such that the total number of dispatchers is greater than 0, then `SHARED_SERVERS` defaults to `1` as in previous releases.

In previous releases, if `DISPATCHERS` was set such that the number of dispatchers is equal to 0, then `SHARED_SERVERS` defaulted to `1`.

New Default Value for MAX_SHARED_SERVERS

Starting with Oracle Database 10*g* Release 1 (10.1), there is no preset default for `MAX_SHARED_SERVERS`. The maximum number of shared servers varies depending on the number of free process slots. If `MAX_SHARED_SERVERS` is not set or is set to a value greater than or equal to `PROCESSES`, then PMON does not spawn any more shared servers if the number of free process slots is either 2 (if `PROCESSES` is less than `24`) or is less than 1 / 8, unless the existing servers are involved in a deadlock situation. If the existing servers are involved in a deadlock situation, then no matter the transaction load, a new server is spawned if there is a free process slot.

In previous releases, the default for `MAX_SHARED_SERVERS` is `20`, or 2 * `SHARED_SERVERS`, whichever is greater, subject to the condition that `MAX_SHARED_SERVERS` does not exceed `PROCESSES`.

Starting with Oracle Database 10*g* Release 1 (10.1), SHARED_SERVERS can be set higher than MAX_SHARED_SERVERS, in which case the number of servers remains constant at the level set for SHARED_SERVERS. This is to allow the range SHARED_SERVERS - MAX_SHARED_SERVERS to be changed without having to change these parameters in a specific order.

In previous releases, SHARED_SERVERS cannot be set higher than MAX_SHARED_SERVERS.

New Default Value for SHARED_SERVER_SESSIONS

Starting with Oracle Database 10*g* Release 1 (10.1), there is no preset default for SHARED_SERVER_SESSIONS. That is, if SHARED_SERVER_SESSIONS is not specified, then shared server sessions can be created as needed and as permitted by the session limit.

In previous releases, the default for SHARED_SERVER_SESSIONS was the maximum number of virtual circuits (CIRCUITS), or the maximum number of database sessions (SESSIONS) - 5, whichever is smaller.

New Default Value for CIRCUITS

Starting with Oracle Database 10*g* Release 1 (10.1), there is no preset default for CIRCUITS. That is, if CIRCUITS is not specified, then circuits can be created as needed and as permitted by dispatcher constraints and system resources.

In previous releases, the default for CIRCUITS was the maximum number of database sessions (SESSIONS) if shared server mode was enabled, 0 otherwise.

New Default Value for MAX_DISPATCHERS

Starting with Oracle Database 10*g* Release 1 (10.1), there is no preset default for MAX_DISPATCHERS. MAX_DISPATCHERS no longer limits the number of dispatchers; the user can increase the number of dispatchers with the DISPATCHERS parameter as long as there are free process slots and system resources.

In previous releases, the default for MAX_DISPATCHERS was 5, or the total number of dispatchers specified with the DISPATCHERS parameter, whichever was greater.

B

Gathering Optimizer Statistics for Upgrading Oracle Database

Oracle provides scripts that collect optimizer statistics for dictionary objects in Oracle Database. By running these scripts before 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 just like any other aspect of the upgrade. Also, some schemas referenced in these scripts might not exist if some database components have not been installed.

This appendix contains the following topics:

- Collecting Statistics for System Component Schemas
- Creating a Statistics Table

Collecting Statistics for System Component Schemas

If you are using Oracle9*i* Release 2 (9.2), then you should use the DBMS_STATS.GATHER_SCHEMA_STATS procedure to gather statistics. The following sample script uses this procedure to collect statistics for system component schemas.

To run this script, connect to the database AS SYSDBA using SQL*Plus.

```
spool gdict

grant analyze any to sys;

exec dbms_stats.gather_schema_stats('WMSYS',options=>'GATHER', -
estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, -
method_opt => 'FOR ALL COLUMNS SIZE AUTO', cascade => TRUE);

exec dbms_stats.gather_schema_stats('MDSYS',options=>'GATHER', -
estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, -
method_opt => 'FOR ALL COLUMNS SIZE AUTO', cascade => TRUE);

exec dbms_stats.gather_schema_stats('CTXSYS',options=>'GATHER', -
estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, -
method_opt => 'FOR ALL COLUMNS SIZE AUTO', cascade => TRUE);

exec dbms_stats.gather_schema_stats('XDB',options=>'GATHER', -
estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, -
method_opt => 'FOR ALL COLUMNS SIZE AUTO', cascade => TRUE);

exec dbms_stats.gather_schema_stats('WKSYS',options=>'GATHER', -
estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, -
method_opt => 'FOR ALL COLUMNS SIZE AUTO', cascade => TRUE);

exec dbms_stats.gather_schema_stats('LBACSYS',options=>'GATHER', -
estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, -
method_opt => 'FOR ALL COLUMNS SIZE AUTO', cascade => TRUE);

exec dbms_stats.gather_schema_stats('ORDSYS',options=>'GATHER', -
estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, -
method_opt => 'FOR ALL COLUMNS SIZE AUTO', cascade => TRUE);

exec dbms_stats.gather_schema_stats('ORDPLUGINS',options=>'GATHER', -
estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, -
method_opt => 'FOR ALL COLUMNS SIZE AUTO', cascade => TRUE);

exec dbms_stats.gather_schema_stats('SI_INFORMTN_SCHEMA',-
options=>'GATHER', estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, -
method_opt => 'FOR ALL COLUMNS SIZE AUTO', cascade => TRUE);

exec dbms_stats.gather_schema_stats('OUTLN',options=>'GATHER', -
estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, -
method_opt => 'FOR ALL COLUMNS SIZE AUTO', cascade => TRUE);

exec dbms_stats.gather_schema_stats('DBSNMP',options=>'GATHER', -
estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, -
method_opt => 'FOR ALL COLUMNS SIZE AUTO', cascade => TRUE);

exec dbms_stats.gather_schema_stats('SYSTEM',options=>'GATHER', -
estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, -
method_opt => 'FOR ALL COLUMNS SIZE AUTO', cascade => TRUE);

exec dbms_stats.gather_schema_stats('SYS',options=>'GATHER', -
estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE, -
method_opt => 'FOR ALL COLUMNS SIZE AUTO', cascade => TRUE);

spool off
```

Note: The statistics collection might give errors 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.

Creating a Statistics Table

This script creates the table, `dictstattab`, and exports the statistics for the RDBMS component schemas into it. The export returns an error 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.

This script is useful when you want to import the statistics back into the database. For example, the following PL/SQL subprograms import the statistics for the `SYS` schema after deleting the existing statistics:

```
EXEC DBMS_STATS.DELETE_SCHEMA_STATS('SYS');
EXEC DBMS_STATS.IMPORT_SCHEMA_STATS('SYS','dictstattab');
```

To run the following script, connect to the database `AS SYSDBA` using SQL*Plus.

```
spool sdict
```

```
grant analyze any to sys;

exec dbms_stats.create_stat_table('SYS','dictstattab');

exec dbms_stats.export_schema_stats('WMSYS','dictstattab',statown => 'SYS');
exec dbms_stats.export_schema_stats('MDSYS','dictstattab',statown => 'SYS');
exec dbms_stats.export_schema_stats('CTXSYS','dictstattab',statown => 'SYS');
exec dbms_stats.export_schema_stats('XDB','dictstattab',statown => 'SYS');
exec dbms_stats.export_schema_stats('WKSYS','dictstattab',statown => 'SYS');
exec dbms_stats.export_schema_stats('LBACSYS','dictstattab',statown => 'SYS');
exec dbms_stats.export_schema_stats('ORDSYS','dictstattab',statown => 'SYS');
exec dbms_stats.export_schema_stats('ORDPLUGINS','dictstattab',statown => 'SYS');
exec dbms_stats.export_schema_stats('SI_INFORMTN_SCHEMA','dictstattab',statown =>
'SYS');
exec dbms_stats.export_schema_stats('OUTLN','dictstattab',statown => 'SYS');
exec dbms_stats.export_schema_stats('DBSNMP','dictstattab',statown => 'SYS');
exec dbms_stats.export_schema_stats('SYSTEM','dictstattab',statown => 'SYS');
exec dbms_stats.export_schema_stats('SYS','dictstattab',statown => 'SYS');

spool off
```

Index