Oracle® Database

Upgrade Guide

11*g* Release 2 (11.2)

E10819-02

August 2009

ORACLE®

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

Primary Author: Tony Morales

Contributors: Thomas Baby, Michael Brey, Bridget Burke, Rae Burns, Vickie Carbonneau, Sunil Chakkappen, Ramesh Chakravarthula, Alan Choi, George Claborn, Ian Dai, Mark Drake, Mohammad Faisal, Mark Fallon, Craig B. Foch, Steve Fogel, Leonidas Galanis, Shie-rei Huang, Pat Huey, John Hwee, Rich Long, Venkat Maddali, Matthew McKerley, Mughees Minhas, Wataru Miyoshi, Valarie Moore, Louise Morin, Carol Palmer, Satish Panchumarthy, Robert Pang, Ravi Pattabhi, Naga Prakash, Irfan Rafiq, Mark Ramacher, Kathy Rich, Tsuyoshi Sano, Viv Schupmann, Ranu Sharma, Janelle Simmons, Virender Singh, Jason Straub, Roy Swonger, Carol Tagliaferri, Venkateshwaran Venkataramani, Sam Wegner, Douglas Williams, Terri Winters, Mohamed Ziauddin

Contents

3   Upgrading to the New Release

7   Moving Data Using Data Pump and Export/Import

A   Behavior Changes

B   Gathering Optimizer Statistics

Index

x

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

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at `http://www.oracle.com/accessibility/`.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at `http://www.fcc.gov/cgb/consumerfacts/trs.html`, and a list of phone numbers is available at `http://www.fcc.gov/cgb/dro/trsphonebk.html`.

Related Documentation

For more information, see these Oracle resources:

- "The Upgrade Companion" Web site available in Note 785351.1 on My Oracle Support at `http://metalink.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 will minimize 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 Database Upgrade Process

This chapter provides an overview of the database upgrade process, as well as information about running multiple releases of Oracle Database.

This chapter covers the following topics:

- Overview of the Database Upgrade Process
- Oracle Release Numbers
- Compatibility and Interoperability
- Using Optimal Flexible Architecture (OFA)
- Converting Databases to 64-bit Oracle Database Software
- About Rolling Upgrades
- Moving From the Standard Edition to the Enterprise Edition
- Moving From the Enterprise Edition to the Standard Edition
- Upgrading from Oracle Database Express Edition to Oracle Database

> **Note:** For late-breaking updates and best practices about pre-upgrade, post-upgrade, compatibility, and interoperability discussions, see Note 785351.1 on My Oracle Support (`http://metalink.oracle.com/`) that links to "The Upgrade Companion" Web site.

Overview of the Database Upgrade Process

This section includes an overview of the major steps required to upgrade an existing Oracle database to the new Oracle Database 11*g* release. These procedures transform an existing Oracle Database system (including associated applications) into an Oracle Database 11*g* Release 2 (11.2) system. Oracle Database 11*g* Release 2 (11.2) is compatible with all earlier Oracle Database releases. Therefore, databases upgraded using the procedures described in this guide can work in the same manner as in earlier releases and, optionally, can leverage new Oracle Database 11*g* features and functions.

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 (GUI) 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

  Provides a command-line upgrade of a database, using SQL scripts and utilities.

- Export and Import utilities

  Uses the Oracle Data Pump Export and Import utilities, available as of Oracle Database 10*g*, or 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, 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.

These tools and methods are described in more detail in "Choose an Upgrade Method" on page 2-3.

The major steps in the upgrade process are illustrated in Figure 1–1.

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      │
└─────────────────────────────┘
```
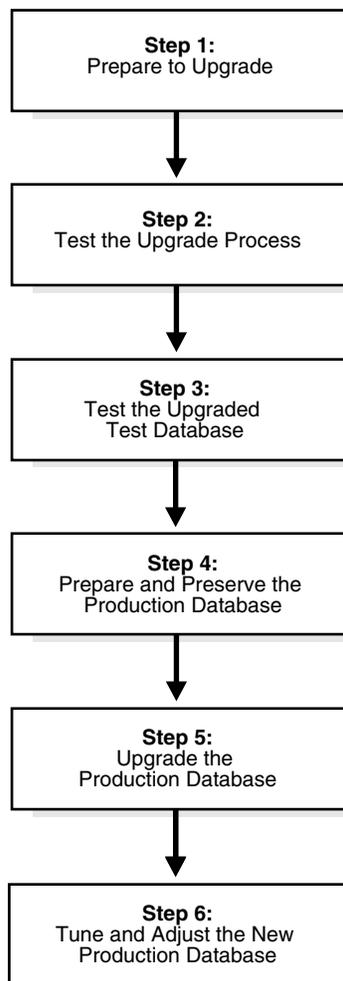
> **Note:**   The upgrade steps apply to all operating systems, with the possible exception of a few operating system-specific details identified in your operating system-specific Oracle documentation.

The following list outlines the major steps performed during the upgrade process. Detailed instructions are provided in the appropriate chapters later in this guide.

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.

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" 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 describes Steps 4 and 5 when using DBUA or when performing a manual upgrade. Chapter 4 describes the backup procedure after the upgrade and other post-upgrade tasks.

Step 6: Tune and Adjust the New Production Database

- Tune the new Oracle Database 11*g* production database. The new production database should perform as well as, or better than, the database prior to the upgrade. Chapter 4, "After Upgrading to the New Release" 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 Your Applications" describes considerations for updating applications.

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 "Running Multiple Oracle Releases" on page 1-5.

Oracle Release Numbers

This guide describes moving between different releases of the Oracle Database server. Figure 1–2 describes what each part of a release number represents.

Figure 1–2   Example of an Oracle Release Number
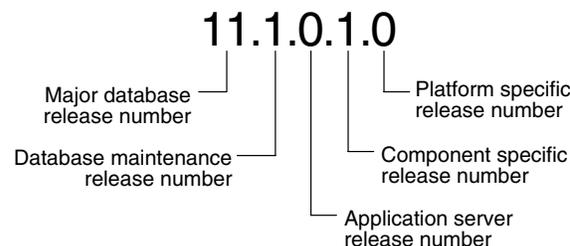
> **Note:**   Starting with Oracle9*i* Release 2 (9.2), maintenance releases of Oracle Database are denoted by a change to the second digit of a release number. In previous releases, the third digit indicated a particular maintenance release.

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

When a statement is made in this 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); references to Oracle9*i* include Oracle9*i* Release 1 (9.0.1) and Oracle9*i* Release 2 (9.2).

Similarly, when a statement is made in this guide about a maintenance release, the statement applies to all component-specific (also referred to as patchset releases) and platform-specific releases within that maintenance release. So, a statement about Oracle9*i* Release 2 (9.2) applies to release 9.2.0.1, release 9.2.0.2, and all other platform-specific releases within Oracle9*i* Release 2 (9.2).

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 using multiple releases:

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

   There cannot be more than one release for each Oracle home. Oracle recommends that you adopt an Optimal Flexible Architecture (OFA) when creating multiple Oracle homes. See "Using Optimal Flexible Architecture (OFA)" on page 1-9 for more information.

> **Caution:** It is not possible to install Oracle Database 11*g* products into an existing Oracle home of a prior major release. This feature was available only for certain previous releases and has not been continued.

- 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. Restrictions may apply on some operating systems.

The following sections provide general information about running multiple releases of Oracle Database.

Install Databases in Multiple Oracle Homes on the Same Computer

You can install Oracle8*i*, Oracle9*i*, Oracle Database 10*g*, and Oracle Database 11*g* databases in multiple (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.

Install 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.

Upgrade a Database 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.

Upgrade Clients to the Current Release

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 be used to access your Oracle8*i*, Oracle9*i*, Oracle Database 10*g*, and Oracle Database 11*g* databases.

Compatibility and Interoperability

This section describes compatibility and interoperability issues that may arise because of differences between Oracle Database releases. These differences might affect general database administration and existing applications.

This section covers the following topics:

- What Is Compatibility?

- What Is Interoperability?

What Is Compatibility?

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

- A new feature stores any data on disk (including data dictionary changes) that cannot be processed with your previous release.

- An existing feature behaves differently in the new environment as compared to the old environment. This type of incompatibility is classified as a **language incompatibility**.

The COMPATIBLE Initialization Parameter

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.

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

Downgrading 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`, which is the lowest possible setting for Oracle Database 11*g* Release 2 (11.2). 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 previous release.

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

- Set to `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)

> **See Also:** [Chapter 6, "Downgrading a Database"](#) for more information about downgrading

How the COMPATIBLE Initialization Parameter Operates 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

Compatibility Level 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 `11.0.0`, then the database runs at 11.0.0 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 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 do this, the database cannot subsequently be downgraded.

What Is Interoperability?

Interoperability is the ability of different releases of Oracle Database to communicate and work together in a distributed environment. A distributed database system can have different releases of Oracle Database, and all supported releases of Oracle Database can participate in a distributed database system. However, the applications that work with a distributed database must understand the features and functions that are available at each node in the system.

Interoperability across 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. This means that you must check drivers, network, and storage compatibilities for all the interim states you will be in during the rolling upgrade.

> **Note:** Because this guide documents upgrading and downgrading between different releases of Oracle Database, this definition of interoperability is appropriate. However, other Oracle Database 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:** Appendix A, "Behavior Changes" for more information on compatibility and interoperability

Using Optimal Flexible Architecture (OFA)

Oracle recommends the Optimal Flexible Architecture (OFA) standard for your Oracle Database installations. The OFA standard is a set of configuration guidelines for efficient and reliable Oracle databases that require little 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 relocating database files

Converting Databases to 64-bit Oracle Database Software

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

However, you must perform the following manual 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:** Your platform-specific installation guide for additional information about 64-bit software installations

About Rolling Upgrades

A rolling upgrade upgrades different databases, database instances of the same database, or 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 ASM instances, apply patchsets, or apply individual patches (sometimes referred to as one-off patches), incurring little or no database downtime.

Table 1–2 summarizes the various methods for performing rolling upgrades. Also, see *Oracle Database High Availability Best Practices* for help choosing a method to perform database upgrades.

Table 1–2   Methods for Performing Rolling Upgrades

Method	Description	Reference
Oracle Data Guard SQL Apply	Use SQL Apply and logical standby databases to upgrade Oracle Database software and patchsets.	"Upgrading Using Standby Databases" on page 3-4
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.	"Upgrading Using Standby Databases" on page 3-4
Oracle Streams	Use 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 or character set, upgrade user-created applications, and apply Oracle Database patches.	*Oracle Streams Concepts and Administration*

Table 1–2   (Cont.) Methods for Performing Rolling Upgrades

Method	Description	Reference
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 or Oracle Cluster Ready Services. 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*
Automatic Storage Management (ASM)	Use ASM to independently upgrade or patch clustered ASM instances. This method allows all of the features of a clustered ASM environment to continue to function even while one or more ASM instances are running different software versions. **Note:** An ASM rolling upgrade to Oracle Database 11*g* Release 2 (11.2) moves the ASM instance to an Oracle grid infrastructure home.	*Oracle Grid Infrastructure Installation Guide* "Using ASM Rolling Upgrade" in *Oracle Database Storage Administrator's Guide*

Moving From the Standard Edition to the Enterprise Edition

If you have a Standard Edition database at a release prior to the new Oracle Database 11*g* release, then you can change it to an Enterprise Edition database by installing the Enterprise Edition and following the normal upgrade procedures, as described in this guide.

To change your existing Standard Edition database to an Enterprise Edition database, follow these steps:

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

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

2. Shut down your database.

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

4. Deinstall the Standard Edition server software.

5. Install the Enterprise Edition server software using the Oracle Universal Installer.

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

6. Start up your database.

Your database is now upgraded to the Enterprise Edition.

Moving From the Enterprise Edition to the Standard Edition

You cannot use the procedure described in "Moving From the Standard Edition to the Enterprise Edition" on page 1-11 to convert an Enterprise Edition database to a Standard Edition database. The Enterprise Edition contains data dictionary objects which are not available in the Standard Edition. If you just install the Standard Edition software, then you will end up with data dictionary objects which are useless. Some of them might be invalid and possibly create problems when maintaining the database.

The only way to properly convert from an Enterprise Edition database to a Standard Edition database is through an Export/Import operation. The Export/Import operation does not introduce data dictionary objects specific to the Enterprise Edition, because the SYS schema objects are not exported. Oracle recommends using the Standard Edition Export utility to export the data.

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 Data Pump and Export/Import"

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 need to install Oracle Database 11*g* on the same system as Oracle Database XE and use the Database Upgrade Assistant to perform the upgrade.

2

Preparing to Upgrade

This chapter describes the steps to complete before upgrading a database to the new Oracle Database 11*g* release. This chapter covers in detail Steps 1 through 3 of the upgrade process that were outlined in "Overview of the Database Upgrade Process" on page 1-1.

This chapter covers the following topics:

- Prepare to Upgrade
- Test the Upgrade Process
- Test the Upgraded Test Database

> **See Also:** *Oracle Database Net Services Administrator's Guide* for information about upgrade considerations for Oracle Net Services

> **Note:** Some aspects of upgrading are operating system-specific. See your operating system-specific Oracle documentation for additional information about preparing to upgrade.

Prepare to Upgrade

Complete the following tasks to prepare to upgrade:

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

Become Familiar with New Oracle Database Features

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 set 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. Connect to the following web page for more information:
>
> http://education.oracle.com/

Determine the Upgrade Path

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 8.1.6, then follow these steps:

1. Upgrade release 8.1.6 to release 8.1.7.4 using the instructions in *Oracle8i Migration* Release 3 (8.1.7).

2. Upgrade release 8.1.7.4 to release 10.2.0.4 using the instructions in *Oracle Database Upgrade Guide* Release 2 (10.2).

3. 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   Upgrade Paths

Current Release	Upgrade Path
7.3.3 and lower 7.3.4 8.0.3 8.0.4 8.0.5 8.0.6 8.1.5 8.1.6 8.1.7.4 9.0.1.4	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: ■ 7.3.3 (or lower) -> 7.3.4 -> 9.2.0.8 -> 11.2 ■ 8.0.5 (or lower) -> 8.0.6 -> 9.2.0.8 -> 11.2 ■ 8.1.7 (or lower) -> 8.1.7.4 -> 10.2.0.4 -> 11.2 ■ 9.0.1.3 (or lower) -> 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".
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.x must be at release 10.2.0.3 (or higher), before you attempt to upgrade it to Oracle Clusterware 11*g*. See "Upgrading an Oracle Real Application Clusters (Oracle RAC) Database" on page 3-2. For release 9.2.0.3, you must first upgrade to an intermediate Oracle Database release, as follows: 9.2.0.3 (or lower) -> 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".

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

Choose an Upgrade Method

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

- Database Upgrade Assistant
- Manual Upgrade
- Export/Import

Database Upgrade Assistant

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 When manually upgrading a database, perform the following pre-upgrade steps:

- Analyze the database using the Pre-Upgrade Information Tool. The Pre-Upgrade Information Tool is a SQL script that ships with the new Oracle Database 11*g* release, and must be run in the environment of the database being upgraded.

  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.
- 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 and adjust initialization parameters that might cause upgrade problems).

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, and should be run in the environment of the new release.

> **See Also:** "Upgrade a Database Manually" on page 3-34

Export/Import

Unlike DBUA or a manual upgrade, the Export/Import utilities physically copy data from your current database to a new database. You can use either the Oracle Data Pump Export and Import utilities, available as of Oracle Database 10*g* Release 1 (10.1), or 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.

When upgrading from Oracle Database 10*g* Release 1 (10.1) or higher, Data Pump Export and Import are recommended for higher performance.

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 already 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.

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 Export/Import offers the following benefits:

- 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 - you can import only the objects, users, and other items that you wish. This is useful for establishing a test environment for the new software on only a subset of the production data. Data Pump Export / Import provides very flexible data subsetting capabilities.

- Serves as a backup archive - you can use a full database export as an archive of the current database.

- Allows the new database to be established on an Operating System or hardware platform that is different from that 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 Data Pump and Export/Import"

Choose an Oracle Home Directory

You must choose an Oracle home directory for the new Oracle Database 11*g* release that is separate from the Oracle home directory of your current release. You cannot install the new software into the same Oracle home directory 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

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 test program.

Whether you use Real Application Testing features like Database Replay or SQL Performance Analyzer, or perform testing manually, your test plan must include the following types of tests:

- Upgrade Testing
- Minimal Testing
- Functional Testing
- High Availability Testing
- Integration Testing
- Performance Testing
- Volume and Load Stress Testing

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

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

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

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 will help evaluate if the Oracle RAC recovery capability has changed.

- Testing your fallback plans and procedures, if relevant.

- 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://metalink.oracle.com/`.

Integration Testing

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 Net8 or Oracle Net Services, then those connections should also be tested and stress tested.

Performance Testing

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

> **Note:** Automatic Workload Repository is not supported for Oracle9*i* Release 2 (9.2). If you are upgrading from Oracle9*i* Release 2 (9.2), then use the SQL trace facility and profile your application with TKPROF.

> **See Also:** *Oracle Database Performance Tuning Guide* for more information on Automatic Workload Repository

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 performance monitoring and reporting tools such as ADDM, AWR, and ASH reports (in addition to the replay report) are at your disposal to remedy 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 prior to 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 allows you to evaluate the impact of a change such as database upgrade by identifying the SQL statements impacted by the upgrade and measuring their performance divergence. This 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 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. This means that if you seed SQL plan management with your current (pre-Oracle Database 11*g*) execution plan, which will become the SQL plan baseline for each statement, the optimizer uses these plans after the upgrade. If the Oracle Database 11*g* optimizer determines that a different plan should be used, then the new plan is queued for verification and will not be 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. Use one of the following techniques 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 2-8.
- Unpack existing SQL plan baselines from a staging table, as described in "Unpacking Existing SQL Plan Baselines from a Staging Table" on page 2-8.

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

Perform the following steps 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 SQL Plan Baselines from a Staging Table

Perform the following steps 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:

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

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. Database Replay feature is ideal for such load testing as it allows you to capture system workload from a production environment and replay it in identical fashion on the test system.

Prepare a Backup Strategy

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 should be used 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

Test the Upgrade Process

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, is performed on an exact copy of the database to be upgraded, rather than on a downsized copy or test data.

> **Caution:** Do not upgrade the actual production database until after you successfully upgrade a test subset of this database and test it with applications, as described in the next step.

Make sure 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 "Upgrading Precompiler and OCI Applications" on page 5-2 for more information.

Test the Upgraded Test 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 make sure that the applications operate in the same manner as they did in the current database.

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

3

Upgrading to the New Release

This chapter guides you through the process of upgrading a database to the new Oracle Database 11*g* release. The upgrade processes discussed in this chapter do not describe how to perform rolling upgrades. See "About Rolling Upgrades" on page 1-10 for more information about rolling upgrades.

This chapter covers the following topics:

- System Considerations and Requirements
- Install the New Oracle Database Software
- Install the Latest Available Patchset Release and Any Required Patches
- Run the Pre-Upgrade Information Tool
- Run the Oracle Net Configuration Assistant
- Upgrade a Database Using Database Upgrade Assistant
- Upgrade a Database Manually
- Upgrade an ASM Instance

> **WARNING:** 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.
>
> Also, before you remove the old Oracle environment, make sure you relocate any data files in that environment to the new Oracle Database environment. If you upgrade with Database Upgrade Assistant (DBUA), then you can do this automatically by selecting the Move Database Files option during the upgrade. See "Upgrade a Database Using Database Upgrade Assistant" on page 3-19 for more information. If you perform a manual upgrade, then see *Oracle Database Administrator's Guide* for information about relocating data files.

> **See Also:** Some aspects of upgrading are operating system-specific. See your operating system-specific Oracle documentation for additional instructions about upgrading on your operating system.

System Considerations and Requirements

The following sections discuss system considerations and requirements:

- Oracle ASM is Installed with Oracle Grid Infrastructure
- Upgrading Oracle Clusterware and Automatic Storage Management (ASM) Databases
- Upgrading with Read-Only and Offline Tablespaces
- Upgrading Using Standby Databases
- Upgrading Your Operating System
- Migrating Data to a Different Operating System
- Upgrading Databases That Use Oracle Streams Downstream Capture
- Upgrading Databases That Use Oracle Database Vault

Oracle ASM is Installed with Oracle Grid Infrastructure

In past releases, Oracle ASM was installed as part of the Oracle Database installation. With Oracle Database 11*g* Release 2 (11.2), Oracle ASM is part of an Oracle grid infrastructure installation. If you want to upgrade an existing Oracle ASM installation, then you must upgrade Oracle ASM by running an Oracle grid infrastructure upgrade.

Upgrading Oracle Clusterware and Automatic Storage Management (ASM) Databases

Prior to Oracle Database 11*g*, either all Oracle software installations were owned by the Oracle user, typically `oracle`, or Oracle Database software was owned by `oracle`, and Oracle Clusterware software was owned by a separate user, typically `crsuser`. Starting with Oracle Database 11*g*, the same user that owned the Oracle Clusterware 10*g* software must perform the Oracle Clusterware 11*g* upgrade.

Starting with Oracle Database 11*g* release 2 (11.2), the Oracle Clusterware software must be upgraded to a new directory, called the Oracle grid infrastructure home. Also, ASM and Oracle Clusterware (and Oracle Restart for single-instance databases) must run in the same Oracle grid infrastructure home.

For a single-instance configuration, ASM and Oracle Restart run from the Oracle grid infrastructure home, and the cluster synchronization services daemon (CSS) runs from this same home. Thus, ASM and CSS will be upgraded to Oracle Database 11*g* Release 2 (11.2) at the same time.

If your configuration does not include ASM, then you should shut down the CSS daemon and delete the CSS service from the system by running the `localconfig` batch file with the `delete` option. For example:

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

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

Upgrading an Oracle Real Application Clusters (Oracle RAC) Database

If you are upgrading an Oracle RAC database with DBUA, then all necessary configuration is for you automatically. If you are manually upgrading an Oracle RAC database, then most of the actions described in this chapter should be performed

on only one node of the system. Actions that must be performed on more than one node are indicated in that particular step.

> **Note:** A new prerequisite check has been added to ensure that Oracle Clusterware release 10.2.0.*n* is at release 10.2.0.3 (or higher), before you attempt to upgrade it to Oracle Clusterware 11*g*. If this check fails, then you are instructed to apply Oracle Clusterware patch set release 10.2.0.3.0 or later to your existing release 10.2.0.2 before it can be upgraded. All other upgrade paths and fresh install cycles are unaffected by this prerequisite check.

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 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 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 ASM instance with SYSASM privileges.

Before upgrading an 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 ASM instance using the SQL*Plus CREATE USER statement:

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

> **Note:** This step is necessary only when upgrading the ASM instance. It is not necessary when upgrading a database to the new Oracle Database 11*g* release without upgrading ASM.
>
> If the default Oracle Database 11*g* security settings are in place, then passwords must be at least 8 characters, and passwords such as welcome and oracle are not allowed. See *Oracle Database Security Guide* for more information.

> **See Also:**
>
> - *Oracle Database Storage Administrator's Guide* for more information about authentication for accessing 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

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 prior to 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

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 patchset release 10.1.0.*n* to the next database 10.1.0.(*n*+1) patchset 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.otn.oracle.com/goto/maa`:
  >
  >   "Rolling Database Upgrades using Data Guard SQL Apply"
  >
  >   "Rolling Database Upgrades for Physical Standby Databases Using Transient Logical Standby 11*g*"

Upgrading Your Operating System

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 Database 11*g* database on Windows 2000 using DBUA.

If you must migrate Oracle Database software to a different operating system, then the best practice is to follow these steps:

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 (`http://metalink.oracle.com/`) for more information on Data Guard support for heterogeneous primary and standby systems in the same 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 endianess 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 Data Pump and Export/Import"

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.

Upgrading Databases That Use Oracle Database Vault

If you have enabled Oracle Database Vault, then you must disable it before upgrading the database, and enable it again when the upgrade is finished.

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

Install the New Oracle Database Software

Installation of the Oracle Database 11*g* Release 2 (11.2) software involves upgrading Clusterware if necessary, and installing Oracle Database 11*g* Release 2 (11.2).

> **Note:** Oracle Clusterware was called Cluster Ready Services in earlier releases.

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

   a. Mount the Oracle grid infrastructure installation media.

   b. Complete 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".

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

   d. 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.

e. 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 (assuming it was necessary), follow the instructions in your Oracle operating system-specific documentation to prepare for installation of Oracle Database software and start the Oracle Universal Installer.

When installation is complete, one or more assistants might be started. If you chose to run DBUA during installation, then you are ready to proceed with the upgrade when DBUA is started. However, 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 "Run the Pre-Upgrade Information Tool" on page 3-8.) You can then run DBUA independently after the installation is complete.

Note also that you must run the Oracle Net Configuration Assistant before running DBUA.

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

Note: If you use Oracle Label Security, Database Vault, or both, then you must perform a custom Oracle Database 11*g* Release 2 (11.2) installation, because these features are not included in the standard Oracle Database installation. See *Oracle Label Security Administrator's Guide* or your platform-specific *Oracle Database Vault Administrator's Guide* for more information.

See Also:

- "Run the Oracle Net Configuration Assistant" on page 3-18

- "Upgrade a Database Using Database Upgrade Assistant" on page 3-19

Install the Latest Available Patchset Release and Any Required Patches

After you have installed Oracle Database 11*g* Release 2 (11.2), check to see if there is a patchset release or critical patch update to be installed:

- Patchsets

    Patchsets are software-release mechanisms for delivering tested and integrated product fixes on a regular basis.

- Critical patch updates

    Critical patch updates contain security patch updates and security fixes. As of 2005, Oracle began providing critical patch updates for all product offerings on a quarterly schedule. The comprehensive patches address significant security vulnerabilities and include fixes that are prerequisites for the security fixes.

The latest patchset release and critical patch update for Oracle Database 11*g* Release 2 (11.2) should be installed prior to upgrading your databases. Refer to the specific patch release and critical patch update documentation for installation information.

Run the Pre-Upgrade Information Tool

After you have installed Oracle Database 11*g* Release 2 (11.2) and any required patches, you should analyze your database before upgrading it to the new release. This is done by running the Pre-Upgrade Information Tool. This is a required step if you are upgrading manually, otherwise the `catupgrd.sql` script will terminate 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.

The Pre-Upgrade Information Tool is a SQL script that ships with Oracle Database 11*g* Release 2 (11.2), and must be copied to and run from the environment of the database being upgraded. Complete the following steps to run the Pre-Upgrade Information Tool:

1. Log in to the system as the owner of the Oracle Database 11*g* Release 2 (11.2) Oracle home directory.

2. Copy the Pre-Upgrade Information Tool (`utlu112i.sql`) from the Oracle Database 11*g* Release 2 (11.2) *ORACLE_HOME*/`rdbms/admin` 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.

3. Log in to the system as the owner of the Oracle home directory of the database to be upgraded.

4. Change to the directory that you copied `utlu112i.sql` to in Step 2.

5. Start SQL*Plus.

6. Connect to the database instance as a user with `SYSDBA` privileges.

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

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

8. Run the Pre-Upgrade Information Tool:

   ```
   SQL> @utlu112i.sql
   ```

9. 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`.

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

```
Oracle Database 11.2 Pre-Upgrade Information Tool    10-14-2008 23:25:25
.
*******************************************************************
Database:
*******************************************************************
--> name:          ORCL
--> version:       10.2.0.2.0
--> compatible:    10.2.0.2
--> blocksize:     8192
```

```
--> platform:      Linux IA (32-bit)
--> timezone file: V2
.
*************************************************************************
Tablespaces: [make adjustments in the current environment]
*************************************************************************
WARNING: --> SYSTEM tablespace is not large enough for the upgrade.
.... currently allocated size: 560 MB
.... minimum required size: 910 MB
.... increase current size by: 350 MB
.... tablespace is NOT AUTOEXTEND ENABLED.
--> UNDOTBS1 tablespace is adequate for the upgrade.
.... minimum required size: 457 MB
.... AUTOEXTEND additional space required: 352 MB
--> SYSAUX tablespace is adequate for the upgrade.
.... minimum required size: 617 MB
.... AUTOEXTEND additional space required: 287 MB
--> TEMP tablespace is adequate for the upgrade.
.... minimum required size: 61 MB
.... AUTOEXTEND additional space required: 41 MB
--> EXAMPLE tablespace is adequate for the upgrade.
.... minimum required size: 69 MB
.
*************************************************************************
Update Parameters: [Update Oracle Database 11.2 init.ora or spfile]
*************************************************************************
WARNING: --> "sga_target" needs to be increased to at least 388 MB
.
*************************************************************************
Renamed Parameters: [Update Oracle Database 11.2 init.ora or spfile]
*************************************************************************
WARNING: --> "plsql_compiler_flags" old value was "INTERPRETED";
new name is "plsql_code_type" new value is "INTERPRETED"
.
*************************************************************************
Obsolete/Deprecated Parameters: [Update Oracle Database 11.2 init.ora or spfile]
*************************************************************************
--> "max_enabled_roles"
--> "remote_os_authent"
--> "background_dump_dest" replaced by "diagnostic_dest"
--> "user_dump_dest" replaced by "diagnostic_dest"
.
*************************************************************************
Components: [The following database components will be upgraded or installed]
*************************************************************************
--> Oracle Catalog Views        [upgrade]  VALID
--> Oracle Packages and Types    [upgrade]  VALID
--> JServer JAVA Virtual Machine [upgrade]  VALID
--> Oracle XDK for Java          [upgrade]  VALID
--> Oracle Workspace Manager     [upgrade]  VALID
--> Messaging Gateway            [upgrade]  VALID
--> OLAP Analytic Workspace      [upgrade]  VALID
--> OLAP Catalog                 [upgrade]  VALID
--> Oracle Label Security        [upgrade]  VALID
--> 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
```

```
--> Data Mining                [upgrade]  VALID
--> Expression Filter          [upgrade]  VALID
--> Rule Manager               [upgrade]  VALID
--> Oracle Application Express [upgrade]
--> Oracle OLAP API            [upgrade]  VALID
.
**********************************************************************
Miscellaneous Warnings
**********************************************************************
WARNING: --> Database is using a timezone file older than version 11.
.... After the release migration, it is recommended that DBMS_DST package
.... be used to upgrade the 10.2.0.2.0 database timezone version
.... to the latest version which comes with the new release.
WARNING: --> Database contains stale optimizer statistics.
.... Refer to the 11g Upgrade Guide for instructions to update
.... statistics prior to upgrading the database.
.... Component Schemas with stale statistics:
....    SYS
....    WMSYS
....    CTXSYS
WARNING: --> Database contains INVALID objects prior to upgrade.
.... The list of invalid SYS/SYSTEM objects was written to
.... registry$sys_inv_objs.
.... The list of non-SYS/SYSTEM objects was written to
.... registry$nonsys_inv_objs.
.... Use utluiobj.sql after the upgrade to identify any new invalid
.... objects due to the upgrade.
.... USER PUBLIC has 7 INVALID objects.
.... USER FLOWS_010600 has 1 INVALID objects.
.... USER SYS has 1 INVALID objects.
WARNING: --> Database contains schemas with objects dependent on network packages.
.... Refer to the 11g Upgrade Guide for instructions to configure Network ACLs.
.... USER WKSYS has dependent objects.
.... USER SYSMAN has dependent objects.
.... USER FLOWS_010600 has dependent objects.
WARNING: --> EM Database Control Repository exists in the database.
.... Direct downgrade of EM Database Control is not supported. Refer to the
.... 11g Upgrade Guide for instructions to save the EM data prior to upgrade.
.
WARNING: --> There are materialized view refreshes in progress.
.... Ensure all materialized view refreshes are complete prior to upgrade.
WARNING: --> There are files which need media recovery.
.... Ensure no files need media recovery prior to upgrade.
WARNING: --> There are files in backup mode.
.... Ensure no files are in backup mode prior to upgrade.
WARNING:--> There are outstanding unresolved distributed transactions.
.... Resolve outstanding distributed transactions prior to upgrade.
WARNING:--> A standby database exists.
.... Sync standby database prior to upgrade.
WARNING: --> log_archive_format must be updated.
.... As of 10.1, log_archive_format requires a %r format qualifier
.... be present in its format string.  Your current setting is:
.... log_archive_format='%t_%s.dbf'.
.... Archive Logging is currently OFF, but failure to add the %r to the
.... format string will still prevent the upgraded database from starting up.
WARNING:--> recycle bin in use.
.... Your recycle bin is turned on and it contains
.... 3 object(s).  It is REQUIRED
.... that the recycle bin is empty prior to upgrading
.... your database.
```

```
.... The command:  PURGE DBA_RECYCLEBIN
.... must be executed immediately prior to executing your upgrade.
.
```

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

The following sections describe the output of the Pre-Upgrade Information Tool.

Database

This section displays global database information about the current database, such as the database name, release number, and compatibility level. A warning is displayed if you must adjust the COMPATIBLE initialization parameter before the database is upgraded.

Logfiles

This section displays a list of redo log files in the current database whose size is less than 4 MB. For each log file, the file name, group number, and recommended size is displayed.

In a manual upgrade using SQL scripts and utilities, new files of at least 4 MB (preferably 10 MB) must be created in the current database, and any redo log files less than 4 MB must be dropped before the database is upgraded. These tasks are performed automatically by the Database Upgrade Assistant.

Tablespaces

This section displays a list of tablespaces in the current database. For each tablespace, the tablespace name and minimum required size is displayed. In addition, a message is displayed if the tablespace is adequate for the upgrade.

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 the Database Upgrade Assistant.

Update Parameters

This section displays a list of initialization parameters in the parameter file of the current database that must be adjusted before the database is upgraded. The adjustments must be made to the parameter file after it is copied to the new Oracle Database 11*g* release.

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

Deprecated Parameters

This section displays a list of initialization parameters in the parameter file of the current database that are deprecated in the new Oracle Database 11*g* release.

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

Obsolete Parameters

This section displays a list of initialization parameters in the parameter file of the current database that are obsolete 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" for a list of initialization parameters that are obsolete in the new Oracle Database 11*g* release

Components

This section displays a list of database components in the new Oracle Database 11*g* release that are upgraded or installed when the current database is upgraded.

Miscellaneous Warnings

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

SYSAUX Tablespace

This section displays the minimum required size for the `SYSAUX` tablespace, which is required in the new Oracle Database 11*g* release. The `SYSAUX` tablespace must be created after the new release is started and *before* the upgrade scripts are invoked.

Pre-Upgrade Information Tool Miscellaneous Warnings

If the Pre-Upgrade Information Tool displays a warning about any of the following issues, then further analysis of the database is recommended prior to upgrading it to the new Oracle Database 11*g* release:

- Deprecated CONNECT Role
- Access Control to Network Utility Packages
- Database Links with Passwords
- TIMESTAMP WITH TIME ZONE Data Type
- Optimizer Statistics
- Invalid Objects
- Save Oracle Enterprise Manager Database Control Data
- Complete Materialized View Refreshes
- Ensure No Files Need Media Recovery
- Ensure No Files Are in Backup Mode
- Resolve Outstanding Distributed Transactions
- Sync Standby Database with the Primary Database
- Purging the Database Recycle Bin

Deprecated CONNECT Role

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, use 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');
```

If users or roles require privileges other than CREATE SESSION, then grant the specific required privileges prior to upgrading. The upgrade scripts adjust the privileges for the Oracle-supplied users.

Access Control to 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, or 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 already installed.

Use the following instructions to assess the dependencies and provide access by adding the appropriate access control lists (ACLs):

1. Run the pre-upgrade information tool as described in "Run the Pre-Upgrade Information Tool" on page 3-8.

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')
      AND owner NOT IN ('SYS','PUBLIC','ORDPLUGINS');
    ```

4. Prepare post-upgrade scripts now so the scripts will be 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 "Configure Fine-Grained Access to External Network Services" on page 4-3. 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 will need to grant the specific required privileges. Access will be based on the usage in the original database.

Database Links with Passwords

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 prior to the downgrade. Consequently, the database links do not exist 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.

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 version 11 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 prior releases. If the time zone file is not version 11 on the database being upgraded, then the Pre-Upgrade Information Tool displays a warning and describes how to proceed. Table 3–1 describes the warnings and summarizes how to rectify a mismatch in time zone file versions.

Table 3–1   Fixing the Time Zone File Version

IF the version on the database being upgraded is ...	**THEN fix the time zone files ...**
Earlier than version 11 and the Pre-Upgrade Information Tool displays "Database is using a timezone 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*.
Later than version 11 and the Pre-Upgrade Information Tool displays "Database is using a timezone file greater than version *n*."	*Before* beginning the database upgrade. You must patch the Oracle home with the appropriate patch for the time zone file version in use. Apply the patch for each database you will be upgrading. Otherwise, the upgrade script will terminate without upgrading the database.

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.

> **See Also:** *Oracle Database Globalization Support Guide* for a detailed description of time zone upgrade, and the following My Oracle Support notes:
>
> ■ Note 359145.1 "Impact of 2007 USA daylight saving changes on the Oracle database"
>
> ■ Note 414590.1 "Time Zone IDs for 7 Time Zones Changed in Time Zone Files >=V3"

Optimizer Statistics

When upgrading to the new Oracle Database 11*g* release, optimizer statistics are collected for dictionary tables that lack statistics. This statistics collection can be time consuming for databases with a large number of dictionary tables, 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, you can collect statistics prior to performing the actual database upgrade. As of Oracle Database 10*g* Release 1 (10.1), Oracle recommends that you use the `DBMS_STATS.GATHER_DICTIONARY_STATS` procedure to gather these statistics. For example, you can enter the following:

```
EXEC DBMS_STATS.GATHER_DICTIONARY_STATS;
```

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.

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`
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 Ultra Search	`WKSYS, WKPROXY,WK_TEST`
Oracle Workspace Manager	`WMSYS`
Oracle XDK	`SYS`
Oracle XML Database	`XDB`

Invalid Objects

Any invalid `SYS/SYSTEM` objects found prior to upgrading the database will be stored in the table named `registry$sys_inv_objs`. Any invalid non-`SYS/SYSTEM` objects found prior to upgrading the database will be stored in `registry$nonsys_inv_objs`. After the upgrade, run `ORACLE_HOME/rdbms/admin/utluiobj.sql` to identify any new invalid objects due to the upgrade.

Save Oracle Enterprise Manager Database Control Data

In order to downgrade Oracle Enterprise Manager Database Control after upgrading to the new Oracle Database 11*g* release, you must save your Database Control files and

data before upgrading your database. This section explains how to use the `emdwgrd` utility 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 new Oracle Database 11*g* release and invoke the script from the new Oracle Database 11*g* release. The `emdwgrd` utility, however, requires that you set `ORACLE_HOME` to the old Oracle home.

The following procedure is for Linux and UNIX. To run it on Windows, simply substitute `emdwgrd.bat` for `emdwgrd`.

Follow these steps to save your Database Control files and data:

1. Install the new Oracle Database 11*g* release.

   This step is not required for an inplace patchset upgrade.

2. Set `ORACLE_HOME` to your old Oracle home.

   This step is not required for an inplace patchset 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:

   - 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
     ```

   - 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.

   On a single-instance database, the `emdwgrd` utility produces output similar to the following:

   ```
   Sat Apr 28 08:49:45 2007 - Verify EM DB Control files ... pass
   Sat Apr 28 08:49:45 2007 - Validating DB Connection to DB102 ... pass
   Sat Apr 28 08:49:51 2007 - creating directory ... created
   Sat Apr 28 08:49:51 2007 - Stopping DB Control ... stopped
   Sat Apr 28 08:50:01 2007 - Saving DB Control files ... saved
   Sat Apr 28 08:50:14 2007 - recompiling invalid objects ... recompiled
   Sat Apr 28 08:50:18 2007 - Exporting sysman schema for DB102 ... exported
   Sat Apr 28 08:51:36 2007 - Starting DB Control ... started
   ```

```
Sat Apr 28 08:53:21 2007 - DB Control was saved successfully.
```

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 —save  -cluster

Enter sys password for database DB102?
*****

Database Unique Name : DB102
Sat Apr 28 08:49:45 2007 - Verify EM DB Control files ... pass
Sat Apr 28 08:49:45 2007 - Validating DB Connection to DB102 ... pass
Sat Apr 28 08:49:51 2007 - creating directory ... created
Sat Apr 28 08:49:51 2007 - Stopping 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

... stopped
Sat Apr 28 08:50:01 2007 - Saving DB Control files
Executing save directories from node Node1
Executing save directories from node Node2

... saved
Sat Apr 28 08:50:14 2007 - Recompiling invalid objects ... recompiled
Sat Apr 28 08:50:18 2007 - Exporting sysman schema for DB102 ... exported
Sat Apr 28 08:53:21 2007 - DB Control was saved successfully.
Sat Apr 28 08:51:36 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 08:57:26 2007 - Dump directory was dropped successfully.
```

> **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. But you will lose all user data accumulated between the upgrade and restore operations. Saving your Database Control files and data enables you to downgrade both your database and Database Control. You will lose all Database Control data accumulated between the upgrade and downgrade, but you will retain all user data.

Complete Materialized View Refreshes

For a list of materialized view refreshes that need to be completed, issue the following statement:

```
SELECT DISTINCT(TRUNC(last_refresh))
FROM dba_snapshot_refresh_times;
```

Ensure No Files Need Media Recovery

For a list of files that require media recovery, issue the following statement:

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

Ensure No Files Are in Backup Mode

For a list of files in backup mode, issue the following statement:

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

Resolve Outstanding Distributed Transactions

Resolve outstanding distributed transactions prior to the upgrade.

```
SELECT * FROM dba_2pc_pending;
```

If this query 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;
```

Sync Standby Database with the Primary Database

To check if a standby database exists, issue the following query:

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

If this query returns a row, then sync the standby database with the Primary database.

1. Make sure all the logs are transported to the standby server after a final log switch in the primary.

2. Start the recovery of the standby database with the NODELAY option.

Purging the Database Recycle Bin

Use the PURGE statement to remove items and their associated objects from the recycle bin and release their storage space:

```
PURGE DBA_RECYCLEBIN
```

It is REQUIRED that the recycle bin be empty during the upgrade to avoid possible ORA-00600 errors as well as to minimize the upgrade time.

Run the Oracle Net Configuration Assistant

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 Real Application Clusters (Oracle RAC) database, then you have two options:

- Upgrade the Oracle RAC database with DBUA, which automatically migrates the listener from your old Oracle home to the Oracle home of the new Oracle Database 11*g* release.

- 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.

  > **Note:** This is your only option if you want to upgrade your Oracle RAC database manually.

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

Upgrade a Database Using Database Upgrade Assistant

The following sections guide you through the process of upgrading a database using Database Upgrade Assistant (DBUA). Please note the following:

- DBUA can be used to upgrade from prior Oracle Database 11*g* patch releases as well as from prior major Oracle Database releases on both Oracle Real Application Clusters 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 "Run the Oracle Net Configuration Assistant" on page 3-18.

  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 prior to 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 will prompt you and start the default listener in either of these cases.

- 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 an error similar to the following:

  ```
  PRKH-1014 Current user user is not the same as owner owner of oracle home
  ```

- 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 you are prompted to provide the name of the correct initialization parameter file or to start the instance. If the instance is already up and running, then DBUA connects to it.

- If you terminate the upgrade, but do not restore the database, then you should not 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.

- If you restore your database manually (not using DBUA), then remove the `Welcome_`*SID*`.txt` file, 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.

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:

- Using the DBUA Graphical User Interface
- Using DBUA in Silent Mode

Using 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 is started automatically. However, if you did not specify that you are upgrading an existing database, then you can start DBUA independently after installation is complete.

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. If Database Vault is enabled, then DBUA returns an error asking you to disable Database Vault prior to upgrade. See "Upgrading Databases That Use Oracle Database Vault" on page 3-6.

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

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 the user 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.

While the upgrade is running, 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.

Complete the following steps to upgrade a database using the DBUA graphical user interface:

1. Do one of the following to start DBUA:

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

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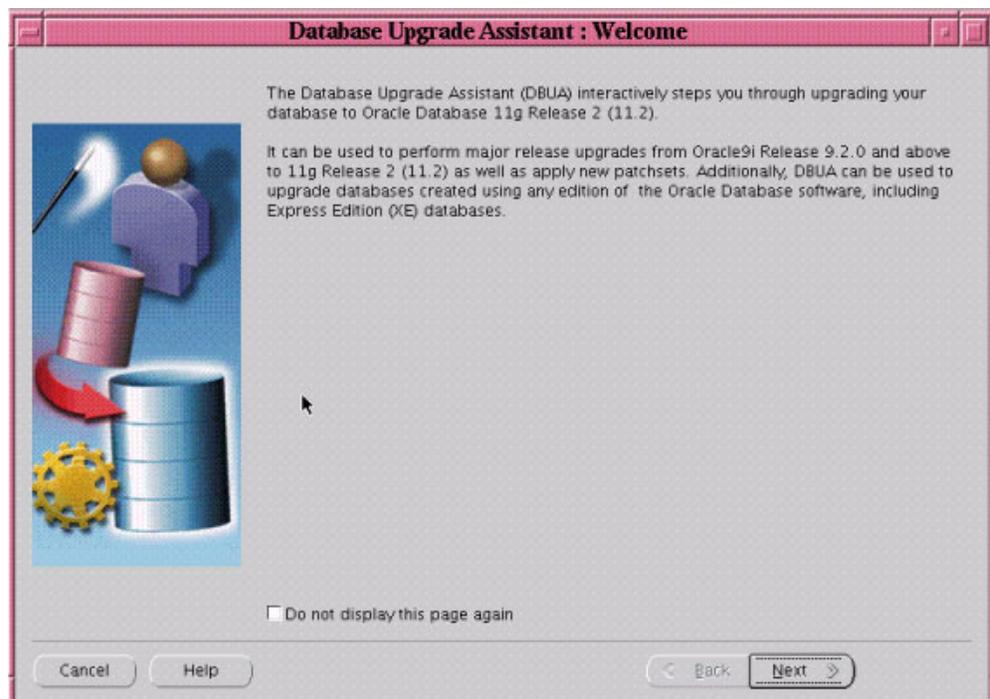
2. If you need help at any screen or want to consult more documentation about DBUA, then click **Help** to open the online help.

   Click **Next**.

   The Select Database page appears, listing the databases available for upgrade.
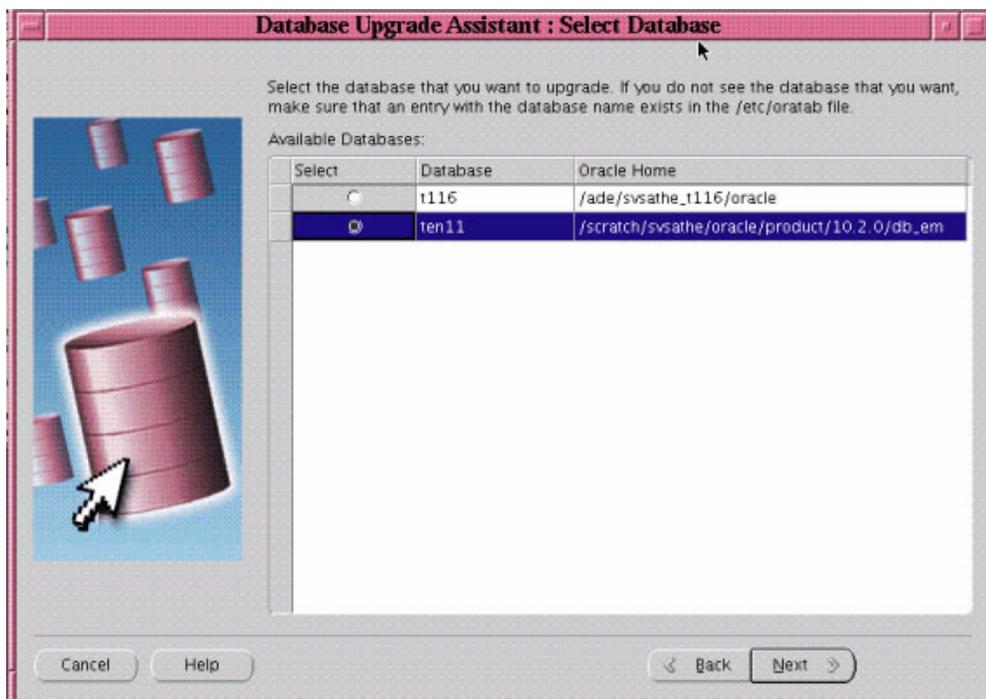
3. Select the database you want to upgrade to Oracle Database 11*g* Release 2 (11.2).

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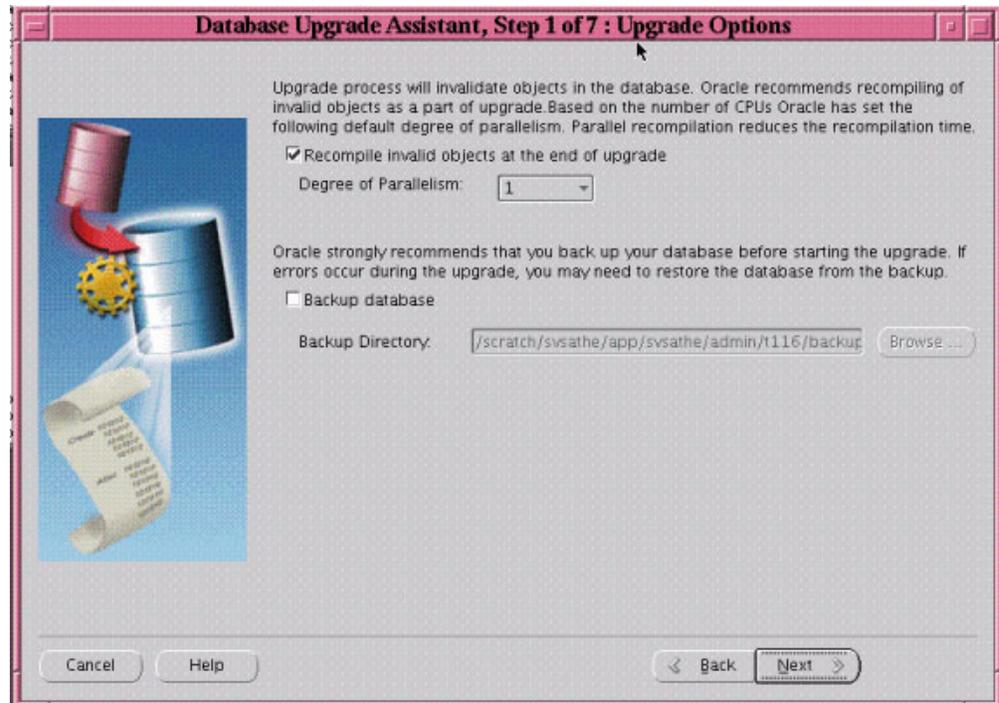
4. The Upgrade Options screens allows you to set the following options:

    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.

Oracle strongly recommends that you back up your database before starting the upgrade. If errors occur during the upgrade, 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 already backed up your database using your own backup procedures.

> **Note:** DBUA does not back up 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.
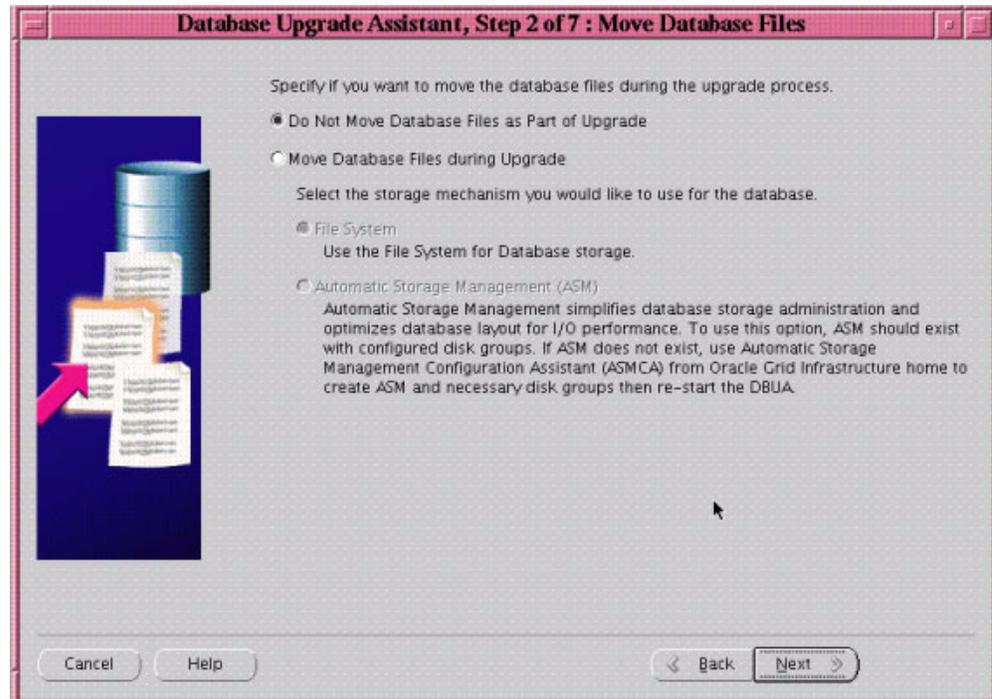
5. Select one of the following options:

   ■ **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 select one of the following:

   ■ **File System**

   Your database files are moved to the host file system.

   ■ **Automatic Storage Management (ASM)**

   Your database files are moved to ASM storage, which must already exist on your system. If you do not have an ASM instance, you can create one using DBCA and then restart DBUA.

   See Also:

   ■ Your platform-specific Oracle Grid Infrastructure Installation Guide for information about installing and configuring ASM

   ■ "Managing ASM Instances With ASM Configuration Assistant" in the *Oracle Database Storage Administrator's Guide*

   Click **Next**.

   The Recovery and Diagnostic Locations screen displays, where you can designate a Fast Recovery Area for your database.
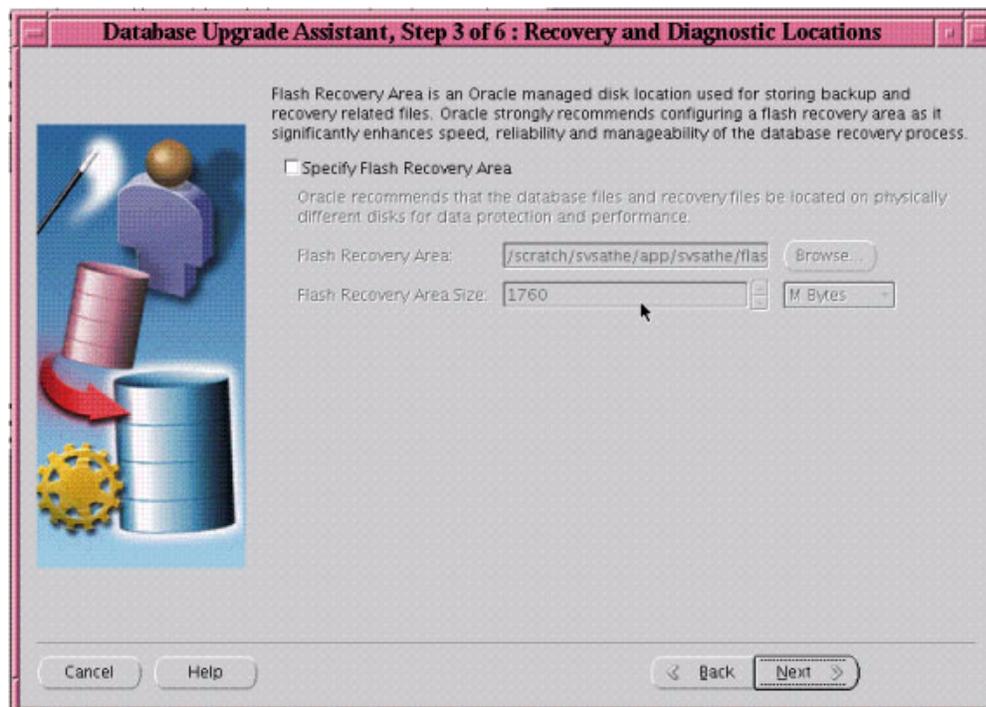
Fast Recovery Area is an Oracle managed disk location used for storing backup and recovery related files. Oracle strongly recommends configuring a Fast Recovery Area, because it significantly enhances speed, reliability, and manageability of the database recovery process. The location of the Fast Recovery 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 Fast Recovery Area. If a Fast Recovery Area is already configured, then the current settings are retained but the screen will display to allow you to override these values.

6. Do one of the following:

   - Accept the default Fast Recovery Area location

   - Enter the full path to a different location in the **Flash Recovery Area** field

   - Click **Browse** and select a different Fast Recovery Area location

   Do one of the following:

   - Accept the default size for the Fast Recovery Area

   - Enter a different value in the **Flash Recovery Area Size** field

   - Click the up or down arrows to adjust the Fast Recovery Area size

     See Also: *Oracle Database 2 Day DBA* for more information about configuring the Fast Recovery Area

   Click **Next**.

   If no other database is already being monitored with Enterprise Manager, then the Management Options screen displays.
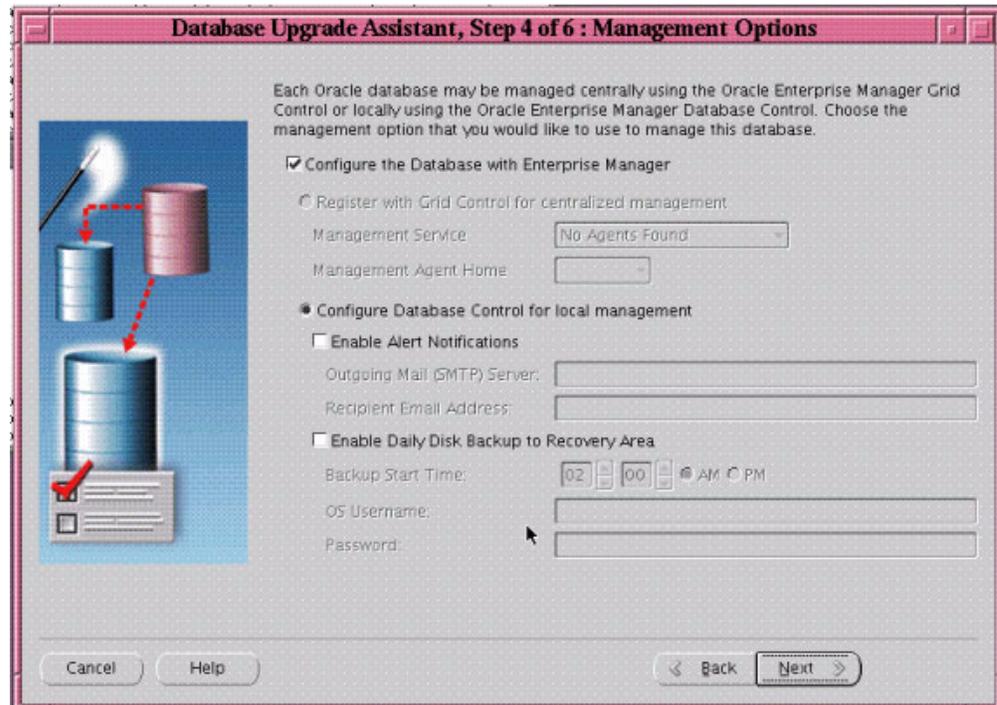
At the Management Options screen, you have the option of setting up your database so it can be managed with Enterprise Manager. Enterprise Manager provides Web-based management tools for managing individual database instances, as well as central management tools for managing your entire Oracle environment, 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 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.

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 8 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.
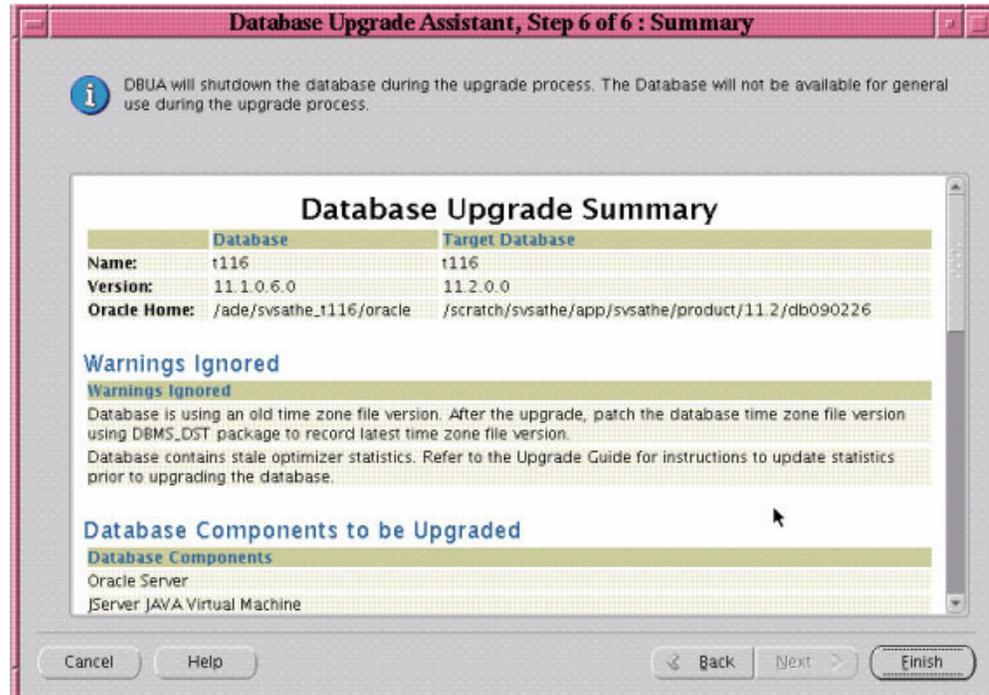
9. The Summary screen shows the following information about the upgrade before it starts:

■ 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 4-12 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 8 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. DBUA removes the entry of the upgraded database from the old listener.ora file and reloads the listener of the old database.

Using DBUA in Silent Mode

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.

For example, the following command upgrades a database named ORCL in silent mode:

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

DBUA Command Line Options

DBUA supports several command line options. 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]
```

Table 3–3 describes the various options and their parameters that are supported by DBUA.

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	This option 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.
–generateMapFile	If you specify this option, then DBUA only generates a database map file in the log location and then exits. This option applies only if you are moving data files or upgrading an Oracle XE database.
–useASM	If the database to upgrade has an ASM instance, then this option tells DBUA to use it for the upgrade. This option applies only if you are moving data files or upgrading an Oracle XE database.
–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.

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

Option	Description
–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	This option 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	This option 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	This option 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.
–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.

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

Option	Description
–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 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.
[–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

Note: If the default Oracle Database 11*g* Release 2 (11.2) security settings are in place, then passwords must be at least 8 characters, and passwords such as `welcome` and `oracle` are not allowed. See *Oracle Database Security Guide* for more information.

Upgrade a Database Manually

The following sections guide you through the process of performing a manual upgrade. They assume that you have already run the Pre-Upgrade Information Tool as described in "Run the Pre-Upgrade Information Tool" on page 3-8.

- Back Up the Database

- Prepare the New Oracle Home
- Upgrade the Database
- Troubleshoot the Upgrade
- Cancel the Upgrade

Back Up the Database

After running the Pre-Upgrade Information Tool and cleanly shutting down the instance, you should perform a full backup of the database. Complete the following steps:

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 TO 'save_controlfile_location';
}
```

Caution: If you encounter problems with the upgrade and wish to abandon the upgrade completely, then you must restore the database from this backup. Therefore, make sure you back up your database now as a precaution.

See Also: *Oracle Database Backup and Recovery User's Guide* for more information about backing up a database

---

Prepare the New Oracle Home

After analyzing the database to be upgraded, complete the following steps to prepare the new Oracle home:

1. Copy configuration files from the Oracle home of the database being upgraded to the 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 Automatic Storage Management (ASM) instance, then backup the parameter file using the following command:

```
CREATE pfile FROM spfile;
```

If you must downgrade the database and your `SPFILE` lived in ASM, then the parameter file must be restored prior to 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 `orapwsid`, located in the *ORACLE_HOME*/dbs directory. On Windows operating systems, the default password file is `pwdsid.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 `initdb_name.ora` file resides within the old environment's Oracle home, then move or copy the `initdb_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. 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" 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.

3. 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`.

Upgrade the Database

After preparing the new Oracle home, you are ready to proceed with the manual upgrade. Complete the following steps to 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 8 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 make 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 in which this cluster database 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.

4. Log in to the system as the owner of the Oracle Database 11*g* Release 2 (11.2) Oracle home directory.

5. At a system prompt, change to the *ORACLE_HOME*/rdbms/admin directory.

6. Start SQL*Plus.

7. Connect to the database instance as a user with `SYSDBA` privileges.

8. 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 will execute without error until after the `catupgrd.sql` script is run. Prior to running `catupgrd.sql`, queries on any other view or the use of PL/SQL will return 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.

9. Do this step only if you are upgrading from Oracle9*i* Release 2 (9.2).

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`

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

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

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

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

11. Run the catupgrd.sql script:

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

Note: If you did not run the Pre-Upgrade Information Tool, the catupgrd.sql script will terminate 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 "Run the Pre-Upgrade Information Tool" on page 3-8.

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

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

```
SQL> STARTUP
```

This restart, following the database shutdown performed as part of the `catupgrd.sql` script, flushes all caches, clears 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.

Also, if you encountered a message listing obsolete initialization parameters when you started the database in Step 8, 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.

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

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

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

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

```
Oracle Database 11.2 Post-Upgrade Status Tool          03-18-2007 22:48:55
.
Component                                 Status          Version   HH:MM:SS
```

```
.
Oracle Server
.                                        VALID      11.2.0.0.1  00:17:31
JServer JAVA Virtual Machine
.                                        VALID      11.1.0.6.0  00:02:32
Oracle Workspace Manager
.                                        VALID      11.1.0.6.0  00:01:02
Messaging Gateway
.                                        VALID      11.1.0.6.0  00:00:46
OLAP Analytic Workspace
.                                        VALID      11.1.0.6.0  00:03:20
OLAP Catalog
.                                        VALID      11.1.0.6.0  00:05:56
Oracle OLAP API
.                                        VALID      11.1.0.6.0  00:00:48
Oracle Label Security
.                                        VALID      11.1.0.6.0  00:00:37
Oracle Enterprise Manager
.                                        VALID      11.1.0.6.0  00:12:02
Oracle XDK
.                                        VALID      11.1.0.6.0  00:00:42
Oracle Text
.                                        VALID      11.1.0.6.0  00:01:02
Oracle XML Database
.                                        VALID      11.1.0.6.0  00:04:24
Oracle Database Java Packages
.                                        VALID      11.1.0.6.0  00:00:27
Oracle Multimedia
.                                        VALID      11.1.0.6.0  00:05:44
Spatial
ERROR at line 1:
ORA-00604: error occurred at recursive SQL level 1
ORA-04031: unable to allocate 4096 bytes of shared memory ("java
pool","sun/tools/java/ConstantsSYS","joxlod exec hp",":SGAClass")
ORA-06512: at "SYS.DBMS_JAVA", line 704
ORA-06512: at line 1
.                                        VALID      11.1.0.6.0  00:08:24
Oracle Expression Filter
.                                        VALID      11.1.0.6.0  00:00:38
Oracle Rules Manager
.                                        VALID      11.1.0.6.0  00:00:21
Oracle Application Express
.                                        VALID      2.2.1.01.01 00:23:25
Gathering Statistics
.      00:05:12
Total Upgrade Time: 01:29:03
```

If the Post-Upgrade Status Tool returns errors or shows components that are not VALID or not the most recent release, then see "Troubleshoot the Upgrade" on page 3-43.

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

```
SQL> @rdbms/admin/catuppst.sql
```

This script can be run concurrently with `utlrp.sql` (see step 15), if you run it in a new SQL*Plus session.

15. 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 as a result of the upgrade process. The `utluiobj.sql` script will only display objects that are now `INVALID` but which were `VALID` prior to the upgrade.

---

16. Exit SQL*Plus.

Your database is now upgraded to the new Oracle Database 11*g* release. Complete the procedures described in Chapter 4, "After Upgrading to the New Release".

WARNING: **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.**

Troubleshoot the Upgrade

This section explains what to do if something goes wrong with your upgrade. It contains the following topics:

- Resource Limits
- Manual Workaround for ORA-01408
- Running the DBMS_DST Package After Upgrade Can Result in ORA-01822
- Component Status
- Rerunning the Upgrade

Resource Limits

If you run out of resources during the upgrade, 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.

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

- `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.

- Fast Recovery Area

  If you are using a Fast Recovery Area and it fills up during the upgrade, you will see the following error in the alert log, followed by some 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 Fast Recovery Area prior to starting the upgrade.

Manual Workaround for ORA-01408

The `ORA-01408` error on the index is a known problem with Oracle Application databases, because the same index exists with a different name in these databases.

```
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" on page 3-45.

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: timezone 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: timezone 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 "TIMESTAMP WITH TIME ZONE Data Type" on page 14 and rerun the upgrade as described in "Rerunning the Upgrade" on page 3-45.

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.

Rerunning the Upgrade

Follow these steps to rerun the upgrade:

1. Shut down the database:

    ```
    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
   ```

Cancel the Upgrade

If you completed the steps in "Back Up the Database" on page 3-35 to back up your database, then the easiest way to cancel the upgrade is to restore that backup. Complete the following steps:

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;
   }
   ```

Upgrade an ASM Instance

The recommended practice is to upgrade an Oracle ASM instance with Oracle grid infrastructure Oracle Universal Installer (OUI). OUI automatically defaults to upgrade mode when it detects an Oracle ASM instance at a previous release level. See "Upgrading an Oracle ASM Instance With Oracle Universal Installer" in *Oracle Database Storage Administrator's Guide*.

Oracle ASM Configuration Assistant enables you to upgrade an existing ASM instance to the current software level and upgrade an older ASM instance to the latest Oracle grid infrastructure home. See "Upgrading an Oracle ASM Instance with Oracle ASM Configuration Assistant" in *Oracle Database Storage Administrator's Guide*.

Upgrading a single-instance ASM requires that Oracle Cluster Synchronization Services (CSS) first be configured to run from the new Oracle Database 11*g* Release 2

(11.2) home. See "Reconfigure Oracle Cluster Synchronization Services" on page 3-47 for more information.

You can also perform a rolling upgrade to clustered ASM instances in environments running Oracle Database 11*g* or later. See your *Oracle Grid Infrastructure Installation Guide* for step-by-step instructions about performing a rolling upgrade of ASM.

Upgrade an ASM Instance Manually

The following sections guide you through the process of performing a manual upgrade. The discussions assume that you have already installed the Oracle Database 11*g* Release 2 (11.2) software and run the Pre-Upgrade Information Tool as described in "Run the Pre-Upgrade Information Tool" on page 3-8.

- Reconfigure Oracle Cluster Synchronization Services
- Prepare the New ASM Oracle Home
- Upgrade ASM
- Troubleshoot the ASM Upgrade
- Cancel the ASM Upgrade

Reconfigure Oracle Cluster Synchronization Services

Upgrading a single-instance ASM requires that Oracle Cluster Synchronization Services (CSS) first be configured to run from the new Oracle Database 11*g* Release 2 (11.2) home. To do this you need to run the following command before upgrading ASM:

```
localconfig reset [destination_Oracle_home]
```

where *destination_Oracle_home* is required if you run the command from the Oracle home where the CSS service is currently configured.

Prepare the New ASM Oracle Home

Complete the following steps to prepare the new Oracle home:

1. Copy configuration files from the Oracle home of the ASM instance being upgraded to the 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 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.

c.   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 `orapwsid`, located in the *ORACLE_HOME*/dbs directory. On Windows operating systems, the default password file is `pwdsid.ora`, located in the *ORACLE_HOME*\database directory. In both cases, *sid* is your Oracle instance ID.

> **Note:**   If you are upgrading a clustered ASM, then perform this step on all nodes that have ASM instances configured.

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

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

d.   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.

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

> **Note:**   If you are upgrading a clustered ASM, then perform this step on all nodes that have ASM instances configured.

Upgrade ASM

After preparing the new Oracle home, you are ready to proceed with the manual upgrade. Complete the following steps to upgrade ASM:

1. Shut down the instance:

```
SQL> SHUTDOWN IMMEDIATE
```

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

a. Stop the `OracleASMServiceSID` Oracle service of the ASM instance you are upgrading, where `SID` is the instance name. For example, if your `SID` is `+ASM`, then enter the following at a command prompt:

```
C:\> NET STOP OracleASMService+ASM
```

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

If your `SID` is `+ASM`, then enter the following command:

```
C:\> ORADIM -DELETE -ASMSID +ASM
```

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 -ASMSID 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 ASM instance you are upgrading.
PASSWORD	The password for the new Oracle Database 11*g* Release 2 (11.2) ASM 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 8 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 home for the new Oracle Database 11*g* Release 2 (11.2) ASM instance. 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 `+ASM`, your `password` (INTPWD) is `TWxy5791`, the maximum number of users (`MAXUSERS`) is 10, and the *ORACLE_HOME* directory is C:\ORACLE\PRODUCT\11.1.0\DB, then enter the following command:

```
C:\> ORADIM -NEW -ASMSID +ASM -SYSPWD TWxy5791 -MAXUSERS 10
-STARTMODE AUTO -PFILE C:\ORACLE\PRODUCT\11.1.0\DB\DATABASE\INIT+ASM.ORA
```

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

a. Your `ORACLE_SID` is set correctly

b. The `oratab` file points to the Oracle home for the new Oracle Database 11*g* Release 2 (11.2) ASM instance

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

       – `ORACLE_HOME`

       – `PATH`

> **Note:** If you are upgrading a clustered ASM, then perform these checks on all nodes that have ASM instances configured.

> **See Also:** Your operating system-specific Oracle Database installation documents for information about setting other important environment variables on your operating system.

4. Log in to the system as the owner of the Oracle home for the new Oracle Database 11*g* Release 2 (11.2) ASM instance.

5. Start SQL*Plus.

6. Connect to the ASM instance as a user with `SYSASM` privileges using operating system (OS) authentication.

7. Start up the instance by issuing the following command:

```
SQL> STARTUP
```

8. If you encounter a message listing obsolete initialization parameters when you start ASM, 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.

9. Grant the `SYSMAN` privilege to `SYS`:

```
GRANT sysasm TO sys;
```

This command allows the `SYS` user to be authenticated using the password file.

10. Exit SQL*Plus.

Your ASM is now upgraded to the Oracle Database 11*g* Release 2 (11.2). Complete the ASM-specific procedures described in Chapter 4, "After Upgrading to the New Release".

Troubleshoot the ASM Upgrade

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

If you run out of resources during the ASM upgrade, then increase the resource allocation. After increasing the resource allocation, you should perform a `SHUTDOWN ABORT` and restart the instance before performing a manual ASM upgrade.

A new Oracle Database release 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.

Cancel the ASM Upgrade

To cancel an ASM upgrade, start the ASM instance from the old Oracle home.

4

After Upgrading to the New Release

This chapter guides you through the procedures to perform after you have completed an upgrade of your database. This chapter contains the following topics:

- Required Tasks After Database Upgrades
- Recommended Tasks After Database Upgrades
- Tasks to Complete Only After Manual Database Upgrades
- Required Tasks After ASM Upgrades
- Recommended Tasks After ASM Upgrades
- Optional Tasks After ASM Upgrades
- Tasks to Complete Only After Manual ASM Upgrades

Required Tasks After Database Upgrades

Complete the following tasks after you have upgraded your database, regardless of whether you performed the upgrade manually or by using Database Upgrade Assistant (DBUA):

- Update Environment Variables (Linux and UNIX Systems Only)
- Upgrade the Recovery Catalog
- Upgrade the Time Zone File Version
- Upgrade Statistics Tables Created by the DBMS_STATS Package
- Upgrade Externally Authenticated SSL Users
- Install Oracle Text Supplied Knowledge Bases
- Update Your Oracle Application Express Configuration
- Configure Fine-Grained Access to External Network Services
- Enable Oracle Database Vault and Revoke the DV_PATCH_ADMIN Role

Update Environment Variables (Linux and UNIX Systems Only)

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 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 cluster database, then perform these checks on all nodes in which this cluster database 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.

Upgrade the Recovery Catalog

For complete information about upgrading the recovery catalog and the UPGRADE CATALOG command, see the *Oracle Database Backup and Recovery User's Guide*.

Upgrade the Time Zone File Version

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:**
> - "TIMESTAMP WITH TIME ZONE Data Type" on page 3-14
> - *Oracle Database Globalization Support Guide* and follow the instructions in "Steps to Upgrade Time Zone File and Timestamp with Time Zone Data"

Upgrade Statistics Tables Created by the DBMS_STATS Package

If you created statistics tables using the `DBMS_STATS.CREATE_STAT_TABLE` procedure, then upgrade these tables by executing 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. Execute this procedure for each statistics table.

Upgrade Externally Authenticated SSL Users

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 following command to upgrade those users:

```
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:** *Oracle Database Enterprise User Security Administrator's Guide* for more information on the `extusrupgrade` script

Install Oracle Text Supplied Knowledge Bases

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 post-installation tasks section of your platform-specific *Oracle Database Installation Guide* for companion products

Update Your Oracle Application Express Configuration

If your database originally included Application Express Version 3.2, 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 a prior version of Application Express (HTML DB), then version 3.2 is automatically installed during the upgrade. You must complete a series of post-installation steps to configure Application Express Version 3.2 for use with the new Oracle Database 11*g* release. These steps are described in the section on post-installation tasks in the *Oracle Application Express Installation Guide*.

If your database was an Oracle Express Edition (XE) database, then it contained a prior version of Application Express tailored for the XE environment. See the OTN document describing the differences between Oracle XE and Oracle Application Express 3.2 at:

```
http://www.oracle.com/technology/products/database/application_
express/html/3.2_and_xe.html
```

The database administration features available with the XE version of Application Express are not available in version 3.2, but Oracle Enterprise Manager DB Control can, optionally, be installed to provide a graphical interface for database administration.

Configure Fine-Grained Access to External Network Services

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 one of these packages, you must install Oracle XML DB if it is not already installed. You must also configure network access control lists (ACLs) in the database before these packages can work as they did in prior 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 already have it. 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

Enable Oracle Database Vault and Revoke the DV_PATCH_ADMIN Role

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:**
  > - "Upgrading Databases That Use Oracle Database Vault" on page 3-6
  > - 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 After Database Upgrades

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 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 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
- Upgrade Automatic Storage Management (ASM)
- 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

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 11*g* security settings are in place, then passwords must be at least 8 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*

Upgrade Automatic Storage Management (ASM)

When using DBUA, you can choose to upgrade an Oracle Database instance, an ASM instance, or both. However, when performing a manual upgrade, you must upgrade ASM separately from the Oracle Database upgrade.

> **See Also:**
>
> - "Upgrade an ASM Instance" on page 3-46
> - "Upgrade an ASM Instance Manually" on page 3-47
> - "Required Tasks After ASM Upgrades" on page 4-14
> - "Recommended Tasks After ASM Upgrades" on page 4-16
> - "Tasks to Complete Only After Manual ASM Upgrades" on page 4-18

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 Your Applications" 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 execute 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 Real Application Clusters (Oracle RAC) configurations, repeat these steps on all instances.

Configure Oracle Data Guard Broker

The Data Guard broker property `LocalListenerAddress` is being deprecated. Because the manner in which broker communication and redo transport setting are being changed, the value of the `LocalListenerAddress` is not maintained in Oracle Database 11*g* Release 1 (11.1).

The broker property `InitialConnectIdentifier` is being changed to `DGConnectIdentifier`. The value of `DGConnectIdentifier` is used for all Data Guard network traffic, all of the time. While upgrading an Oracle Database 10*g* configuration to Oracle Database 11*g* Release 1 (11.1), the `InitialConnectIdentifier` value is retained as the new

DGConnectIdentifier value for that database. During the upgrade, it is up to the database administrator to ensure that the InitialConnectIdentifier reaches all instances if this is an Oracle Real Application Clusters (Oracle RAC) database.

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, the ALTER TABLE statement can be used 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 make sure that the applications operate in the same manner as they did before the upgrade.

> **See Also:** Chapter 5, "Upgrading Your Applications" 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 versions, enabling asynchronous CDC to leverage any existing Oracle9*i* Release 2 (9.2) system as a source.

See the *Oracle Database Data Warehousing Guide* for complete information about how to upgrade 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. The discussion describes the supported configurations for the Distributed HotLog mode of Change Data Capture as well as the restrictions.

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 XDB configuration file is automatically upgraded so that the XDB 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 XDB configuration file by default during an upgrade. If you wish to have 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 XDB configuration file, you must configure the database and the listener to enable Oracle XML DB to use HTTPS. Additionally, if the following steps were not taken before the upgrade, then you must perform them after the upgrade:

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 already 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.

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 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`. It can be used to disallow unauthenticated access to your Oracle XML DB Repository data through HTTP even if you unlock the `ANONYMOUS` user account. It is not added to the 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 wish 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 DB Control during the DBUA upgrade, then you can install it, along with any other components you would like to have in the database.

Tasks to Complete Only After Manual Database Upgrades

If you are performing a manual upgrade rather than using DBUA, then you must perform the following tasks after your database is upgraded:

- Change Passwords for Oracle-Supplied Accounts
- Migrate Your Initialization Parameter File to a Server Parameter File
- Upgrade Oracle Text
- Upgrade the Oracle Cluster Registry (OCR) Configuration
- Adjust the Initialization Parameter File for the New Release
- Configure Enterprise Manager
- Set CLUSTER_DATABASE Initialization Parameter

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 8 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;
```

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 Real Application Clusters, 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 Real Application Cluster 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 the above files by:

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 Support Note 300172.1 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

Upgrade the Oracle Cluster Registry (OCR) Configuration

If you are using Oracle Clusterware, then you must upgrade the Oracle Cluster Registry (OCR) keys for the database.

Use one of the following options to upgrade the OCR configuration to the new Oracle Database 11*g* release:

- Use srvconfig from the Oracle Database 11*g* Release 2 (11.2) Oracle home. For example:

  ```
  % srvconfig -upgrade -dbname db_name -orahome pre-11g_Oracle_home
  ```

- Run srvctl. For example:

  ```
  pre-11g_Oracle_home/bin/srvctl remove database -d db_name
  11g_Oracle_home/bin/srvctl add database -d db_name -o 11g_Oracle_home
  11g_Oracle_home/bin/srvctl add instance -d db_name -i instance -n node
  ```

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 should adjust the parameter file to account for these changes and to take advantage of new initialization parameters that might be beneficial to your system.

> **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" for lists of obsolete and deprecated initialization parameters in the new Oracle Database 11*g* release

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.

Setting the COMPATIBLE Initialization Parameter

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.

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 need to 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

In "Prepare the New Oracle Home" on page 3-35, 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 ASM Upgrades

The following tasks are required after an ASM Upgrade:

- Set Environment Variables
- Single-Instance ASM Upgrade
- Cluster ASM Upgrade

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 ASM, then perform these checks on all nodes in which this clustered 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 ASM Upgrade

If ASM version 10.2 is installed in Oracle Home 1 (OH1) and the operating system user is orauser, execute the following steps:

1. As orauser, upgrade ASM to release 11.2 using OUI and DBUA. The new ASM release 11.2 should be running in a new Oracle Home 2 (OH2). ASM should still be running as orauser.

2. As `orauser`, bring down the 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 ASM instance path and `connect-string` role.

9. Make sure the disks are owned by `asmuser` and by OSDBA for ASM. They should also have the `0660` permission set.

10. As `asmuser`, start the listener.

11. As `asmuser`, start ASM (connect as `SYSASM`).

12. Run the command, `GRANT sysasm TO sys`.

Cluster ASM Upgrade

For cluster ASM upgrade, execute the following steps:

1. As `orauser`, upgrade ASM to release 11.2 using OUI and DBUA. The new ASM release 11.2 should be running in a new Oracle Home 2 (`OH2`). ASM should still be running as `orauser`.

2. Bring down the ASM and Listener resources from CRS home.

3. As a new user (`asmuser`), install 11.2 in a third Oracle Home (`OH3`). This should 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
```

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 ASM instance path and `connect-string` role.

8. Make sure the disks are owned by `asmuser` and by OSDBA for ASM. They should also have the `0660` permission set.

9. Start ASM and Listener resources from the new 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 ASM instances, then you also have the option of performing a rolling ASM upgrade. A rolling upgrade enables you to independently upgrade or patch ASM nodes without affecting database availability, thus providing greater uptime.

> **See Also:** *Oracle Database Storage Administrator's Guide* for more information on rolling ASM upgrades

Recommended Tasks After ASM Upgrades

Performing the following tasks is recommended, but not required, after you have upgraded ASM:

- Reset ASM Passwords to Enforce Case-Sensitivity
- Advance the ASM and Oracle Database Disk Group Compatibility
- Set Up 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 4-6
- "Develop New Administrative Procedures as Needed" on page 4-6

Reset 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 ASM instances, there are no additional tasks or management requirements. For upgraded 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 8 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 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 ASM disk group compatibility settings across software versions.

Advancing compatibility enables new features only available in the new version. However, it makes the disk group incompatible with older versions of the software. Note that advancing on-disk compatibility is an irreversible operation.

You use the `compatible.rdbms` and `compatible.asm` attributes to specify the minimum software version required by the database instance and the ASM instance, respectively, to access the disk group. For example, the following `ALTER DISKGROUP` statement advances the 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 ASM software of version 11.1 or higher, while any database client of version 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 ASM Preferred Read Failure Groups

ASM administrators can specify some disks to be preferred over others for read I/O operations. When ASM preferred read failure groups are defined, 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 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

Optional Tasks After ASM Upgrades

If you separate the operating system ownership of Oracle homes belonging to ASM and to one or more databases, then you must migrate the operating system user of an upgraded ASM or database home by executing the steps in the following section:

- Database Upgrade After ASM Upgrade

Database Upgrade After ASM Upgrade

If you are migrating to an environment with separate operating system users, you must upgrade the database after upgrading ASM. The database user must be a member of OSDBA for ASM.

There are three scenarios to consider:

- Keeping orauser as the Operating System User

- Changing the Operating System User For a Single-Instance Database

- Changing the Operating System User for an Oracle RAC Database

> **See Also:** *Oracle Database Storage Administrator's Guide* for information on making an ASM disk group compatible with Oracle Database 10*g* and Oracle Database 11*g*

Keeping orauser as the Operating System User

If you are maintaining the operating system user as `orauser`, run DBUA to upgrade the database from 10.2 to 11.1 in a new Oracle Home (`OH4`) which is separate from the ASM Oracle Home (`OH3`).

Changing the Operating System User For a Single-Instance Database

Consider your 10.2 database is installed in Oracle Home 4 (`OH4`) and currently running `orauser` as the operating system user:

1. As `orauser`, run DBUA to upgrade the database from 10.2 to 11.1 in a new Oracle Home (`OH5`).

2. As `orauser`, bring down the database instance.

3. As `newuser`, install 11.1 in a separate Oracle Home (`OH6`).

4. Update `/etc/oratab` to that `OH6` is the Oracle home with the database entry.

5. Copy `sqlnet.ora`. Update listener.ora so that all occurrences of `OH5` are now `OH6`.

6. Modify the `SPFILE` in the ASM disk group to use `OH6` instead of `OH5`.

7. Copy the password file from `OH5` to `OH6`.

8. Run EMCP to change ASM instance path and `connect-string` role.

9. Start the database instance.

This is useful if you have two databases using ASM. If needed, the user can change the operating system user of the database in order to allow separate databases to run as separate operating system users.

Changing the Operating System User for an Oracle RAC Database

Consider your 10.2 database is installed in Oracle Home 4 (`OH4`) and currently running `orauser` as the operating system user:

1. As `orauser`, run DBUA to upgrade the database from 10.2 to 11.1 in a new Oracle Home (`OH5`).

2. As `orauser`, bring down the database instance.

3. Run `srvctl remove <`*db-name*`>`.

4. As `newuser`, install 11.1 in a separate Oracle Home (`OH6`).

5. Update `/etc/oratab` to that `OH6` is the Oracle home with the database entry.

6. Copy `sqlnet.ora`. Update listener.ora so that all occurrences of `OH5` are now `OH6`.

7. Modify the `SPFILE` in the ASM disk group to use `OH6` instead of `OH5`.

8. Copy the password file from `OH5` to `OH6`.

9. Run EMCP to change ASM instance path and `connect-string` role.

10. Run `srvctl add <`*db-name*`>`.

11. Start the database instance.

Tasks to Complete Only After Manual ASM Upgrades

If you are performing a manual upgrade, then you must perform the following tasks after ASM is upgraded:

- Upgrade the Oracle Cluster Registry (OCR) Configuration for ASM
- Adjust the ASM Initialization Parameter File for the New Release
- Install and Configure Enterprise Manager Database Control for ASM

Upgrade the Oracle Cluster Registry (OCR) Configuration for ASM

If you are using Oracle Clusterware, then you must upgrade the Oracle Cluster Registry (OCR) keys for ASM by running the following command:

```
srvctl modify asm -n node [-p spfile] -o asm_home -i instance
```

The **-p** option is necessary only if you are using an SPFILE and it was moved.

Adjust the ASM 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 should adjust the parameter file to account for these changes and to take advantage of new initialization parameters that might be beneficial to your system.

> **See Also:**
>
> - The "What's New in Oracle Database Reference" section in *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" for lists of obsolete and deprecated initialization parameters in the new Oracle Database 11*g* release

Install and Configure Enterprise Manager Database Control for ASM

If you want to use Enterprise Manager Database Control with ASM, then you must install and configure it.

> **See Also:** *Oracle Enterprise Manager Advanced Configuration* for information on configuring Enterprise Manager Database Control

5

Upgrading Your Applications

This chapter describes upgrading your current applications and covers the following topics:

- Overview of Upgrading Applications
- Upgrading Precompiler and OCI Applications
- Upgrading SQL*Plus Scripts and PL/SQL
- Upgrading Oracle Forms or Oracle Developer Applications

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

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" 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

Net8 and Oracle Net Services work with various Oracle Database releases. Thus, Oracle8*i*, Oracle9*i*, Oracle Database 10*g*, and Oracle Database 11*g* can communicate by using Net8 and Oracle Net Services.

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.

> **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.

Types of Software Upgrades

Two types of upgrades are possible for Oracle Database client and server software.

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.

> **Note:** Starting with Oracle9*i* Release 2 (9.2), maintenance releases of Oracle Database software are denoted by a change to the second digit of a release number. In previous releases, the third digit indicated a particular maintenance release.

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 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. 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 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 drop-in compatible with the current release. That is, the Oracle client-side dynamic library is upward compatible with the previous version of the library. The 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 and its environment 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 and then relink the application using the new Oracle Database 11*g* release. Application code must be changed if any APIs are deprecated or changed.

- **Option 3:** Change the application code to use new Oracle Database 11*g* features. Then, 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, however, 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. However, you are required to precompile or compile, and relink your application only once, regardless of the number of clients you have. The advantages, however, can be quite large.

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. 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 code for them 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.

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 a Database

This chapter guides you through the process of downgrading a database to a previous Oracle Database release. In all discussions of downgrading, it is important to understand that you can only downgrade to the release from which you upgraded. For example, if you upgraded from Oracle Database 10*g* Release 1 (10.1.0.5) to Oracle Database 11*g* Release 2 (11.2), then you could not subsequently downgrade to Oracle Database 10*g* Release 2 (10.2); you could only downgrade to Oracle Database 10*g* Release 1 (10.1.0.5).

This chapter covers the following topics:

- Supported Releases for Downgrading
- Check for Incompatibilities
- Perform a Full Backup
- Downgrade the Database
- Perform Post-Downgrade Tasks

> **See Also:** Some aspects of downgrading are operating system-specific. See your operating system-specific Oracle documentation for additional instructions about downgrading on your operating system.

Supported Releases for Downgrading

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 prior to 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:**
>
> - "Save Oracle Enterprise Manager Database Control Data" on page 3-15
> - "Restoring Oracle Enterprise Manager" on page 6-8

Check for Incompatibilities

Check the compatibility level of your 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 are not able to downgrade.

> **See Also:** "Compatibility Level" on page 1-8

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:** "Downgrading and Compatibility" on page 1-7 and Appendix A, "Behavior Changes"

Perform a Full Backup

Perform a full backup of your Oracle Database 11*g* Release 2 (11.2) database before you downgrade.

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

Downgrade the Database

Complete the following steps to downgrade your Oracle Database 11*g* Release 2 (11.2) database to a major release or a relevant patchset upgrade:

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 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 datetime data types and time zone support

3. 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.

4. Log in to the system as the owner of the Oracle Database 11*g* Release 2 (11.2) Oracle home directory.

5. 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 Real Application Clusters (Oracle RAC) database, then perform this step on all instances.

6. 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 a voting disk path discovered in the previous step:

```
Oracle_Clusterware_Home/bin/crsctl delete css votedisk path
```

7. 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`.

8. Start SQL*Plus.

9. Connect to the database instance as a user with `SYSDBA` privileges.

10. 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.

11. 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" on page 6-8 to validate the synonyms.

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

```
SQL> SPOOL downgrade.log
```

13. 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.

14. 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 12; the suggested name was `downgrade.log`. Correct any problems you find in this file and rerun the downgrade script if necessary.

15. Shut down the instance:

```
SQL> SHUTDOWN IMMEDIATE
```

16. Exit SQL*Plus.

17. 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:** Your operating system-specific Oracle Database 11*g* Release 2 (11.2) installation documents for information about setting other important environment variables on your operating system

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

   a. Stop all Oracle services, including the `OracleServiceSID` 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:** The Oracle Database Platform Guide for Microsoft Windows that is specific to your Microsoft Windows environment for information about stopping services

   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.

Variable	Description
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.

19. 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.

20. At a system prompt, change to the *ORACLE_HOME*/rdbms/admin directory of the previous release.

21. Start SQL*Plus.

22. Connect to the database instance as a user with SYSDBA privileges.

23. Start up the instance:

```
SQL> STARTUP UPGRADE
```

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

```
SQL> SPOOL reload.log
```

25. Run catrelod.sql:

```
SQL> @catrelod.sql
```

The `catrelod.sql` script reloads the appropriate version of all of the database components in the downgraded database.

26. 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
```

27. If you are downgrading to Oracle Database 10*g* Release 1 (10.1.0.5) and you have XDB in your database, then run the following script after running `catrelod.sql`:

```
@dbmsxdbt.sql
```

28. 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 24; the suggested name was `reload.log`. Correct any problems you find in this file and rerun the appropriate script if necessary.

29. 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.

30. 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.

31. 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.

32. Exit SQL*Plus.

Your database is now downgraded.

Perform Post-Downgrade Tasks

This section discusses tasks that might be required after downgrading a database. This section contains the following topics:

- Downgrading Oracle Clusterware Configuration
- Restoring Oracle Enterprise Manager
    - Single-Instance Oracle Database Without ASM
    - Oracle RAC Database Without ASM
    - Single-Instance Oracle ASM Instance
    - Oracle RAC ASM Instance
    - Single-Instance Oracle Database With ASM
    - Oracle RAC Database and ASM Instance
- Enabling Oracle Database Vault
- Reload Oracle Application Express

Downgrading Oracle Clusterware Configuration

Use one of the following options to downgrade the Oracle Clusterware 11*g* configuration:

- Use `srvconfig` from the Oracle Database 11*g* Release 2 (11.2) Oracle home. For example:

  ```
  % srvconfig -downgrade -dbname db_name -orahome 11g_Oracle_home
  ```

- Run `srvctl`. For example:

  ```
  11g_Oracle_home/bin/srvctl remove database -d db_name
  pre-11g_Oracle_home/bin/srvctl add database -d db_name -o pre-11g_Oracle_home
  pre-11g_Oracle_home/bin/srvctl add instance -d db_name -i instance -n node
  ```

  > **Note:** Downgrade the database to release 10*g* before running `emca -restore`.

Restoring Oracle Enterprise Manager

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, you must have saved your Oracle Enterprise Manager files and data before upgrading.

> **See Also:** "Save Oracle Enterprise Manager Database Control Data" on page 3-15

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.

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 Real Application Clusters (Oracle RAC) database or an Automatic Storage Management (ASM) database, as follows:

> **Note:** Use the Oracle Database 11*g* Release 2 (11.2) version of `emca` for this procedure.

Single-Instance Oracle Database Without 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 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
- ASM port
- ASM SID

Oracle RAC ASM Instance

`111Home/bin/emca -restore asm -cluster`

You are prompted to enter the following information:

- Oracle home for the database to be restored
- ASM port

Single-Instance Oracle Database With 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
- ASM port

- ASM Oracle home

- ASM SID [+ASM]

Oracle RAC Database and 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

- ASM port

- ASM Oracle home

- ASM SID [+ASM]

The output of emca will vary with the options you specify and the values you enter at the prompts; but it will look something like this:

```
> emca -restore db

STARTED EMCA at Mar 23, 2007 2:44:17 PM
EM Configuration Assistant, Version 11.1.0.3.0 Production
Copyright (c) 2003, 2005, Oracle.  All rights reserved.
Enter the following information:
Mar 23, 2007 2:44:17 PM oracle.sysman.emcp.util.GeneralUtil initSQLEngine
SEVERE: No SID specified
ORACLE_HOME for the database to be restored: /scratch/oracle/10.2.0/product/db_1
Database SID: DB102
Listener port number: 1521
Password for SYS user:  oracle
Do you wish to continue? [yes(Y)/no(N)]: Y
Mar 23, 2007 2:47:29 PM oracle.sysman.emcp.EMConfig perform
INFO: This operation is being logged at
/scratch/oracle/cfgtoollogs/emca/DB102/emca_2007_03_23_14_44_17.log.
Mar 23, 2007 2:47:30 PM oracle.sysman.emcp.util.DBControlUtil stopOMS
INFO: Stopping Database Control (this may take a while) ...
Mar 23, 2007 2:47:41 PM oracle.sysman.emcp.util.DBControlUtil startOMS
INFO: Starting Database Control (this may take a while) ...
Mar 23, 2007 2:48:06 PM oracle.sysman.emcp.EMDBPostConfig performRestore
INFO: Database Control started successfully
Mar 23, 2007 2:48:06 PM oracle.sysman.emcp.EMDBPostConfig performRestore
INFO: >>>> The Database Control URL is http://stadd17.us.oracle.com:1158/em <<<<<
Enterprise Manager configuration completed successfully
FINISHED EMCA at Mar 23, 2007 2:48:06 PM
```

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 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.
```

Enabling Oracle Database Vault

If you use Oracle Database Vault, then you were instructed to disable it before downgrading your database. You must now re-enable Database Vault. Connect to SQL*Plus as a user who has been granted the `DV_OWNER` role and 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* for instructions
> about enabling Oracle Database Vault

Reload Oracle Application Express

Change to the directory to which you copied the `apxrelod.sql` file (in step 3 of the downgrade steps), and manually reload Oracle Application Express by running the `apxrelod.sql` file.

7

Moving Data Using Data Pump and Export/Import

This chapter guides you through the process of using Data Pump Export and Import utilities, or the original Export and Import utilities, to move data from one database to another.

This chapter covers the following topics:

- When to Use Data Pump Export/Import Versus Original Export/Import
- Export and Import Requirements
- Upgrade the Database Using Export/Import

> **See Also:** *Oracle Database Utilities* for detailed information about Data Pump and the Export and Import utilities

When to Use Data Pump Export/Import Versus Original Export/Import

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 from Oracle Database 10*g* Release 1 (10.1) or higher, Oracle recommends the Data Pump Export and Import utilities in most cases because these utilities provide greatly enhanced performance compared to the original Export and Import utilities.

> **See Also:**
>
> - "Export/Import" 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:

- Much 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.
- 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*)

- Allow you to specify how partitioned tables should be handled during import operations, using the PARTITION_OPTIONS parameter.

- 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 prior to 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 prior to Oracle Database 10*g* Release 1 (10.1), then you must use the original Export/Import utilities.

- If you must downgrade to a release prior to 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

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.

Export and Import Requirements for Upgrades

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 Datab ase 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 perform your 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.

Upgrade the Database Using Export/Import

To upgrade a database using the 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, prior to 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 covered in your operating system-specific Oracle documentation.

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 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 prior to making it available to users. See Step 1 for more information.

13. Complete the procedures described in Chapter 4, "After Upgrading to the New Release".

Importing a Full Database Using a Network Link

As an alternative to the procedure in "Upgrade the Database Using Export/Import" on page 7-4, you can use the Data Pump Import utility with a database link to do a full

database import from a source database to a destination database without intermediate dump files. Follow these steps:

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.

   > **See Also:** Note 466181.1 on Oracle*Metalink* (https://metalink.oracle.com/) for more information on Data Pump Imports using a database link

A

Behavior Changes

This appendix documents important changes in behavior 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 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 covers 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 the new 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 the new 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 following sections 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 issues.

- 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)
- LOG_ARCHIVE_DEST_n Parameters

Initialization Parameters Deprecated in Oracle Database 11*g* Release 2 (11.2)

The following initialization parameters were deprecated in Oracle Database 11*g* Release 2 (11.2). 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.

```
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)

The following static data dictionary views were deprecated in Oracle Database 11*g* Release 2 (11.2).

```
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)

The following 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)

This section lists Oracle Database features 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.

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

- 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)

The following initialization parameters were made obsolete in Oracle Database 11*g* Release 1 (11.1).

> **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.

DDL_WAIT_FOR_LOCKS
LOGMNR_MAX_PERSISTENT_SESSIONS
PLSQL_COMPILER_FLAGS

Static Data Dictionary Views with Dropped Columns in Oracle Database 11*g* Release 1 (11.1)

The following 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 off

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 "Upgrading System Authentication for Oracle ASM Instances" on page 3-3.

> **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 and using them causes errors.

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 10g 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 10g 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 11g 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 11g Release 1 (11.1), and subsequently decide to downgrade the database to Oracle Database 10g 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 11g 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 11g 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 10g 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 10g Release 1 (10.1) to Oracle Database 11g 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 11g 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 10g 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

The following sections describe compatibility and interoperability changes introduced in PL/SQL for Oracle Database 11*g* Release 1 (11.1).

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:** "Configure Fine-Grained Access to External Network Services" on page 4-3

PL/SQL Control Parameters

The behavior of some of the Oracle parameters which control the behavior of PL/SQL changes in 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. Prior to 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 executed prior to 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 segmentation fault, a core file is usually generated. The system default for that core file name is core 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 ora_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.

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

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.

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.

JOB_QUEUE_PROCESSES Parameter

Starting with Oracle Database 11*g* Release 1 (11.1), `JOB_QUEUE_PROCESSES` has changed from a basic initialization parameter to a non-basic initialization parameter. Most databases should be required to have only basic parameters set to run properly and efficiently.

In earlier Oracle Database releases, `DBMS_JOB` and `DBMS_SCHEDULER` shared the same job coordinator, and its behavior was controlled by the `JOB_QUEUE_PROCESSES` parameter. Now `DBMS_JOB` and `DBMS_SCHEDULER` work without setting this initialization parameter. You can still set it if you want, but you are no longer required to set it.

The range of supported values for `JOB_QUEUE_PROCESSES` is still 0 to 1000. If it is set to 0, then `DBMS_SCHEDULER` jobs run and `DBMS_JOB` jobs do not run. The number of slave processes created for `DBMS_SCHEDULER` jobs is autotuned based on the load of the computer.

If `JOB_QUEUE_PROCESSES` is set to a value in the range 1 to 1000, then both `DBMS_JOB` jobs and `DBMS_SCHEDULER` jobs run, and the number of slave processes created for these jobs is autotuned with an additional restriction that the total number of slave processes is capped at the value of `JOB_QUEUE_PROCESSES`.

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 prior to 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. Prior to 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 prior to 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 "Deprecated CONNECT Role" on page 3-12.

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 "TIMESTAMP WITH TIME ZONE Data Type" on page 3-14.

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. Prior to 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 prior to 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
- 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 10g Release 1 (10.1)

The following initialization parameters were deprecated in Oracle Database 10g 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 10g Release 1 (10.1)

The following initialization parameters were made obsolete in Oracle Database 10g 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 prior to 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

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 prior to 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.

Prior to 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/technology/tech/pl_sql/htdocs/ncomp_
  faq.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 10g Release 1 (10.1) use INTEGER arithmetic (approximately 9 significant digits) for some expressions whereas Oracle9i 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 10g 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 10g 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 lower 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.

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.

Prior to 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

This appendix provides scripts that collect optimizer statistics for dictionary objects. By running these scripts prior to performing the actual database upgrade, you can decrease the amount of downtime incurred during the database upgrade.

This process should be tested on a test database 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. The following topics are discussed:

- 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

Numerics

32-bit to 64-bit conversion. *See* word size

A

access control lists (ACLs)
    granting access to network utility packages,   3-13
access control to network utility packages,   A-9
access-control lists (ACLs)
    changed behavior in 11*g* release 1 (11.1),   A-9
applications
    compatibility,   5-1
    linking with newer libraries,   5-4
    running against older server,   5-4
    upgrading,   5-1
        client/server configurations,   5-2
        compatibility rules,   5-3
        options,   5-5
        relinking rules,   5-3
apxrelod.sql file
    reloading after downgrade,   6-3, 6-13
ASM
    cluster ASM upgrade,   4-15
    database upgrade after,   4-17
    disk group compatibility,   A-6
    optional tasks after upgrade,   4-17
ASM_PREFERRED_READ_FAILURE_
    GROUPS,   4-17
automatic maintenance tasks management
    AutoTask,   A-5
Automatic Storage Management (ASM)
    disk group compatibility,   4-16
    file access control,   3
    post-upgrade topics,   4-6
    preferred read failure groups,   4-17
    rolling upgrades
        ASM,   1-11
    upgrading,   3-2, 3-46
automatic undo management
    migrating to,   4-6
    UNDO_MANAGEMENT,   A-11
AutoTask,   A-5

B

backups
    after upgrading,   4-5
    before downgrading,   6-2
    preparing a strategy,   2-9
binary XML storage,   A-8

C

capturing and replaying database workload,   2-7
case sensitivity
    for passwords,   4-5
catdwgrd.sql script,   6-4
CATRELOD.SQL script,   6-6, 6-7
CATUPGRD.SQL script,   3-40
change passwords
    for oracle-supplied accounts,   4-10
client software
    upgrading,   5-4
client-server configurations,   1-5
cluster ASM,   4-15
collecting optimizer statistics,   B-1
command line options
    for Database Upgrade Assistant,   3-31
compatibility
    applications,   5-1
    checking for incompatibilities,   6-2
    compatibility level,   1-8
    COMPATIBLE initialization parameter,   1-7
    downgrading,   1-7
    original Export utility,   7-3
COMPATIBLE initialization parameter,   1-7
    checking,   1-8
    database structures,   1-8
    setting,   4-12
    when to set,   1-8

D

data copying
    using Export/Import,   7-1
data mining models,   A-6
Data Pump
    advantages of using,   2-4
Data Pump Export/Import